



TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Luftfahrt, Raumfahrt und Geodäsie

Lehrstuhl für Flugsystemdynamik

**About robust accuracy, dynamic continuity and
reception resiliency of GNSS user equipment**

Stefan Johannes Haßler, Dipl.-Ing. Univ.

Vollständiger Abdruck der von der Fakultät für Luftfahrt, Raumfahrt und Geodäsie der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Prof. Dr. sc. nat. Christoph Günther

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Johann Dambeck
2. Prof. Dr.-Ing. habil. Erwin Biebl
3. Prof. Dr.-Ing. Florian Holzapfel

Die Dissertation wurde am 25.06.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Luftfahrt, Raumfahrt und Geodäsie am 19.01.2021 angenommen.

“intentionally left free”

Abstract

Against the background of an increasing demand for safety relevant and robust GNSS positioning, this thesis is about analyzing and developing possibilities, to ensure a robust and continuous tracking and positioning solution in low and high dynamic scenarios, even in broadband jamming environments.

The tracking filter respectively tracking architecture is one vital component of the GNSS receiver. Therefore, in the first section, the goal is to find an optimal tracking architecture, providing robust and continuous tracking, even in high dynamic and jammed scenarios. Here for, this thesis clusters the possible tracking architectures into categories and compares the best representative of each category against all others. The categories are scalar classical fixed gain tracking, scalar optimal filter tracking, vector tracking, everything both in total state space and error state space realization, as well unaided and aided. Some tracking architectures are being new developed, while for others, the appropriate architectures are taken from literature. Especially for aided tracking architectures, different new tracking architectures are developed, together with appropriate aiding error models. Comparisons are made, which aiding quality is at least necessary to outperform unaided error state vector tracking.

A novel tuning approach will be introduced, being best suited for optimal filter-based tracking loops. All mission parameters, like platform dynamic and line of sight dynamic characteristics, together with assumed signal to noise and jamming to signal characteristics, being mapped on Kalman filter tuning parameters.

For simulation of all introduced and developed tracking architectures, this thesis develops a high dimensional state space representation of the whole tracking loop in equivalent base band. For stimulation, line of sight phase dynamics, link budget models and jamming to signal ratio models are used. Besides a realistic tracking simulation, including realistic loss of lock prediction, the developed simulation approach provides the closed loop tracking bandwidth for each satellite channel, even in case of vector tracking. Moreover, an analytical stability verification of the tracking loop is possible, by considering the eigenvalues of the respective tracking loop.

Because high interference respectively jamming poses a severe hazard, if GNSS positioning is used in safety critical application, this thesis evaluates the impact of broadband interference signals on tracking behavior and stability. For antijam improvement, different counter measures within a GPS receiver are analyzed and an optimal set of parameters for maximum antijam will be derived, considering the mutual influence of antijam measures within the GPS receiver. Measures for radiation pattern control are one of the most effective antijam measures, especially beam forming is often referred by literature to be the best method. This thesis determines high performing beam forming approaches and evaluates the real antijam improvement, given different array geometries and different imperfections. Because beam forming is not possible in standard civil GPS receivers, Nulling is evaluated in detail. Different Nulling algorithms are compared and a novel Nulling algorithm is developed. The optimal antenna array geometry for different beam forming and Nulling algorithms will be identified, together with the analysis of antijam improvement, given different imperfections. Finally, there will be an evaluation if Nulling provides a comparable antijam improvement as beam forming.

Zusammenfassung

Vor dem Hintergrund einer ständig steigenden Nachfrage nach GPS Positionslösungen in sicherheitsgerichteten Anwendungen, befasst sich diese Arbeit mit der Analyse und Entwicklung von Möglichkeiten, ein robustes und unterbrechungsfreies Tracking von Satelliten, sowie eine robuste Positionslösung, sicherzustellen. Neben Szenarien niedriger Dynamik, werden vor allem hochdynamische Plattformtrajektorien betrachtet, bei zusätzlicher Anwesenheit von Breitbandstörern.

Die Trackingarchitektur ist eine der Hauptkomponenten im GNSS Empfänger, welche verantwortlich für ein robustes und unterbrechungsfreies Tracking ist. Daher entwickelt diese Arbeit im ersten Abschnitt eine Einteilung der möglichen Tracking Architekturen in wenige charakteristische Kategorien und vergleicht anschließend den besten Repräsentanten der jeweiligen Tracking Architektur mit allen Anderen. Für jede dieser Tracking Architekturen, wozu sowohl klassische Tracking Architekturen mit proportionaler und integraler Verstärkung, als auch skalare Kalman Filter basierte Architekturen und Kalman Filter basierte Vektor Tracking Architekturen gehören, werden "Total State" und "Error State" Realisierungen betrachtet und zusätzlich sowohl ungestützte als auch gestützte Varianten. Für einige Kategorien werden eigene, neue Tracking Architekturen entwickelt. Für andere Kategorien werden Literaturrecherchen durchgeführt und die jeweils vielversprechendsten Tracking Filter verwendet. Für gestützte Trackingarchitekturen werden Mindestanforderungen an die Stützinformationen abgeleitet, um äquivalente und sogar bessere Ergebnisse zu liefern, als die beste ungestützte Trackingarchitektur.

Ein neuartiger Tuning-Ansatz wird in dieser Arbeit entwickelt, welcher optimal für Kalman Filter basierte Tracking-Architekturen geeignet ist. Alle Missionsparameter, wie Plattformdynamik und Sichtliniendynamik, sowie das erwartete Signal- zu Rauschverhältnis und Störer- zu Signalverhältnis, können direkt auf Parameter des Kalman Filters abgebildet werden.

Zur Simulation aller eingeführten und betrachteten Tracking Architekturen, entwickelt diese Arbeit eine hochdimensionale, zustandsbasierte Darstellung der gesamten Tracking Architektur. Hierzu erfolgt eine Darstellung durch ein gekoppeltes Differentialgleichungssystem erster Ordnung, wobei ausschließlich mit der Phasendynamik und Modellen für das Signal- zu Rauschverhältnis und Störer- zu Signalverhältnis gearbeitet wird. Zusätzlich zu einer realistischen Simulation des Tracking Verhaltens, wozu auch eine realistische Simulation des durch Störer verursachten Satellitenverlustes zählt, bietet die entwickelte Darstellung der Trackingloop die Möglichkeit, die Trackingbandbreite für jeden Satelliten zu berechnen, sowohl im skalaren als auch im vektoriellen Fall. Weiterhin ermöglicht die entwickelte Simulationsarchitektur eine analytische Berechnung der Stabilität, durch Berechnung von Eigenwerten entlang einer Trajektorie.

Da Jammer bzw. Störsignale ein hohes Risiko für GPS Positionierung darstellen, insbesondere bei Verwendung für sicherheitskritische Aufgaben, analysiert diese Arbeit den Einfluss verschiedener Störertypen auf die Tracking Stabilität und die Tracking Genauigkeit. Für die Verbesserung der Störfestigkeit, werden verschiedene Gegenmaßnahmen innerhalb eines GPS Empfängers analysiert. Es wird eine optimale Kombination der möglichen Maßnahmen zur Störfestigkeitserhöhung hergeleitet, wobei insbesondere die gegenseitige Abhängigkeit dieser Maßnahmen berücksichtigt wird. Es werden Methoden zur Optimierung des Antennen Empfangsfeldes analysiert, insbesondere "Beam Forming", welches von der Literatur überwiegend

als eines der besten Verfahren dargestellt wird. Diese Arbeit identifiziert die wirkungsvollsten “Beam Forming” Algorithmen und bewertet die reale Störerdämpfung sowie den realen Satellitengewinn, abhängig von verschiedenen Antennensegment-Geometrien und möglichen Störfaktoren. “Beam Forming” ist nicht in Kombination mit zivilen standard GPS Empfängern möglich. Aus diesem Grund wird in dieser Arbeit ein alternativer Ansatz, nämlich Nulling, betrachtet. Hierfür werden zunächst für verschiedene Nulling Algorithmen, die Vor- und Nachteile erarbeitet, sowie ein neuartiger Nulling Algorithmus entwickelt. Die Störerdämpfung sowie der Antennengewinn in Richtung der Satelliten, hängen stark von der jeweiligen Antennensegment-Geometrie ab. Es wird für verschiedene Nulling Algorithmen die beste Geometrie identifiziert, sowie die reale Störerdämpfung in Abhängigkeit verschiedener Fehlerquellen diskutiert. Abschließend erfolgt ein Vergleich zwischen “Beam Forming” und Nulling. Ziel ist es zu evaluieren, ob unter realen Bedingungen, unter Berücksichtigung nicht perfekter Antennengeometrien und nicht perfekter Störer-Richtungsschätzung, Nulling eine vergleichbare Störfestigkeitserhöhung wie “Beam Forming” aufweist.

Vorwort

Die vorliegende Arbeit entstand während meiner Zugehörigkeit zum Lehrstuhl für Flugsystemdynamik der Technischen Universität München in den Jahren 2011 bis 2020, parallel zu meiner beruflichen Tätigkeit bei der Firma Siemens AG.

Zuallererst möchte ich mich bei Prof. Dr.-Ing. Florian Holzapfel, dem Lehrstuhlinhaber, bedanken und bei Prof. Dr.-Ing. Johann Dambeck, meinem Betreuer der Promotion. Durch sie wurde es mir ermöglicht, parallel zu meiner beruflichen Tätigkeit bei der Firma Siemens AG, eine Promotion im Bereich der Satellitennavigation durchzuführen.

Besonders möchte ich mich bei Prof. Dr.-Ing. Johann Dambeck für die unzähligen – oftmals bis spät in die Nacht andauernden - fachlichen Diskussionen bedanken. Seine kritischen Nachfragen, seine Denkanstöße und seine Beharrlichkeit, immer den Kern des Problems zu finden sowie die Sachen auf den Punkt zu bringen, waren fordernd und zugleich sehr lehrreich und hilfreich.

Ich möchte mich auch bei meinem Vorgesetzten in der Firma Siemens AG, Herrn Michael Goerner bedanken, der mir durch sein Entgegenkommen mit einer flexiblen Arbeitszeitgestaltung sehr half, die Promotion parallel zu meiner beruflichen Tätigkeit durchzuführen.

Ein weiterer Dank gilt Christopher, Benjamin, Marc Andreas, Thaddäus, Philip und Christoph, die mir immer einen Platz in ihrem Büro am Lehrstuhl freihielten oder freiräumten 😊. Ich habe es an den regelmäßigen Lehrstuhltagen sehr genossen, immer wieder “Uni-Luft” zu schnuppern.

Bedanken möchte ich mich auch bei meinen Eltern, für die Unterstützung während Ausbildung und Studium.

Ein ganz besonderer Dank gilt meiner Frau Martina, die über viele Jahre hinweg, sehr viel Verständnis aufbrachte und mich die ganze Zeit nach Kräften unterstützte, die Promotion fertig zu bekommen.

Contents

- I Motivation 13
- II Introduction..... 17
- III Robust accuracy and dynamic continuity..... 18
 - III - 1 Motivation 18
 - III - 2 Classical scalar fixed gain tracking..... 22
 - III - 2.1 Motivation 22
 - III - 2.2 Tracking architecture..... 24
 - III - 2.3 Tuning with bandwidth focus..... 25
 - III - 2.4 Equivalent base band dynamic state space simulation..... 31
 - III - 2.5 Error covariance model 36
 - III - 2.6 Robustness and accuracy evaluation 40
 - III - 3 Scalar total state optimal filter tracking (unaided) 44
 - III - 3.1 Motivation 44
 - III - 3.2 Tracking Architecture 45
 - III - 3.3 Error modeling..... 49
 - III - 3.4 Novel scenario matched adaptive tuning..... 53
 - III - 3.5 Equivalent base band dynamic state space simulation..... 62
 - III - 3.6 Steady state adaptive tuning vs. continuous adaptive tuning 69
 - III - 3.7 Classical scalar vs. optimal scalar tracking 71
 - III - 4 Scalar optimal filter carrier smoothing..... 78
 - III - 4.1 Motivation 78
 - III - 4.2 Tracking architecture – classical fixed gain carrier smoothing..... 78
 - III - 4.3 Tracking architecture – optimal filter carrier smoothing 87
 - III - 4.4 Equivalent base band dynamic state space simulation..... 88
 - III - 4.5 Classical fixed gain vs. optimal filter carrier smoothing 89
 - III - 5 Scalar error state optimal filter tracking (unaided)..... 91
 - III - 5.1 Motivation 91
 - III - 5.2 Tracking Architecture 92
 - III - 5.3 Equivalent base band dynamic state space simulation..... 94
 - III - 5.4 Scalar total state vs. scalar error state tracking 97
 - III - 6 Positioning solution using optimal filter..... 102
 - III - 6.1 Motivation 102

III - 6.2 Filter architecture	102
III - 7 Total state vector tracking - unaided	108
III - 7.1 Motivation	108
III - 7.2 Tracking architecture.....	109
III - 7.3 Error modeling and tuning	114
III - 7.4 Equivalent base band dynamic state space simulation.....	115
III - 7.5 Scalar total state tracking vs. Vector total state tracking	121
III - 8 Error state vector tracking - unaided	127
III - 8.1 Motivation	127
III - 8.2 Tracking architecture.....	127
III - 8.3 Equivalent base band dynamic state space simulation.....	129
III - 8.4 Vector error state vs. vector total state (unaided).....	136
III - 9 Aided tracking – aiding error modeling.....	141
III - 9.1 Motivation	141
III - 9.2 Aiding source	141
III - 9.3 Noise like aiding error	143
III - 9.4 Low aiding rate	145
III - 9.5 Aiding delay	147
III - 9.6 Tilt error.....	148
III - 10 Aided error state vector tracking	149
III - 10.1 Motivation	149
III - 10.2 Tracking architecture – Aiding as “control input”	150
III - 10.3 Equivalent base band dynamic state space simulation – aiding as control input.....	155
III - 10.4 Tracking architecture – Aiding as measurement.....	161
III - 10.5 Equivalent space band dynamic state space simulation – aiding as measurement	171
III - 10.6 Aiding as control input vs. aiding as measurement	174
III - 10.7 Estimation of aiding delay	179
III - 10.8 Tilt error estimation	184
III - 10.9 Aided vs. unaided tracking	187
III - 11 Aided total state vector tracking.....	195
III - 11.1 Motivation	195
III - 11.2 Tracking architecture.....	195
III - 11.3 Equivalent base band dynamic state space simulation.....	198
III - 11.4 Aiding delay estimation.....	203

III - 11.5 Tilt error estimation	207
III - 11.6 Aided total state vector tracking vs. aided error state vector tracking	210
III - 12 Aided scalar state tracking	214
III - 12.1 Motivation	214
III - 12.2 Tracking architecture.....	214
III - 12.3 Aiding delay estimation.....	220
III - 12.4 Tilt error estimation	222
III - 12.5 Aided scalar tracking vs. aided vector tracking	224
III - 13 Summary of tracking architectures	229
IV GPS Modeling and Simulation	232
IV - 1 Motivation	232
IV - 2 Equivalent base band dynamic state space simulation.....	234
IV - 2.1 Motivation.....	234
IV - 2.2 Simulation architecture and concept.....	236
IV - 2.3 Flight trajectory generation	240
IV - 2.4 Line of sight geometry.....	241
IV - 2.5 Satellite link budget model	244
IV - 2.6 Jammer link budget model.....	247
IV - 2.7 Discriminator error variance prediction.....	250
IV - 2.8 Code and carrier phase dynamic.....	251
IV - 2.9 State space representation of carrier wipe out, correlation and discrimination	252
IV - 2.10 Distributed state space representation	256
IV - 2.11 Centralized state space representation	257
IV - 2.12 Closed loop tracking bandwidth	258
IV - 2.13 Eigenvalue based stability analysis	260
IV - 3 Software-based IF space segment simulator and IF GPS receiver	266
IV - 3.1 Introduction and motivation	266
IV - 3.2 IF band Space Segment Simulator.....	267
IV - 3.3 Software defined IF band GPS receiver with modular tracking kernels	274
V Reception Resilience	283
V - 1 Motivation	283
V - 2 Discriminator error variance in jammed environments.....	284
V - 2.1 Introduction	284
V - 2.2 Considered receiver stages	284

V - 2.3	Jammer power at I and Q.....	285
V - 2.4	Jammer power after correlation.....	287
V - 2.5	Discriminator error variance with spreading effect consideration.....	292
V - 3	Optimal IF bandwidth – given WGN jammer.....	295
V - 4	Optimal IF bandwidth – given different PRN jammer realizations.....	297
V - 5	Overall noise figure minimization.....	299
V - 6	Jammer mitigation by beam forming and nulling.....	306
V - 6.1	Introduction to reception pattern shaping.....	306
V - 6.2	Benefits of reception pattern control in unjammed scenarios.....	311
V - 6.3	Optimal array geometry with real gain patch reception pattern.....	321
V - 6.4	Multi satellite beam forming.....	330
V - 6.5	Performance of different nulling algorithms.....	334
V - 6.6	Comparing beam forming and nulling.....	348
V - 6.7	Comparing beam forming and nulling in case of more than one jammer.....	349
V - 6.8	Sensitivity of beam forming and nulling regarding array geometry errors.....	355
V - 6.9	Beam forming and nulling performance in case of jammer attitude error.....	359
V - 6.10	Conclusion reception pattern shaping.....	361
V - 7	Mission planning in jammed environments.....	362
VI	Summary.....	365
VII	Appendix.....	372
VII - 1	Error modeling.....	372
VII - 1.1	Receiver clock error model.....	372
VII - 1.2	Tuning a Gauss Markov process corresponding to line of sight dynamics.....	375
VIII	Literature.....	377

Nomenclature

s_{IF}	received satellite signal at IF frequency	T_{CAchip}	code chip length
f_{IF0}	L1 base frequency at IF	$\Delta\tilde{\chi}$	received code phase progress
$\Delta\hat{\rho}(t)$	pseudorange change estimation	$\hat{\chi}_{NCO}$	estimated code phase progress
$\delta\hat{f}_c$	code doppler frequency	\mathbf{v}_e^n	platform velocity, given in n-frame coordinates with respect to e-frame coordinates
f_{CA0}	code base frequency	λ	geodetic longitude
$\chi_{CA}(t)$	code phase [chips]	ϕ	geodetic latitude
c	speed of light	$\mathbf{R}_{en}(\lambda, \phi)$	transformation matrix from e-frame into n-frame
$V_{LoS, \#Sv}$	line of sight velocity for satellite Sv	$\hat{\mathbf{x}}_e$	position in e-frame coordinates
$\Delta\chi_{clk,R}$	code phase influence of receiver clock error	$R_N(\phi)$	geodetic ellipsoid normal radius
T_{corr}	correlation time	$\tilde{\mathbf{v}}_{e,aiding}^e$	aiding velocity
$\delta\tilde{\tau}$	discriminator measurement in [s]	$\delta\mathbf{v}_{e,N}^e$	noise-like aiding error
$\Delta\tilde{\chi}$	estimated code phase correction	$\delta\mathbf{v}_{e,R}^e$	rate aiding error
\mathbf{K}	Kalman gain matrix	$\delta\mathbf{v}_{e,00s}^e$	velocity aiding delay error
\mathbf{P}_{k+1}^-	state error covariance matrix before update	$\delta\mathbf{v}_{e,\delta\psi}^e$	tilt aiding error
\mathbf{P}_{k+1}^+	state error covariance matrix after update	Δt_{00s}	aiding delay
\mathbf{R}	measurement error covariance matrix	$\mathbf{R}_{n\tilde{n}}(\Delta\hat{\Phi}, \Delta\hat{\Theta})$	transformation from tilted n-frame into correct n-frame
\mathbf{H}	observation matrix	$\mathbf{\Omega}_{en}^n$	transport rate
l	ionosphere error	$\Delta\chi_{SV}$	received code phase
T	troposphere error	$\Delta\varphi_{SV}$	received carrier phase
$\delta\tau$	discriminator output in [s]	$\Delta\tilde{\chi}_{SV}$	received code phase with errors
$n_{\delta\tau}$	noise discriminator error	$s_{IF}(t)$	received satellite signal at IF band
δt_{Sv}	satellite clock error	I	in-phase signal
δt_{CLK}	receiver clock error	Q	quadrature-phase signal
$\sigma_{n,\delta\tau}^2$	error variance discriminator error	A_{IF}	signal amplitude at IF band

$\frac{C_{IF}}{N_{IF}}$	signal to noise ratio, valid at IF interface, after band pass filter	D	navigation message
J_{IF}	jammer signal at IF band		
n_{IF}	noise at IF band		
$\phi_{IJ,IJ}$	jammer spectral power density, in-phase component		
$\phi_{QJ,QJ}$	jammer spectral power density, quadrature-phase component		
$I_{2,E}$	in-phase signal path after de-spreading with early code replication		
$I_{2,L}$	in-phase signal path after de-spreading with late code replication		
$Q_{2,E}$	quadrature-phase signal path after de-spreading with early code replication		
$Q_{2,L}$	quadrature-phase signal path after de-spreading with late code replication		
$I_{3,E}$	in-phase signal component after correlation (early path)		
k	Boltzmann constant		
$F_A(T_0)$	Noise figure, valid for reference temperature T_0		
$\mathbf{v}_k(\mathbf{k})$	array manifold vector		
\mathbf{k}	wave number		
$B(\omega, \theta, \phi)$	complex antenna gain		

I Motivation

Robust Accuracy and Dynamic Continuity

Standalone GPS navigation or integrated GPS navigation systems are currently used at an increasing number in safety critical applications or at least applications, with safety related requirements to GPS navigation. Examples of such applications are auto landing of civil aircrafts, UAV missions in inhabited areas or even autonomous driving cars or positioning of high-speed trains in the context of safety critical section control.

Regarding line of sight dynamic between the platform and satellites and regarding high dynamic signal to noise ratio variations, the tracking filter itself is the GNSS receiver component, which determines mainly the robustness and accuracy of the raw data solution and positioning solution. This is valid in unjammed as well in high interference environments. Given jamming, there are vital other mitigation measures like CRPA's, but nevertheless, an appropriate tracking architecture and an optimal tuning of the tracking architecture are vital for a robust and accurate solution.

There are a lot of different tracking architectures possible and many are already published in literature. The possible tracking architectures are unaided classical scalar fixed gain architectures, unaided scalar optimal filter tracking in total or error state space and unaided optimal vector tracking in total or error state space. For aiding, also scalar and vector tracking architectures, in total state or error state realization are possible. Designing a GNSS receiver, an important question is, which architecture to select, which architecture provides best results in different dynamic and interference scenarios and what are the differences, advantages or disadvantages of these architectures.

Starting with the question, if scalar optimal tracking filters provide advantages compared to classical scalar fixed gain tracking, a structured comparison of all different realizations of tracking loops is necessary. Second, most paper suggest that vector tracking provides higher robustness, given high dynamic scenarios or high interference environments. But a remaining question is, what is the quantitative advantage, given consistent scenarios regarding dynamic and jamming and given comparable tuning settings. In the context of comparing scalar tracking and vector tracking, also the topic of distributed filter architectures and a centralized filter architecture must be discussed. Centralized or vector tracking filters are more complex. All error components or states must be modeled with a high precision in order to prevent the centralized filter from assigning estimates to wrong states. In contrast, a distributed approach, to which scalar tracking belongs due to local scalar filters for pseudorange tracking and a subsequent filter for calculating the positioning solution, suffers the difficulty of colored error components in the pseudorange estimation. The pseudorange estimations are measurements in the subsequent positioning filter. The question is, how these colored error components can be considered within the positioning filter.

If there is no aiding, the tracking filter must care about the whole platform and satellite dynamic, the clock error dynamic and other low dynamic errors like ionosphere delay. If there is no aiding, literature does not provide a comprehensive answer, if an error state approach does provide any advantages. In case of aided tracking, the remaining dynamics within the tracking filter are the aiding errors, the clock errors and also low dynamic error components. One might expect that in this case, an error state approach is the optimal approach. But even here, literature does not provide any structured and comprehensive answer. That's why this thesis provides a deep and detailed

analysis of total and error state space approaches, given unaided and aided scenarios.

Concerning aiding, essential for good results is an understanding of possible aiding errors and their appropriate consideration respectively estimation within the aided tracking architecture. For that, aiding error models must be developed, fitting the respective architecture. Also, the tracking architecture itself must be tailored for an optimal integration of the aiding error models. For aiding, a scalar optimal filter approach or a vector tracking approach is possible, both in error or in total state space. This thesis develops for all these architectures an aided variant and evaluates at first, if all given aiding errors are observable at all and if, which architecture provides the best solution in unjammed and jammed scenarios.

Unaided vector tracking architectures provide already robust tracking results, to a certain extent even in jammed environments. Aided tracking architectures show robust tracking results at far more higher interference levels, but it is to investigate, if this is valid only for a high aiding rate or even for lower aiding rates. The question is, what is the minimum needed aiding rate, given different dynamic scenarios, to provide at least equal results as the best unaided tracking architecture.

A further interesting question regarding aiding is the way, how aiding is applied to the tracking architecture. In scalar as well in vector tracking architectures, aiding can be applied as an additional measurement or as a kind of “control input”. It is important to evaluate, which option provides better results regarding aiding error estimation and overall performance.

The tuning of tracking loops is a critical task. There are two main tuning requirements being contradictory. For optimal trajectory following capability and the associated small dynamic stress, a high bandwidth is necessary. For maximum robustness respectively high antijam, a low bandwidth is necessary. Therefore, this thesis answers the question, if there is an optimal tuning, at least for optimal tracking filters, being able to map assumed or pre known mission dynamic and mission interference characteristics, directly on tuning parameters. If this is possible, online measurements of actual jamming to signal ratios and short time dynamic predictions, based on flight control commands, can directly be used to adapt the tracking loop in an optimal sense.

GPS Modeling and Simulation

Using GNSS positioning in safety critical applications requires a stability, robustness and accuracy analysis, pre mission, before the receiver is used. The platform dynamic and interference dynamic, with which the receiver is confronted, varies in a wide range, depending on the particular application. Therefore, a powerful GNSS receiver tracking simulation tool is necessary, providing tracking results, being equivalent to the one of real receivers. An additional requirement for simulation is to provide also a deep analytical insight into the tracking loop, like tracking bandwidth calculation and analytical stability verification. The effects of line of sight dynamic as well interference and jamming must be emulated within the tracking simulation, in order to get the same stochastic significance and loss of lock behavior as on real missions.

Optimal tracking filters, as scalar optimal filters, vector tracking filters and aided optimal filters, are adaptive, regarding changing line of sight dynamic and varying interference levels. For application in safety of life applications, it is absolutely necessary to evaluate pre mission, the boundaries of stability, given a defined region of parameter variations. For this, bandwidth variations, eigenvalue variations and stability margin variations along assumed missions, given a defined set of parameter variations, shall be provided by the simulation approach.

For receiver and tracking filter developers, also a pre mission simulation is necessary, in order to evaluate the tracking robustness and accuracy in different scenarios and to calculate eigenvalues and closed loop tracking bandwidths for a defined set of parameter variations.

Moreover, the simulation concept must be designed in a modular way, that it is easy to exchange tracking filters, discriminators and correlators.

Classical GNSS receivers and software GNSS receivers provide tracking results in real time. For pre mission analysis of a wide range of parameters, a simulation being much faster than real time must be developed.

Reception Resilience

For reliable GNSS positioning, even in high interference or jammed environments, dedicated antijam methods must be implemented. There is a vast variety of antijam methods, starting with the usage of CRPAs, pre-whitening transformations, notch filters, minimum bandwidth IF stage filters and optimized minimum bandwidth tracking filters, to mention only the most important one. All these methods are not equally suitable for all types of jammers. Moreover, some of these methods worsen the correlation gain or show other disadvantages.

Especially for wide band interference, having the most severe impact on tracking robustness, the usage of CRPAs, reducing the IF bandwidth or reducing the overall closed loop tracking bandwidth, are the best mitigation strategies.

With focus on wide band jamming, the detailed impact of characteristic jamming types at different stages within a GNSS receiver is of vital interest for an optimal design of the IF bandpass filter and the optimal tuning of the tracking filter.

For tuning the tracking filter, especially Kalman filter-based tracking architectures, the discriminator error variance in high interference environments is a direct input to the measurement covariance matrices. Therefore, the exact discriminator error variance must be predicted, depending on the assumed or measured jamming to signal ratio. Discriminator error variance predictions, available in literature, are mainly based on a white noise jammer assumption at IF stage, together with the closed loop noise bandwidth of the tracking loop. But even in case of white noise jammers, these formulas neglect the effect of spreading and the subsequent correlation. For real jammer impact, stability or loss of lock predictions in simulation and for tuning the real receiver, it is important to compensate these errors, given by standard formulas.

Reducing the IF bandwidth reduces also the overall interference power, entering the tracking loop, but also lowers the correlation gain and therefore the effective signal to noise ratio. Here, a loss of 10 dB can already cause loss of lock, which underlines the importance of a coordinate design of the IF bandwidth together with the correlation time.

Besides white noise jammers, PRN code jammers represent a severe impact on GPS tracking. For PRN code jammers, different designs are possible. It is of great importance, to understand the different design possibilities and their corresponding impact within the GPS tracking loop in order to predict loss of lock, pre mission.

As already mentioned above, CRPAs are one of the most effective mitigation methods, especially for wide band jamming. Basically, two different approaches are used for radiation pattern control. On the one hand it is nulling, which does not require special GPS receiver interfaces and can therefore be used together with civil GPS receivers. On the other hand, it is beam forming, requiring a special receiver interface, which makes it impossible to implement beam forming in off the shelf, civil GPS receivers. Due to its applicability of Nulling in civil GPS receivers, it is important to figure out the differences of beam forming and nulling and to evaluate all factors and design criteria's, contributing to a maximum antijam.

Given the growing demand of GPS navigation in civil, safety of life applications, the question arises, if nulling can provide similar antijam capabilities as beam forming.

II Introduction

In the first section of this thesis, for each meaningful tracking architecture type, a representative tracking loop is introduced and some tracking architectures are being developed. A novel tuning approach is developed, being called “scenario matched tuning”. This new tuning method is explained in detail for each tracking architecture. At first only unaided tracking architectures will be developed. In the second part of this first section, also aided tracking architectures are developed. In this context, different aiding errors are analyzed in detail and aiding error models are developed, allowing the estimation and compensation of the aiding errors. The aiding error models are developed individually for each aided tracking architecture type. Also special tuning challenges, regarding tuning the aiding error models, are discussed in detail.

This first section starts with classical scalar fixed gain tracking architectures, being the standard tracking architectures in today’s mass market GNSS receiver. Building on that, a scalar total state optimal filter tracking architecture is developed and a scalar error state tracking architecture. The next improvement is achieved by total state vector tracking architectures and error state vector tracking architectures. Afterwards, for all these tracking architectures, aided versions are developed.

The order of the different tracking loops is chosen in such a way, promising an increasing better tracking performance and robustness.

The successive architectures are compared to each other, evaluating advantages and disadvantages.

For comparing tracking performance, robustness and stability, a novel state space equivalent base band simulation concept is developed along each tracking architecture. The principles of this simulation approach are developed in the second chapter of this thesis.

The second chapter develops a novel concept for pre mission simulation of GPS tracking. Starting from a detailed satellite and jammer link budget model, considering flight dynamic and antenna characteristics, up to a detailed simulation of the I/Q generation, correlation and the tracking filter itself. The developed concept is set up in a modular way, providing the possibility to exchange the tracking filter in order to evaluate different kinds of filters. The simulation concept is based on an equivalent base band state space approach, providing the additional possibility to derive eigenvalues for each satellite channel – also in case of vector tracking – and to calculate the closed loop tracking bandwidth for each satellite channel.

In this second chapter, the basic principle and concept of this state space equivalent base band simulation concept is developed, using the example of a selected tracking architecture. The state space simulation equations for all other architectures are developed in chapter one, alongside of each tracking architecture.

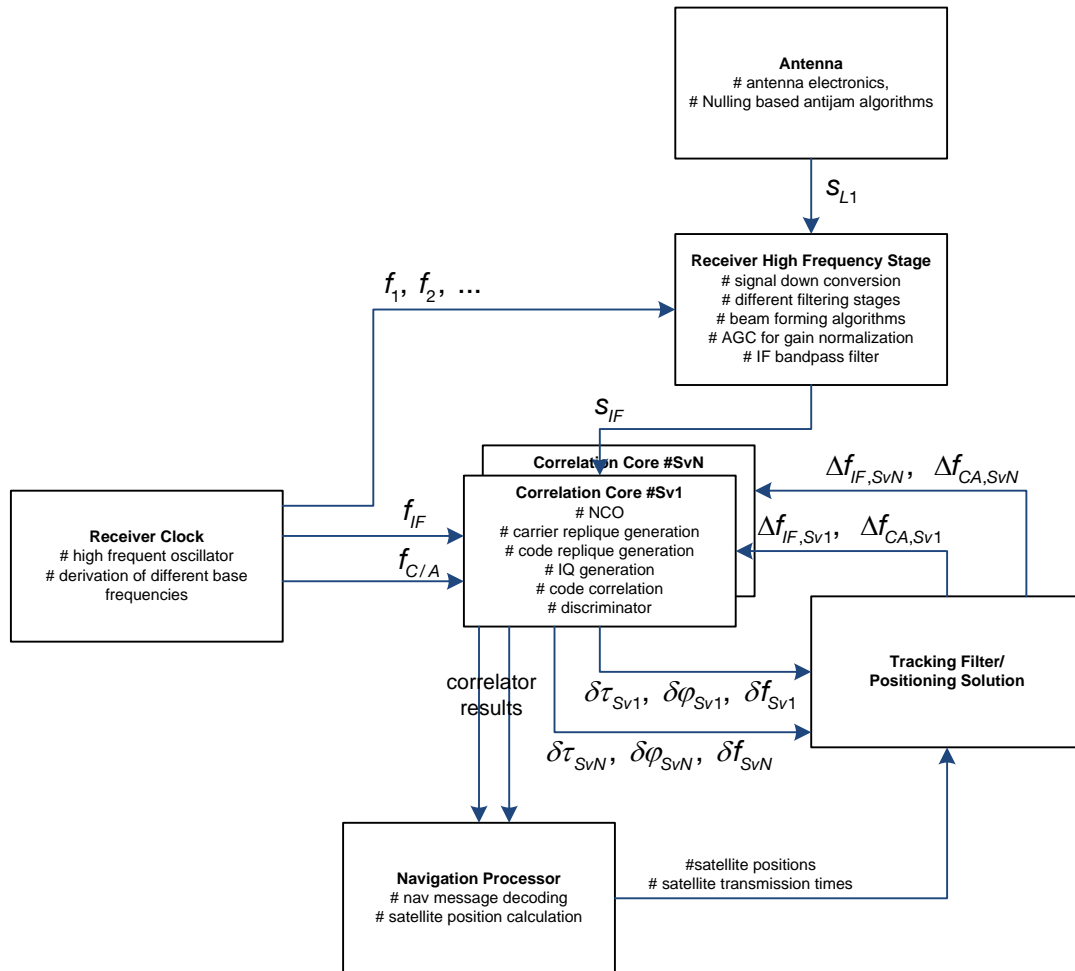
The third part is about improving antijam. At first, the impact of different jammer types on all vital components within the GPS receiver reception and signal processing chain is analyzed. For vital components, different algorithms and modifications to improve antijam, are developed.

Besides the tracking filter design and tuning for maximum antijam, also antenna radiation pattern control is evaluated. Different algorithms for beam steering, beam forming and nulling are discussed and some new algorithms are developed. One central topic are the advantages of beam forming against nulling.

III Robust accuracy and dynamic continuity

III - 1 Motivation

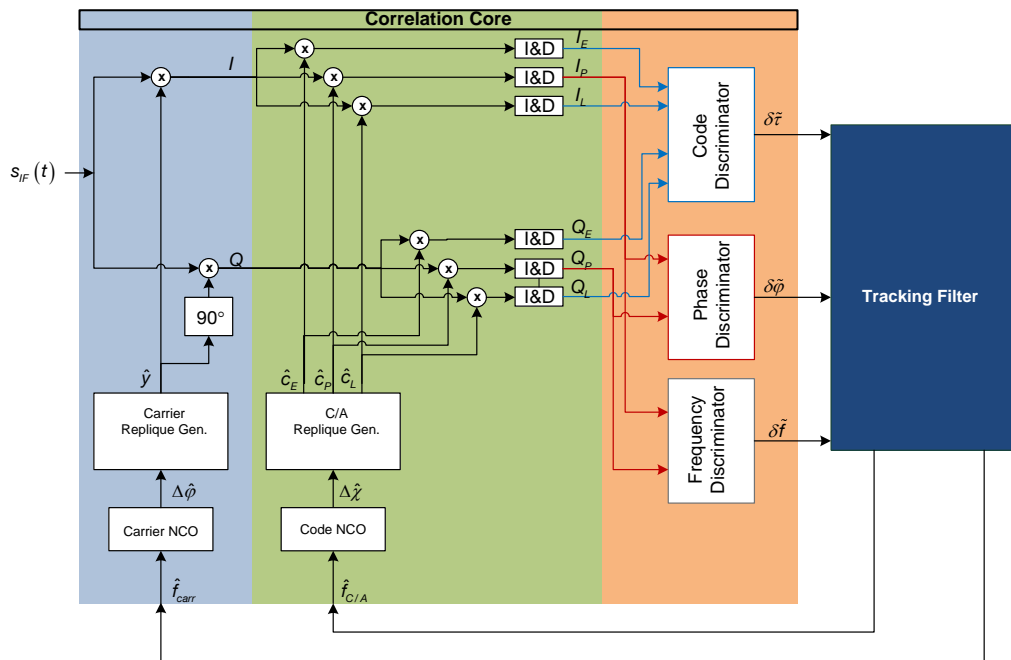
The tracking loop itself is denoted as GPS receiver key technology and mainly responsible for robust and accurate tracking results. The following figure gives a rough overview of a GPS receiver, in order show the location of the tracking loop within the GPS receiver.



III-1 GPS receiver architecture

Similar representations can be found in [1] or [2].

The tracking loop consists out of the correlation core and the tracking filter. The following figure gives an overview of these two main components of the GNSS receiver.



III-2 Basic structure tracking loop

The correlation core generates the code and carrier repliche and measures the phase difference between the replicated code, respective carrier signal and the received ones.

The tracking filter uses the code phase and carrier phase difference as measurements, to adjust the replicated code and carrier phase in order to minimize the corresponding phase difference.

The combination of correlation core and tracking filter represents a control loop.

The tracking architecture itself has a major influence on tracking robustness and tracking stability in different dynamic scenarios and different noise or jamming environments.

There are many possible tracking architectures. But most possible tracking architectures can be subsumed into few categorical types, as given in the following figure.

	Scalar Tracking	Vector Tracking	
unaided	classical scalar 1 st and 2 nd order tracking filter	optimal total state vector tracking filter	total state
	optimal total state scalar tracking filter		
	optimal error state scalar tracking filter	optimal error state vector tracking filter	error state
aided	(no useful variant)	aided optimal total state vector tracking filter	total state
	aided optimal error state scalar tracking filter	aided optimal error state vector tracking filter (aiding as “control input”)	error state
		aided optimal error state vector tracking filter (aiding as “measurement”)	

III-3 Different types of tracking architectures

There is already literature on all components shown in the table above.

In literature like [3] or [1], mostly one special tracking filter is developed and analyzed in some selected scenario. There is no comprehensive analysis and structured comparison of possible architectures.

Classical scalar tracking loops, without any Kalman filter are given for example in [4], [5], [1], [6], [7] and [8]. There, the basic structure for delay locked loops and phase locked loops is introduced. Also tuning is discussed, mainly based on phase margin design and settling properties. In [9], an in-depth analysis of tracking loop poles is shown for determination of tracking stability margin. In [10], the nonlinear characteristic of the discriminator is considered and the tracking loop is formulated as a nonlinear differential equation, where stability is argued by domain of attraction.

Scalar optimal filter based tracking loops are derived and evaluated in [11], [1], [12], and [13]. In [14], also an optimal scalar tracking loop is introduced. The focus of this paper is to map the settled Kalman gains on classical fixed gain tracking filter gains and to show that a settled scalar optimal filter, can also be converted into a classical fixed gain tracking loop structure.

Most introduced scalar optimal tracking loops are given in error state representation and not in total state representation. But there is no analysis of the advantages or disadvantages, using error state space representation, even without any aiding. This chapter evaluates both implementations and compares it afterwards. In literature, there is also no tuning scheme given for the introduced tracking loops. In this chapter, a novel scenario-matched tuning approach will be developed, being applicable for all tracking architectures. Using this new approach, all architectures can be tuned in a similar way, making the architectures more comparable.

An interesting scalar optimal filter based tracking architecture is developed in [15], where the focus are low signal to noise levels, especially for side lobe usage in space navigation. There, an extended scalar Kalman filter is used as tracking filter, whereby the Kalman filter is adapted, using a Bayesian filter approach.

More optimal filter based code tracking loops are given in [16] and [17]. In [18], the equivalence between a phase locked loop and a Kalman filter-based loop filter is analyzed.

Different realizations of vector tracking architectures are given in [1], [19], [20], [19], [21] and [22]. In all these papers, pseudorange or position state vector tracking realizations are given. Most of the stated literature compares the introduced vector tracking architectures against scalar tracking in a special scenario. In [23], different forms of vector tracking architectures are given, including a comparison against classical scalar tracking and Kalman filter based scalar tracking.

Scalar aided tracking architectures are given in [24], [25] and [26]. The given tracking architectures are mostly classical fixed gain tracking loops, whereby the tuning is realized by bandwidth design and pole placement, using Laplace transfer functions. Aiding errors are neither considered nor estimated.

There is almost no literature on aided vector tracking loops. But aided vector tracking architectures are given in the context of tightly coupled tracking architectures as in [27] or [28].

Based on the given literature, this chapter provides the following additional contributions.

Contribution 1: Structured development of different types of tracking architectures, as given in the table above. The architectures are compared to each other and the advantages and disadvantages of the different architectures are evaluated – in unjammed and jammed scenarios, as well as for low and high dynamic trajectories. A novel comparing scheme will be introduced, based on the correlation between tracking error covariance and tracking error correlation time.

Contribution 2: Introduction of a novel equivalent based band representation in state space form. On the one hand, a coupled “distributed” state space form is developed, enabling a tracking simulation in equivalent base band, showing equal results as real tracking.

Contribution 3: Development of new tracking architectures, especially aided ones.

Contribution 4: A novel tuning scheme is developed, which will be named “scenario matched tuning”. Tuning of a GPS tracking loop is always a compromise between optimal dynamic following respectively minimum dynamic stress and minimum pseudorange or positioning error variance. Both tuning criteria are being contradictory. The novel scenario matched tuning, being developed in this chapter, will give inherently the optimal balance between these two tuning requirements. Additionally, for all different developed tracking loops, special tuning hints and pitfalls will be discussed in order to optimize tracking behavior.

Contribution 5: In addition to various unaided tracking architectures, this chapter introduces and evaluates also aided tracking architectures in a structured and comparable way. Starting from aided scalar tracking, to aided total state vector tracking and aided error state vector tracking.

Contribution 6: In case of aiding, for good tracking results, it is essential to consider the possible aiding errors. A further contribution of this chapter is the introduction and modeling of different

aiding errors. For each developed aided tracking architecture, error models will be developed which enable the tracking architecture to consider or estimate different kind of aiding errors.

Contribution 7: There will be an analysis, which aiding quality is necessary to outperform the best unaided tracking architectures.

Contribution 8: Finally, all derived tracking architectures will be compared regarding their antijam capability and tracking behavior in jamming scenarios.

III - 2 Classical scalar fixed gain tracking

III - 2.1 Motivation

The first architecture being discussed is the classical scalar fixed gain tracking architecture as given in [29], based on 1st and 2nd order loop filters.

Classic scalar fixed gain tracking loops are the most used tracking architectures in today's consumer GPS receivers. But especially these tracking architectures suffer a lot of disadvantages which are:

- There is no definite mapping of mission parameters like platform dynamic, expected signal to noise ratios and so on, on tuning figures of the loop filter. Tuning is mostly based on tracking bandwidth and settling characteristics.
- The two main requirements on the tracking loop are minimum tracking bandwidth and minimum dynamic stress. These two requirements are contradictory regarding the closed loop tracking bandwidth. There is no analytical approach to find the optimal bandwidth
- Selection of the appropriate filter order is crucial for minimum dynamic stress. In case of scalar fixed gain loops, the selection must be done manually.
- There is no inherent tracking error covariance information available, which is necessary for positioning solution or even tightly coupled architectures.

As it will be shown in the next sections, the tuning of the tracking loop is about finding the optimal balance between minimum dynamic stress error and minimum noise like error. For fixed gain tracking loops, the closed loop tracking bandwidth is the most important tuning figure. But it is difficult to determine the tracking bandwidth in order to get the optimal balance. Off the shelf receivers implement a tuning, covering a wide range of dynamic situations, which in turn leads to a bad behavior in high noise or even jammed environments.

Even if the mission characteristics are known, there is no exact mapping of dynamic mission figures on the tuning parameters, used in scalar fixed gain tracking loops.

In addition to the closed loop tracking bandwidth, also the order of the tracking filter plays an important role. Depending on the line of sight dynamic respectively the resulting phase dynamic, different orders of loop filters are necessary. The check, if the selected loop filter is appropriate for the given line of sight dynamic, must be done manually by using for example the final value theorem in Laplace space, to verify that there is no remaining control deviation.

A further weakness of scalar fixed gain tracking loops is the missing covariance information of the tracked pseudoranges and range rates. To get a covariance information for the raw data, one possibility is to transform the covariance information of the positioning solution to covariance information for pseudoranges by using the line of sight transformation.

On the one hand, this section discusses elementary topics of classical tracking loops which are

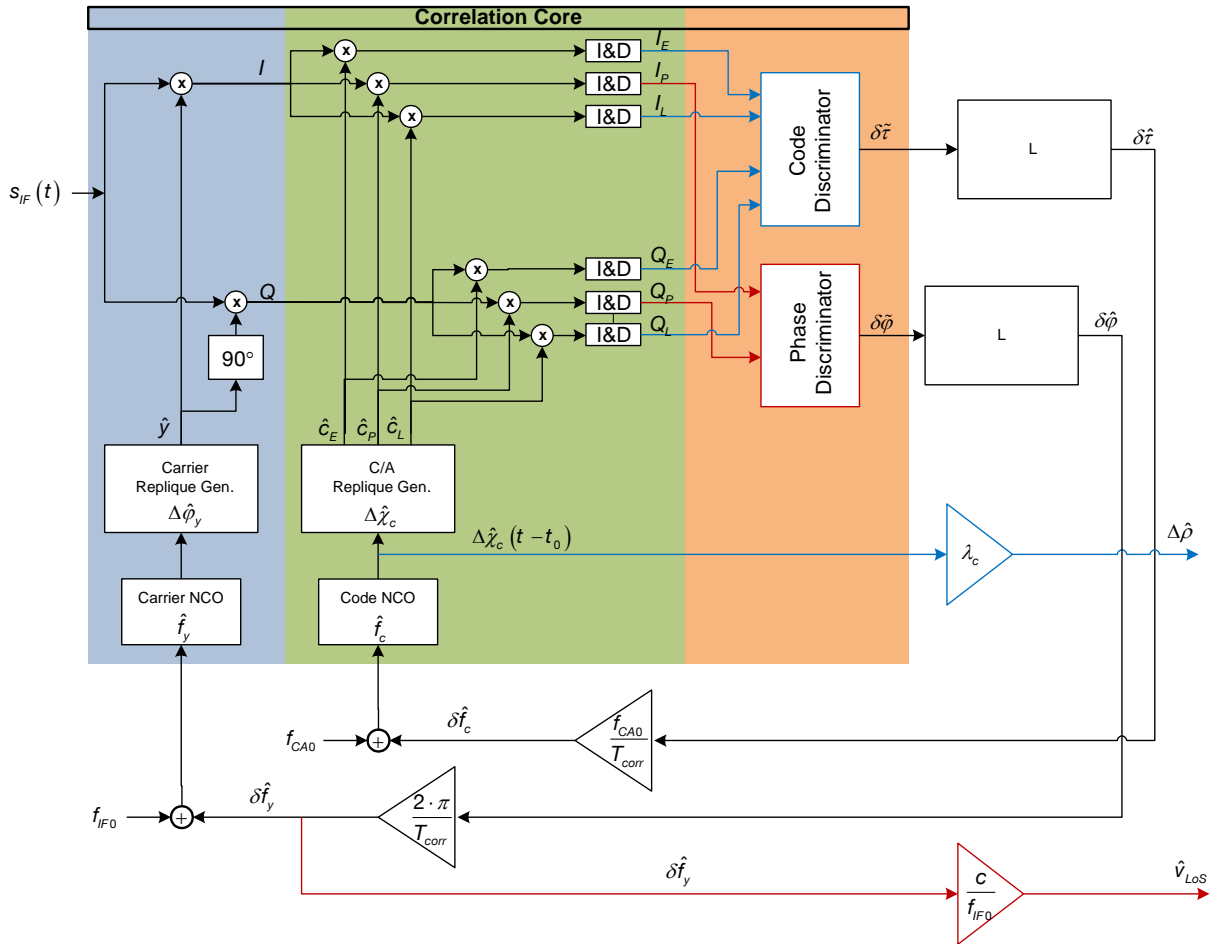
- Brief introduction to classical scalar fixed gain loop filter structures of 1st and 2nd order
- Discussion of tuning criteria being important for loop filter design
- Bandwidth oriented tuning of 1st and 2nd order loop filter, using the available degree of freedom for optimal dynamic behavior.

On the other hand, despite classical scalar fixed gain tracking loops are already well discussed in literature, this section provides the following contributions:

- Laplace representation of 1st and 2nd order tracking loop incorporating also the I/Q generation, correlation and discriminator. This Laplace representation, being also an equivalent base band model of the tracking loop, is developed as a distributed state space form and also as a centralized MIMO state space form, providing many useful analytic possibilities.
- Development of a state space error covariance propagation model for scalar 1st and 2nd order tracking loops. The entries of the resulting state error covariance matrix can directly be used as measurement error covariance information in Kalman filter-based positioning solutions or in tightly coupled filters.

III - 2.2 Tracking architecture

One of the simplest GPS tracking loops is the classical PI loop-filter based scalar code tracking loop. The following schematic shows the principle structure of this tracking loop.



III-4 Classical fixed gain code and carrier tracking loop (uncoupled)

Figure III-4 shows only the tracking loop itself. The input signal s_{IF} is the received signal, down converted to an intermediate frequency. The architecture above shows the code tracking loop together with the carrier tracking loop. Both loops are not coupled in this version.

The first stage within the correlation core generates the I/Q signals in baseband, by using the replicated carrier signal $\hat{y}(t)$ at intermediate frequency for a carrier wipe off. Both, the I and Q signals are multiplied with an early, late and prompt version of the C/A code replica, for de-spreading of received C/A code. De-spreading in this context means, the power of the received C/A code, having a bandwidth of at least 1.023 MHz, is concentrated at 0 Hz. On the other side, the power of jamming signals is spread over a much wider bandwidth.

In the third section, the early, late and prompt signals from I and Q are used to calculate the control deviation, respectively phase or timing difference between the received satellite signal and the replicated signal.

The I/Q generation and spreading processes are well described in [30], [31] and [1].

The measured code and carrier phase errors are provided to the loop filters. Their output is used to control the code and carrier NCO in such a way, that the replicated signals are in phase with the received one. The feedback variables are scaled to frequencies $\delta\hat{f}_c$ and $\delta\hat{f}_y$, which correspond to the line of sight code and carrier doppler frequencies. The basic code frequency f_{CA0} and the basic carrier frequency f_{F0} are added as constant values.

The code tracking loop itself only provides the pseudorange variation $\Delta\hat{\rho}(t)$, since the beginning of tracking.

The complete raw data are calculated according to the following equation

$$\hat{\rho}(t) = \hat{\rho}_0(t_0) + \Delta\hat{\rho}(t) \quad (\text{III-1})$$

The initial pseudorange $\hat{\rho}_0(t_0)$ is calculated with the help of the navigation processor, which is not depicted in the figure above. An explanation of how to calculate this initial pseudorange, using the transmission time within the navigation message, is given in [5] and [4].

Because this chapter is about comparing different tracking architectures, mainly code tracking will be considered.

Typical loop filter architectures for code tracking are of 1st or 2nd order as given below, depending on the expected platform dynamic.

First order loop filter:

$$L_{DLL}(s) = K_p + \frac{K_i}{s} \quad (\text{III-2})$$

Second order loop filter:

$$L_{DLL}(s) = \frac{a_1 \cdot \omega_n \cdot s^2 + a_2 \cdot \omega_n^2 \cdot s + \omega_n^3}{s^2} \quad (\text{III-3})$$

(III-4)

K_p := proportional gain

K_i := integral gain

ω_n := resonance frequency

a_1 := gain coefficient

a_2 := gain coefficient

III - 2.3 Tuning with bandwidth focus

The task of the loop filter L_{DLL} is to control the frequency of the NCO in that way, to synchronize the phase of the received C/A code with the replicated one. Important is, to avoid a phase deviation between the received C/A code and the replicated C/A code of more than 0.5 code chips, because

beyond, the discriminator function is in its nonlinear region, which leads to unstable tracking behavior. Depending on the selected discriminator function, the unstable region can be reached already at smaller phase deviations. A detailed description of different discriminator functions can be found in [2].

Concerning the code tracking loop, there are three main tuning criteria.

Criterion 1: The closed loop tracking bandwidth shall be as small as possible to minimize raw data error variance and to provide a high antijam.

Criterion 2: The tracking bandwidth must be high enough, to allow the loop filter to follow the line of sight dynamic respectively the resulting code phase dynamic.

Criterion 3: Besides the tracking bandwidth, also the order of the loop filter must be chosen, depending on the expected line of sight dynamic, in order to avoid remaining control deviations.

The line of sight dynamic correlates with the Doppler shift of the carrier and C/A code frequency of the received signal and therefore correlates also with the phase dynamic of code and carrier phase, as given below.

The following equation gives the code phase dynamic.

$$\chi_{CA}(t) = \chi_{CA}(t_0) + f_{C/A} \cdot (t - t_0) + \frac{f_{C/A}}{c} \int_{t_0}^t v_{LoS, \#Sv}(t) dt + \Delta\chi_{clk, \#Sv}(t, t_0) + \Delta\chi_{clk, R}(t, t_0) + \Delta\chi_{\epsilon, \#Sv}(t, t_0) \quad (III-5)$$

The clock drift within the satellite and also the clock drift of the receiver NCO causes a phase shift

$$\Delta\chi_{clk, \#Sv}(t) = f_{C/A} \cdot \delta t_{Sv}(t) = f_{C/A} \cdot (t_{Sv}(t) - t) \quad (III-6)$$

$$\Delta\chi_{clk, R}(t) = f_{C/A} \cdot \delta t_R(t) = f_{C/A} \cdot (t_R(t) - t) \quad (III-7)$$

A constant line of sight velocity gives a linear code phase progression and an accelerated line of sight dynamic gives a quadratic code phase progression. The tracking loop must be able to follow this dynamic.

In order to select the appropriate loop filter and especially its order, an analysis of the expected flight envelope respectively line of sight dynamic is necessary. In the context of this work, a DA42 approach as well as a fighter aircraft trajectory will be used to compare the tracking algorithms. A detailed introduction to the mentioned scenarios is given in chapter IV - 2.3 .

Tuning a DLL using a 1st order loop filter

As first order loop filter a PI filter is chosen, which leads to zero static tracking error in case of a constant line of sight velocity.

$$G_{LF}(s) = K_p + \frac{K_i}{s} \quad (III-8)$$

In order to set up the open loop transfer function of the code tracking loop, all components must be written in Laplace space. For the correlation core and NCO, a Laplace representation is developed in the context of equivalent base band representation in chapter IV - 2 . The equivalent

base band respectively Laplace representation of the correlation core is given as

$$G_{corr}(s) = \left(\frac{1}{1 + \frac{T_{corr}}{2} \cdot s} \right) \quad (III-9)$$

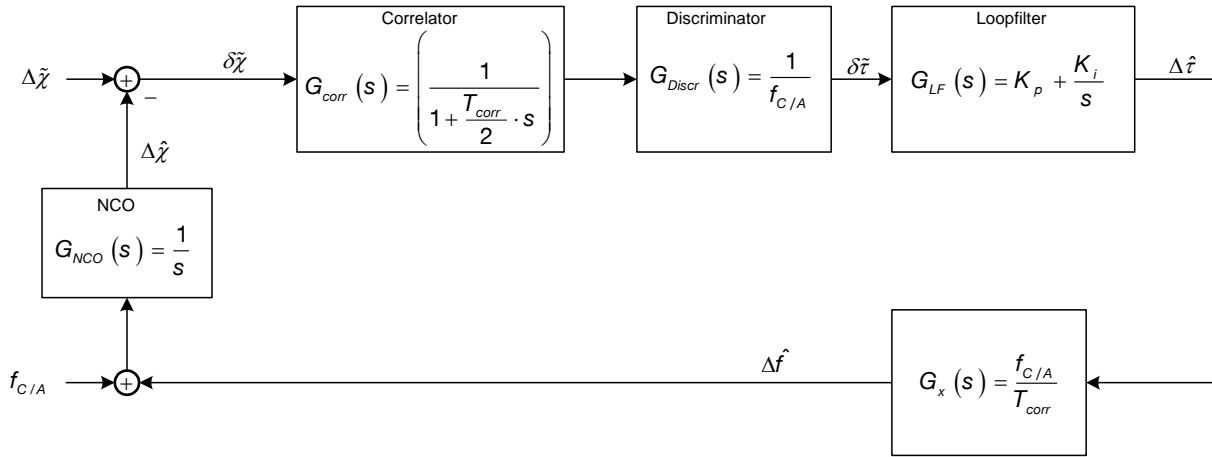
In Laplace representation, the discriminator is just a factor, converting the phase error into a timing error. At least in the linear region.

$$G_{Discr}(s) = \frac{1}{f_{C/A}} \quad (III-10)$$

The controller output $\Delta \hat{\tau}$ is transformed into a frequency correction by using the following equation.

$$G_x(s) = \frac{f_{C/A}}{T_{corr}} \quad (III-11)$$

The following schematic shows the Laplace form of the code tracking loop.



III-5 DLL - Laplace form

The open loop transfer function of the code tracking loop gets

$$G_{DLL,0}(s) = \left(K_p + \frac{K_i}{s} \right) \cdot \left(\frac{1}{1 + \frac{T_{corr}}{2} \cdot s} \right) \cdot \frac{1}{s \cdot T_{corr}} \quad (III-12)$$

The closed loop tracking transfer function can be written as

$$G_{DLL}(s) = \frac{G_{DLL,0}(s)}{1 + G_{DLL,0}(s)} \quad (III-13)$$

Using the final value theorem according to [32, p. 323], it can be shown that in case of a constant line of sight velocity, there will be no remaining steady state error. The control deviation can be written according to the following equation:

$$G_{\delta\chi, DLL}(s) = \frac{1}{1 + G_{DLL,0}(s)} = \frac{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr}}{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr} + K_p \cdot s + K_i} \quad (III-14)$$

A constant line of sight velocity gives a linear increasing code phase input, which can be represented as a ramp input.

$$\lim_{t \rightarrow \infty} \delta\chi(t) = \lim_{s \rightarrow 0} s \cdot \frac{1}{s^2} \cdot G_{\delta\chi, DLL}(s) = \lim_{s \rightarrow 0} s \cdot \frac{1}{s^2} \cdot \frac{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr}}{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr} + K_p \cdot s + K_i} = 0 \quad (III-15)$$

Equation (III-15) shows that a first order filter can eliminate a steady state tracking error in case of a constant line of sight velocity. But the next equation shows that an accelerated line of sight dynamic, leads to a constant tracking error.

$$\lim_{t \rightarrow \infty} \delta\chi(t) = \lim_{s \rightarrow 0} s \cdot \frac{1}{s^3} \cdot G_{\delta\chi, DLL}(s) = \lim_{s \rightarrow 0} s \cdot \frac{1}{s^3} \cdot \frac{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr}}{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr} + K_p \cdot s + K_i} = \frac{T_{corr}}{K_i} \quad (III-16)$$

Tuning of tracking loops is driven mainly by two criteria. The most important is the closed loop tracking bandwidth in order to maximize tracking stability in case of low signal to noise scenarios, or to minimize the impact of jamming. The second important tuning criteria is the capability of the loop, to follow the line of sight dynamic.

As can be seen from equation (III-16), in case of accelerated platform dynamic, there is a remaining tracking error. Therefore, for high dynamic scenarios, at least a 2nd order loop filter is necessary.

For the DA42 trajectory, a tuning bandwidth of 2 Hz is chosen. For the fighter trajectory, a bandwidth of 10 Hz is chosen.

In the first approach, a 1st order loop filter is selected.

For derivation of the loop gains, the bode diagram is required to provide a damping of 3 dB at the chosen closed loop tracking bandwidth.

The expression $\left(\frac{1}{1 + \frac{T_{corr}}{2} s} \right)$ can be neglected for derivation the controller gains. The absolute

value of the closed loop transfer function can be written as following:

$$|G_{DLL}(s)| = \frac{|K_p \cdot s + K_i|}{|T_{corr} \cdot s^2 + K_p \cdot s + K_i|} \quad (III-17)$$

For calculation of controller gains, the Laplace variable s is replaced by $j\omega$.

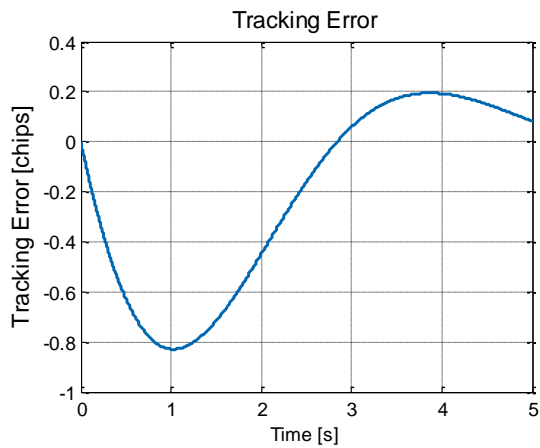
$$|G_{DLL}(j\omega)| = \frac{|K_p \cdot j\omega + K_i|}{|-T_{corr} \cdot \omega^2 + K_p \cdot j\omega + K_i|} = \frac{\sqrt{K_p^2 \cdot \omega^2 + K_i^2}}{\sqrt{K_p^2 \cdot \omega^2 + (K_i - T_{corr} \cdot \omega^2)^2}} = \frac{1}{\sqrt{2}} \quad (III-18)$$

Equation (III-18) is solved with regard to K_p , depending on K_i , for the selected bandwidth.

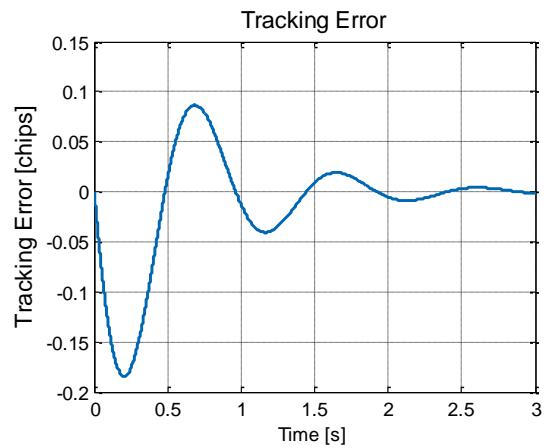
The integral gain K_i provides an additional degree of freedom, which can be used to optimize the dynamic behavior of the tracking loop.

$$K_p = \sqrt{\frac{(K_i - T_{corr} \cdot \omega^2)^2 - K_i^2}{\omega^2}}, \text{ with } (K_i - T_{corr} \cdot \omega^2)^2 - K_i^2 > 0 \quad (III-19)$$

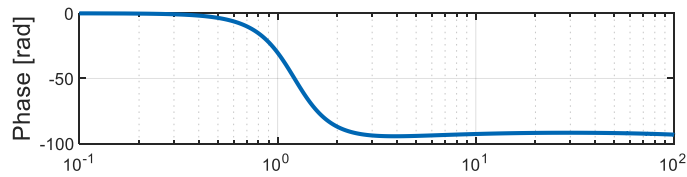
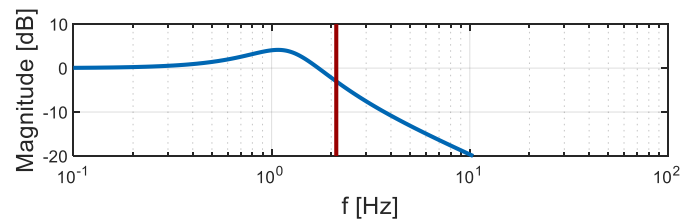
The following figures show the tuning result for the 2 Hz tuning.



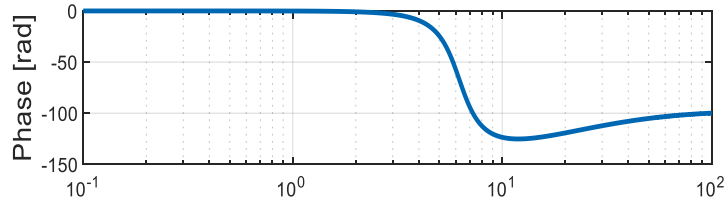
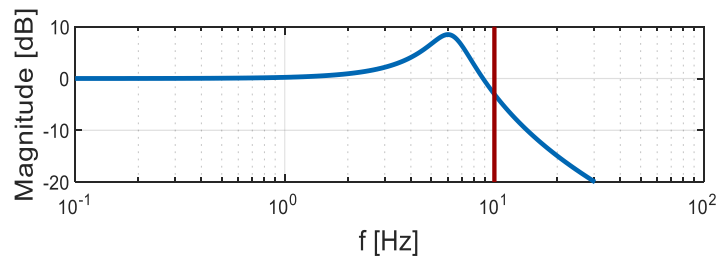
III-6ramp response of transfer function with 1st order filter (2 Hz bandwidth)



III-7ramp response of transfer function with 1st order filter (10 Hz bandwidth)



III-8 Closed loop tracking bandwidth DLL1st order (2Hz)



III-9 Closed loop tracking bandwidth DLL 1st order (10Hz)

For the examples above, as input, a constant line of sight velocity of 500 m/s was chosen.

Tuning a DLL using a 2nd order loop filter

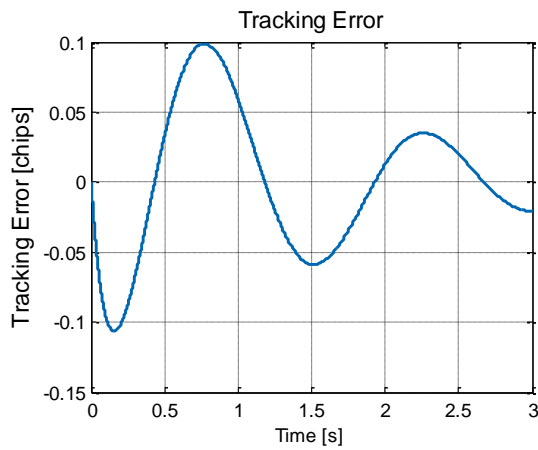
For high dynamic scenarios with large accelerations, a 2nd order loop filter is necessary to prevent remaining tracking errors.

The following equation shows a 2nd order loop filter, as used in [33].

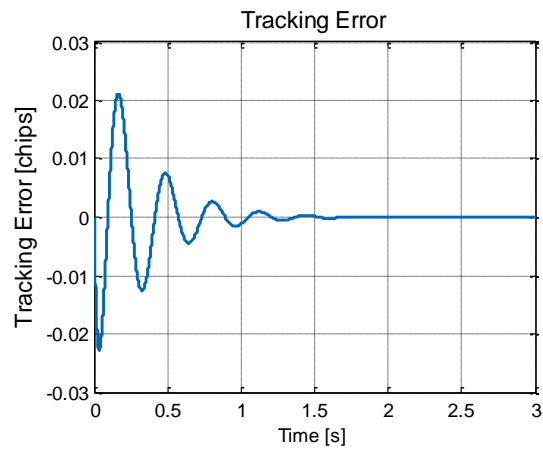
$$G_{LF}(s) = K_p \cdot \frac{a_0 + a_1 \cdot s + a_2 \cdot s^2}{s^2} \quad (\text{III-20})$$

The conversion factor of the discriminator and the conversion factor from the controller output to frequency are subsumed in the gain K_p .

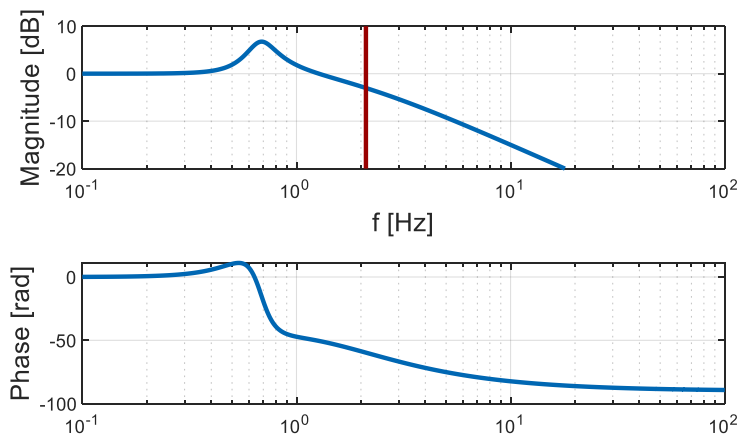
The following figures show the ramp response and closed loop tracking bandwidth for a 2 Hz and 10 Hz tuning.



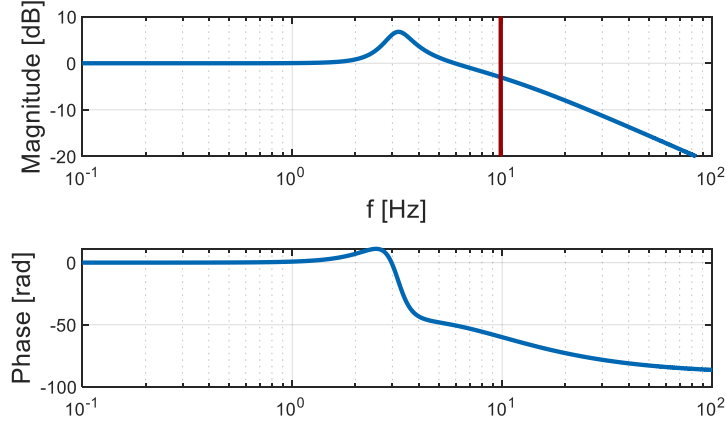
III-10 ramp response of transfer function with 2nd order filter (2 Hz bandwidth)



III-11 ramp response of transfer function with 2nd order filter (10 Hz bandwidth)



III-12 Closed loop tracking bandwidth DLL 2nd order (2Hz)



III-13 Closed loop tracking bandwidth DLL 2nd order (10Hz)

Even if the dynamic characteristics are well known, as in case of the DA42 approach, a direct mapping of dynamic mission parameters on loop gains is not possible. This is one major disadvantage of scalar fixed gain tracking loops. Optimal tracking loops, being developed in the next chapter, do not suffer that disadvantage.

III - 2.4 Equivalent base band dynamic state space simulation

III - 2.4.1 Distributed state space form

The fixed gain scalar code tracking loop can be simulated in equivalent based band, by using a distributed state space approach.

The distributed state space representation is composed out of two components. The equivalent state space representation of the correlation core and the state space representation of the loop filter itself.

The correlation core for the code tracking loop in state space form is

$$\begin{bmatrix} \delta\dot{\tau} \\ \Delta\dot{\hat{\chi}} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi} \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta\chi_{SV} \\ \Delta\hat{f}_{C/A} \end{bmatrix} \quad (III-21)$$

A detailed derivation can be found in chapter IV - 2 .

The loop filter can also be written in state space form. At first the 2nd order loop filter will be considered.

$$\delta\hat{\tau} = \frac{K_p \cdot s + K_i}{s} \cdot \delta\tilde{\tau} \quad (III-22)$$

The loop filter is now written as differential equation in time space.

$$\delta\dot{\hat{\tau}} = K_p \cdot \delta\ddot{\tau} + K_i \cdot \delta\dot{\tau} \text{ with the basic differential equation: } \delta\dot{\hat{\tau}} = \delta\dot{\tau} \quad (\text{III-23})$$

The basic differential equation can now be transformed to state space.

$$\mathbf{z}_{LF} = \mathbf{x}_1 := \delta\hat{\tau} \quad \dot{\mathbf{x}}_1 = \mathbf{0} \cdot \mathbf{x}_1 + \mathbf{1} \cdot \delta\dot{\tau} \quad (\text{III-24})$$

$$\dot{\mathbf{z}}_{LF} \quad \mathbf{A}_{LF} \quad \mathbf{B}_{LF}$$

The input $K_i \cdot \delta\dot{\tau}$ leads to an output $K_i \cdot \mathbf{x}_1$.

The input $K_p \cdot \delta\ddot{\tau}$ leads to an output $K_p \cdot \dot{\mathbf{x}}_1$ and according to equation (III-24), $K_p \cdot \delta\dot{\tau}$.

The complete estimated code phase error in units of [s] after the 1st order loop filter can now be written as

$$\delta\hat{\tau} = \mathbf{C}_{LF}^T \cdot \mathbf{x}_1 + \mathbf{D}_{LF} \cdot \delta\dot{\tau} \quad (\text{III-25})$$

This can be written using the known state space form:

$$\begin{aligned} \dot{\mathbf{z}}_{LF} &= \mathbf{A}_{LF} \cdot \mathbf{z}_{LF} + \mathbf{B}_{LF} \cdot \mathbf{u}_{LF} & \mathbf{z}_{LF} &= \delta\hat{\tau}, \quad \mathbf{A}_{LF} = 0, \quad \mathbf{B}_{LF} = 1, \quad \mathbf{u}_{LF} = \delta\dot{\tau} \\ \mathbf{y}_{LF} &= \mathbf{C}_{LF}^T \cdot \mathbf{z}_{LF} + \mathbf{D}_{LF} \cdot \mathbf{u}_{LF} & \Rightarrow \mathbf{y}_{LF} &= \delta\hat{\tau} \rightarrow \mathbf{C}_{LF}^T = K_i, \quad \mathbf{D}_{LF} = K_p \end{aligned} \quad (\text{III-26})$$

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{0} \cdot \mathbf{x}_1 + \mathbf{1} \cdot \delta\dot{\tau} \\ \delta\hat{\tau} &= K_i \cdot \mathbf{x}_1 + K_p \cdot \delta\dot{\tau} \end{aligned}$$

The feedback, respectively the phase control of the C/A replica is realized via the estimated Doppler frequency $\Delta\hat{f}_{C/A}$. Here for, the loop filter output $\delta\hat{\tau}$ must be converted into the corresponding Doppler frequency.

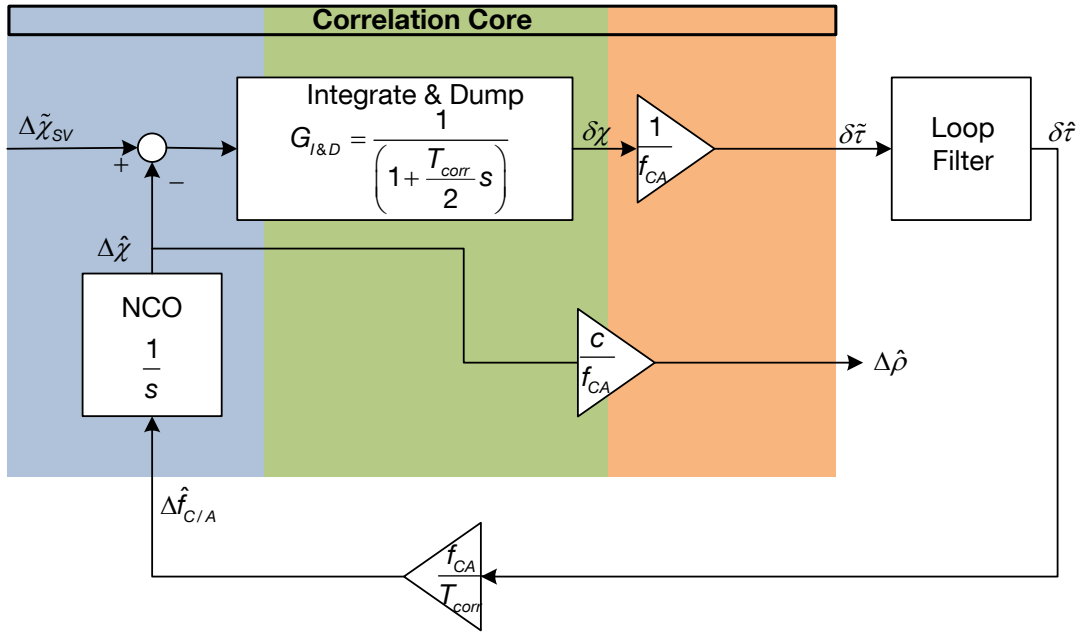
$$\Delta\hat{f}_{C/A} = \frac{f_{C/A}}{T_{corr}} \cdot \delta\hat{\tau} \quad (\text{III-27})$$

Using equation (III-27), the state space form of the loop filter can be written as following.

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{0} \cdot \mathbf{x}_1 + \mathbf{1} \cdot \delta\dot{\tau} \\ \Delta\hat{f}_{C/A} &= \frac{f_{C/A} \cdot K_i}{T_{corr}} \cdot \mathbf{x}_1 + \frac{f_{C/A} \cdot K_p}{T_{corr}} \cdot \delta\dot{\tau} \end{aligned} \quad (\text{III-28})$$

As figure III-14 shows, the NCO command only consists out of the estimated Doppler frequency $\Delta\hat{f}_{C/A}$, caused by the line of sight movement between the GPS receiver and the corresponding satellite. In the real receiver, the NCO command would consist out of $\hat{f}_{C/A} = f_{C/A,0} + \Delta\hat{f}_{C/A}$. Due to equivalent base band simulation, only the code change caused by the line of sight dynamics is considered. The continuous code phase progress, arising from time progress, is not available in equivalent base band.

The pseudorange change since tracking begin, is derived directly out of the estimated code phase of the correlation core, as given in figure III-14.

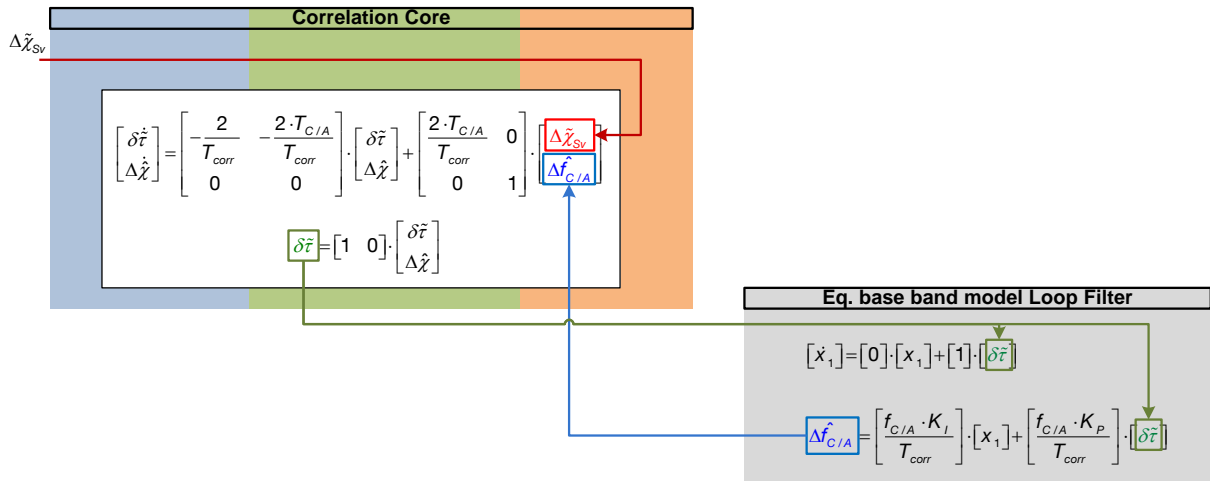


III-14 Deriving pseudorange out of DLL

The line of sight velocity is calculated out of the estimated C/A doppler frequency.

$$v_{LoS} = \frac{\Delta \hat{f}_{C/A}}{f_{C/A}} \cdot c \quad (III-29)$$

The following figure shows the derived distributed state space form, including the coupling between correlation core and loop filter.



III-15 Distributed state space architecture 1st order loop filter

The 2nd order loop filter differential equation can be written as

$$\delta \ddot{\tau} = K_p \cdot a_0 \cdot \delta \tau + K_p \cdot a_1 \cdot \delta \dot{\tau} + K_p \cdot a_2 \cdot \delta \ddot{\tau} \quad (III-30)$$

The differential equation above is transformed into the corresponding state space form.

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot u \\ y &= \mathbf{C}^T \cdot \mathbf{z} + d \cdot u\end{aligned}\quad (\text{III-31})$$

For setting up $\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot u$, at first the following differential equation is used.

$$\delta\ddot{\tau} = \delta\tau \quad (\text{III-32})$$

The complete state space representation of the 2nd order loop filter is given in the following equation.

$$\begin{aligned}x_1 := \delta\hat{\tau} \quad \Rightarrow \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \delta\tilde{\tau} \\ \mathbf{y} = \delta\hat{\tau} &= \underbrace{\begin{bmatrix} K_p \cdot a_0 & K_p \cdot a_1 \end{bmatrix}}_{\mathbf{C}_{LF}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{z}_{LF}} + \underbrace{\begin{bmatrix} K_p \cdot a_2 \end{bmatrix}}_{\mathbf{D}_{LF}} \cdot \underbrace{\delta\tilde{\tau}}_{\mathbf{u}_{LF}}\end{aligned}\quad (\text{III-33})$$

The feedback respectively the phase control of the C/A replica is realized via the estimated Doppler frequency $\Delta\hat{f}_{C/A}$. Here for, the loop filter output $\delta\hat{\tau}$ must be converted into the corresponding doppler frequency.

III - 2.4.2 Centralized state space form

The fixed gain code tracking loop can also be written as a centralized closed loop state space representation. This form is necessary for deriving the closed loop tracking bandwidth and also for stability evaluations through eigenvalues.

In order to get the centralized state space representation, both parts of the distributed state space representation are combined into one state space matrix. The coupling between the two parts is also realized within the system matrix \mathbf{A} .

In general, the closed loop tracking differential equation system can be written as

$$\dot{\hat{\mathbf{z}}} = \mathbf{A} \cdot \hat{\mathbf{z}} + \mathbf{B} \cdot \mathbf{u} \quad (\text{III-34})$$

The centralized state space system is a MIMO system with several inputs and outputs. It is possible to calculate the transfer function between any of the available inputs and outputs. A single path through the MIMO system will be called SISO path.

The most important transfer function is the SISO path from the received code phase to the estimated pseudorange,

$$\Delta\tilde{\chi}_{SV} \rightarrow \Delta\hat{\rho} \quad (\text{III-35})$$

which is the typical closed loop tracking bandwidth.

At first, the centralized state space form for the 1st order loop filter is derived. For that, the correlation core according to (III-21) and the state space form of the loop filter according to (III-33), is combined within one matrix.71

The measurement equation of the loop filter

$$\delta\hat{\tau} = K_i \cdot x_1 + K_p \cdot \delta\tilde{\tau} \quad (\text{III-36})$$

gives the estimated time shift. For integration into the centralized state space matrix, it must be converted to the estimated frequency offset, which is used to control the NCO.

$$\Delta\hat{f}_{CA} = \frac{f_{C/A0} \cdot K_p(t)}{T_{corr}} \cdot x_1 + \frac{f_{C/A0} \cdot K_p(t)}{T_{corr}} \cdot \delta\tilde{\tau} \quad (\text{III-37})$$

$$\begin{bmatrix} \delta\dot{\tilde{\tau}} \\ \Delta\dot{\hat{\chi}} \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 \\ \frac{f_{C/A0} \cdot K_p(t)}{T_{corr}} & 0 & \frac{f_{C/A0} \cdot K_i(t)}{T_{corr}} \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi} \\ x_1 \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A0}}{T_{corr}} \\ 0 \\ 0 \end{bmatrix} \cdot \Delta\tilde{\chi}_{SV} \quad (\text{III-38})$$

The pseudorange is derived according to the next equation.

$$\Delta\hat{\rho} = \begin{bmatrix} 0 & \frac{c}{f_{C/A0}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi} \\ x_1 \end{bmatrix} \quad (\text{III-39})$$

Equation (III-39) shows only the pseudorange variation since the beginning of tracking. The absolute pseudorange is given by

$$\hat{\rho} = \rho_0 + \Delta\hat{\rho} \quad (\text{III-40})$$

The initial pseudorange ρ_0 is derived by the navigation processor within the GPS receiver, by evaluating the reception and satellite transmission time and by counting the samples within the navigation frame since the clock header in the navigation frame. A detailed explanation of initial pseudorange determination can be found in [5] and [4].

The loop gains may be time dependent by using gain scheduling based on actual signal to noise measurements. As a result, also the coefficients of the matrices **A**, **B** are time dependent.

The centralized state space representation of the code tracking loop using a second order loop filter is given in the next equation

$$\begin{bmatrix} \delta \hat{\tau} \\ \Delta \hat{\chi} \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 & 0 \\ \frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} & 0 & \frac{K_p \cdot a_0 \cdot f_{CA}}{T_{corr}} & \frac{K_p \cdot a_1 \cdot f_{CA}}{T_{corr}} \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi} \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A0}}{T_{corr}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi}_{SV} \quad (III-41)$$

Here also, for NCO control the corresponding frequency shift is calculated using the derived observation equation.

$$\Delta \hat{f}_{CA} = \frac{f_{CA}}{T_{corr}} \cdot \delta \hat{\tau} = \frac{f_{CA}}{T_{corr}} \cdot \underbrace{\begin{bmatrix} K_p \cdot a_0 & K_p \cdot a_1 \end{bmatrix}}_{\mathbf{C}_{LF}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{z}_{LF}} + \frac{f_{CA}}{T_{corr}} \cdot \underbrace{\begin{bmatrix} K_p \cdot a_2 \end{bmatrix}}_{\mathbf{D}_{LF}} \cdot \underbrace{\delta \tilde{\tau}}_{\mathbf{u}_{LF}} \quad (III-42)$$

III - 2.5 Error covariance model

The problem of scalar fixed tracking loops is that there is no available error covariance information regarding the derived pseudoranges and range rates. This information is necessary for integrating the raw data into a tightly coupled system. But not only for external integration this information is important, even within the GPS receiver, this information is important for the positioning solution.

In classical fixed gain architectures, where this information is not available, the positioning solution uses some general weighting for the pseudorange measurements, which reflect the signal to noise ratio in the corresponding channel.

Classical GPS tracking architecture which provide raw data at the interface, calculate the corresponding raw data error variances out of the DOP values.

$$\delta \rho_{\#Sv} = \frac{\partial \rho_{\#Sv}}{\partial \mathbf{x}} \cdot \delta \mathbf{x} = \frac{\partial \sqrt{(\mathbf{x}_{\#Sv} - \mathbf{x})^T \cdot (\mathbf{x}_{\#Sv} - \mathbf{x})}}{\partial \mathbf{x}} \cdot \delta \mathbf{x} = \mathbf{h}_{\#Sv}^T \cdot \delta \mathbf{x} \quad (III-43)$$

$$\sigma_{\delta \rho}^2 = \mathcal{E} \left\{ (\delta \mathbf{p} - \mathcal{E} \{ \delta \mathbf{p} \}) (\delta \mathbf{p} - \mathcal{E} \{ \delta \mathbf{p} \})^T \right\} = \mathcal{E} \left\{ \mathbf{H}^T \cdot \delta \mathbf{x} \cdot \delta \mathbf{x}^T \cdot \mathbf{H} \right\} = \mathcal{E} \left\{ \mathbf{H}^T \cdot \mathbf{P}_{xx} \cdot \mathbf{H} \right\} \quad (III-44)$$

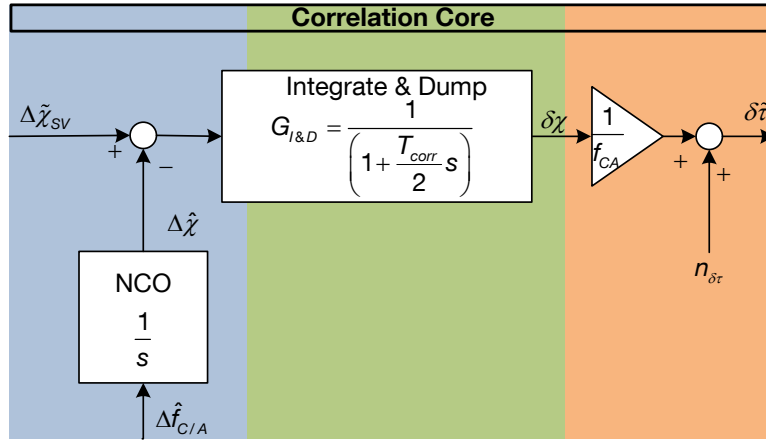
This section introduces a model-based error covariance calculation, which is propagated in parallel to the tracking algorithm and generates for each pseudorange, the appropriate error covariance information. The derived pseudorange error variance can now be used internally for a better tuning of the positioning solution or can be provided at the user interface and used for further integration of the provided raw data.

For error covariance propagation, the following errors can be included into the model.

$$\begin{aligned} \rho &= \tilde{\rho} + \delta \rho \\ \delta \rho &= \mathbf{c} \cdot \delta t_{clk} - \mathbf{c} \cdot \delta t_{Sv} + I + T + n_\rho \end{aligned} \quad (III-45)$$

This section discusses at first the error caused by the received noise or even by jamming signals.

The received noise causes noise like code phase error measurements at the output of the discriminator. Therefore, the effect of the noise can be modeled according to the following figure.



III-16 Considering noise like errors

The noise, respectively its variance $\sigma_{n,\delta\tau}^2$ can be derived, based on the received signal to noise ratio $\frac{C}{N}$ and the noise figure of the whole GPS receiver analog electronic reception chain.

At first, the pure error covariance propagation of noise like errors is shown in the following equation, bias like or long correlated errors are not yet considered.

The output of the loop filter $\delta\hat{\tau}$, if only noise according to figure III-16 is stimulated, gets

$$\delta\hat{\tau}_e = \frac{K_p \cdot s + K_i}{s} \cdot n_{\delta\tau} \quad (\text{III-46})$$

The sub-index “e” is used to mark the states as error states.

In state space form it gets

$$\begin{aligned} \dot{x}_{e,1} &= 0 \cdot x_{e,1} + 1 \cdot n_{\delta\tau} \\ \delta\hat{\tau}_e &= K_i \cdot x_1 + K_p \cdot n_{\delta\tau} \end{aligned} \quad (\text{III-47})$$

In equation (III-48), the coupled correlation core differential equations and the differential equation of the loop filter are combined into one state space form.

$$\delta \dot{\mathbf{z}} = \mathbf{A} \cdot \delta \mathbf{z} + \mathbf{G} \cdot n_{\delta\tau}$$

$$\begin{bmatrix} \delta \dot{\tilde{\tau}}_e \\ \Delta \dot{\hat{\chi}}_e \\ \dot{x}_{e,1} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 \\ \frac{f_{C/A0} \cdot K_p}{T_{corr}} & 0 & \frac{f_{C/A0} \cdot K_i}{T_{corr}} \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_e \\ \Delta \hat{\chi}_e \\ x_{e,1} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{f_{C/A0} \cdot K_p}{T_{corr}} \\ 1 \end{bmatrix} \cdot n_{\delta\tau} \quad (\text{III-48})$$

The corresponding centralized state space form for error covariance propagation, using a second order loop filter gets

$$\begin{bmatrix} \delta \dot{\tilde{\tau}}_e \\ \Delta \dot{\hat{\chi}}_e \\ \dot{x}_{e,1} \\ \dot{x}_{e,2} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{CA}}{T_{corr}} & 0 & 0 \\ K_p \cdot a_2 \cdot f_{CA} & 0 & K_p \cdot a_0 \cdot f_{CA} & K_p \cdot a_1 \cdot f_{CA} \\ \frac{1}{T_{corr}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_e \\ \Delta \hat{\chi}_e \\ x_{e,1} \\ x_{e,2} \end{bmatrix} + \begin{bmatrix} 0 \\ K_p \cdot a_2 \cdot f_{CA} \\ 0 \\ 1 \end{bmatrix} \cdot n_{\delta\tau} \quad (\text{III-49})$$

General error covariance propagation of scalar tracking loops

The discrete solution of the given differential equation system can be written as

$$\delta \mathbf{z}_{e,k+1} = e^{\mathbf{A}_e \cdot (t_{k+1} - t_k)} \cdot \delta \mathbf{z}_{e,k} + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \quad (\text{III-50})$$

using the assumptions: $\mathbf{n}(\tau) = \mathbf{n}_k$ for $t_k < \tau < t_{k+1}$, $e^{\mathbf{A}_e \cdot (t_{k+1} - t_k)} \approx \mathbf{I} + \mathbf{A}_e \cdot \Delta T = \Phi_e$

Using (III-50), the error covariance of the closed loop tracking states and especially the error covariance of the estimated code phase error, which correlates with the error covariance of the estimated pseudorange, can be calculated using the following covariance propagation.

$$\mathbf{P}_{k+1} = \mathcal{E} \left\{ \left(\delta \mathbf{z}_{e,k+1} - \underbrace{\mathcal{E} \{ \delta \mathbf{z}_{e,k+1} \}}_{\rightarrow 0} \right) \cdot \left(\delta \mathbf{z}_{e,k+1} - \mathcal{E} \{ \delta \mathbf{z}_{e,k+1} \} \right)^T \right\} = \mathcal{E} \{ \delta \mathbf{z}_{e,k+1} \cdot \delta \mathbf{z}_{e,k+1}^T \} \quad (\text{III-51})$$

$$\begin{aligned}
\mathbf{P}_{k+1} &= \mathcal{E} \left\{ \left(\Phi_{e,k} \cdot \delta \mathbf{z}_{e,k} + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \right) \cdot \left(\Phi_{e,k} \cdot \delta \mathbf{z}_{e,k} + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \right)^T \right\} \\
&= \Phi_{e,k} \cdot \mathcal{E} \left\{ \delta \mathbf{z}_{e,k} \cdot \delta \mathbf{z}_{e,k}^T \right\} \cdot \Phi_{e,k}^T + \mathcal{E} \left\{ \Phi_{e,k} \cdot \delta \mathbf{z}_{e,k} \cdot \left(\int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \right)^T \right\} + \\
&\quad \dots + \mathcal{E} \left\{ \delta \mathbf{z}_{e,k}^T \cdot \Phi_{e,k}^T \left(\int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \right) \right\} + \\
&\quad \dots + \mathcal{E} \left\{ \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) \cdot \mathbf{n}^T(\chi) \cdot \mathbf{G}_e^T \cdot \left(e^{\mathbf{A}_e \cdot (t_{k+1} - \chi)} \right)^T d\tau d\chi \right\}
\end{aligned} \tag{III-52}$$

$$\text{assuming } \mathcal{E} \left\{ \delta \mathbf{z}_{e,k}^T \cdot \Phi_{e,k}^T \left(\int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) d\tau \right) \right\} \approx \mathbf{0} \tag{III-53}$$

$$\begin{aligned}
\mathbf{P}_{k+1} &= \Phi_{e,k} \cdot \mathbf{P}_k \cdot \Phi_{e,k}^T + \mathcal{E} \left\{ \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathbf{n}(\tau) \cdot \mathbf{n}^T(\chi) \cdot \delta(\tau - \chi) \cdot \mathbf{G}_e^T \cdot \left(e^{\mathbf{A}_e \cdot (t_{k+1} - \chi)} \right)^T d\tau d\chi \right\} \\
&= \Phi_{e,k} \cdot \mathbf{P}_k \cdot \Phi_{e,k}^T + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{G}_e \cdot \mathcal{E} \left\{ \mathbf{n}(\tau) \cdot \mathbf{n}^T(\tau) \right\} \cdot \mathbf{G}_e^T \cdot \left(e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \right)^T d\tau \\
&= \Phi_{e,k} \cdot \mathbf{P}_k \cdot \Phi_{e,k}^T + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \cdot \mathbf{Q} \cdot \left(e^{\mathbf{A}_e \cdot (t_{k+1} - \tau)} \right)^T d\tau
\end{aligned} \tag{III-54}$$

using [34, p. 85]

$$\mathbf{P}_{k+1} \approx \Phi_{e,k} \cdot \mathbf{P}_k \cdot \Phi_{e,k}^T + \mathbf{Q}_k \quad \text{with } \mathbf{Q}_k = \mathbf{Q} \cdot \Delta T^2$$

Using (III-48), the matrix \mathbf{Q}_k for the 2nd order tracking loop gets

$$\mathbf{Q}_k = \Delta T^2 \cdot \begin{bmatrix} 0 \\ \frac{f_{C/A0} \cdot K_p}{T_{corr}} \\ 1 \end{bmatrix} \cdot \sigma_{n,\delta\tau}^2 \cdot \begin{bmatrix} 0 & \frac{f_{C/A0} \cdot K_p}{T_{corr}} & 1 \end{bmatrix} = \Delta T^2 \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\sigma_{n,\delta\tau}^2 \cdot f_{C/A0}^2 \cdot K_p^2}{T_{corr}^2} & \frac{\sigma_{n,\delta\tau}^2 \cdot f_{C/A0} \cdot K_p}{T_{corr}} \\ 0 & \frac{\sigma_{n,\delta\tau}^2 \cdot f_{C/A0} \cdot K_p}{T_{corr}} & \sigma_{n,\delta\tau}^2 \end{bmatrix} \tag{III-55}$$

The matrix \mathbf{Q}_k for the 3rd order tracking loop gets

$$\begin{aligned}
\mathbf{Q}_k &= \Delta T^2 \cdot \begin{bmatrix} 0 \\ \frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} \\ 0 \\ 1 \end{bmatrix} \cdot \sigma_{n,\delta\tau}^2 \cdot \begin{bmatrix} 0 & \frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} & 0 & 1 \end{bmatrix} \\
&= \Delta T^2 \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_{n,\delta\tau}^2 \cdot \left(\frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} \right)^2 & 0 & \sigma_{n,\delta\tau}^2 \cdot \frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} \\ 0 & 0 & 0 & 0 \\ 0 & \sigma_{n,\delta\tau}^2 \cdot \frac{K_p \cdot a_2 \cdot f_{CA}}{T_{corr}} & 0 & \sigma_{n,\delta\tau}^2 \end{bmatrix} \tag{III-56}
\end{aligned}$$

The pseudorange error covariance can be taken from the propagated error covariance matrix \mathbf{P} .

$$\sigma_{\Delta\hat{p}}^2 = \frac{c^2}{f_{C/A}^2} \cdot p_{22} \tag{III-57}$$

Important to keep in mind is that the provided error covariance only considers noise like errors and if modeled, clock errors. Tracking errors due to dynamic stress are not included.

III - 2.6 Robustness and accuracy evaluation

Using the introduced distributed state space representation, together with the equivalent base band code phase dynamic and the signal to noise behavior along the mission, it is possible to evaluate the tracking behavior and the pseudorange error variance along the whole mission.

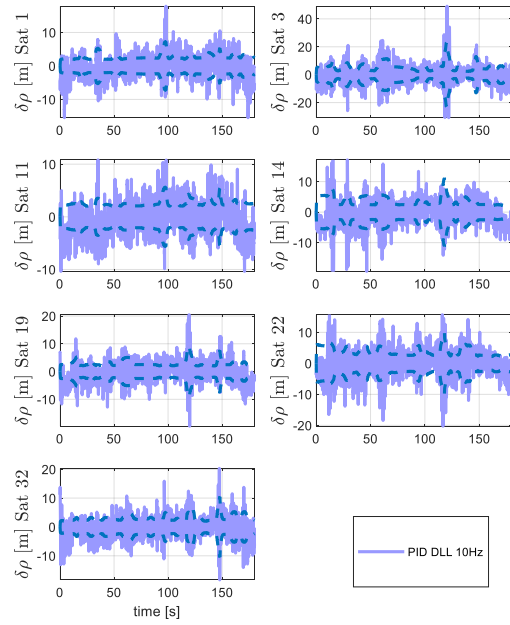
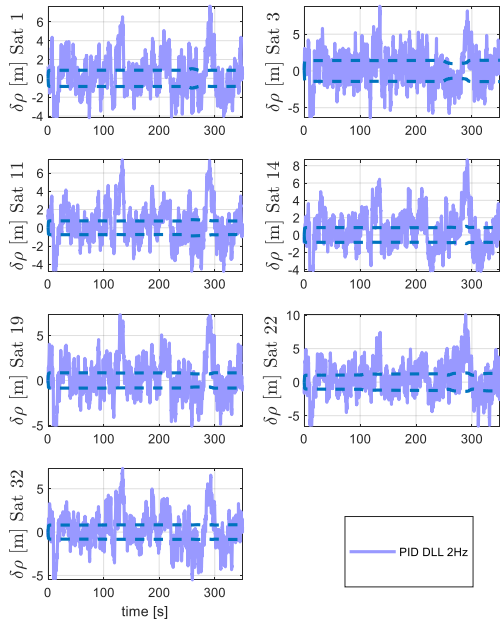
For tracking loop designers, distributed state space simulation provides the possibility to evaluate the tuning performance and to test different tuning settings in different scenarios, pre mission.

For the following simulation, a 2nd order tracking is loop is used.

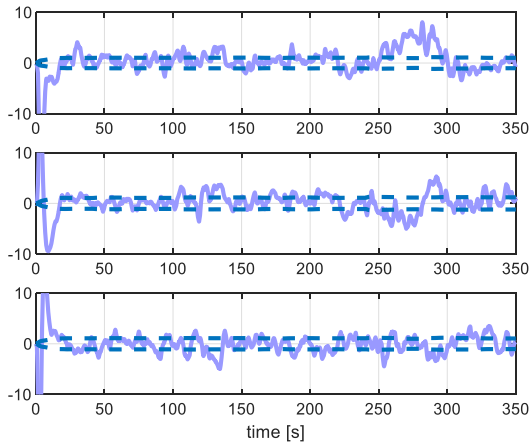
For the DA42 approach, a closed loop tracking bandwidth of 2 Hz is chosen. For the high dynamic trajectory, a closed loop tracking bandwidth of 10 Hz is used.

The first simulation shows the tracking behavior of a 2 Hz, 2nd order loop, given the unjammed DA42 scenario.

UNJAMMED

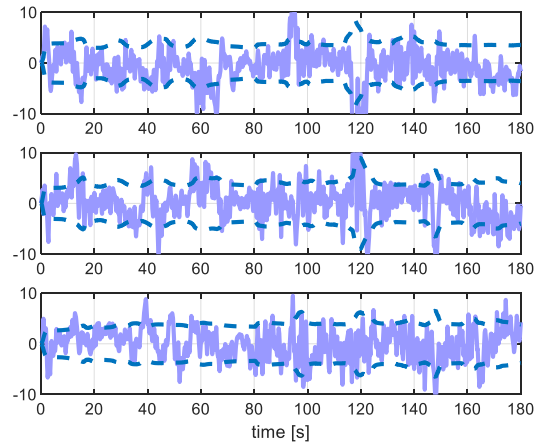


III-17 pseudorange error - DA42

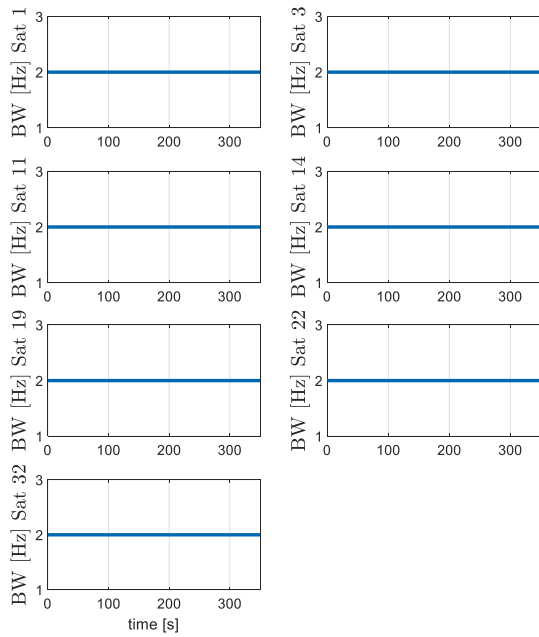


III-19 positioning error - DA42

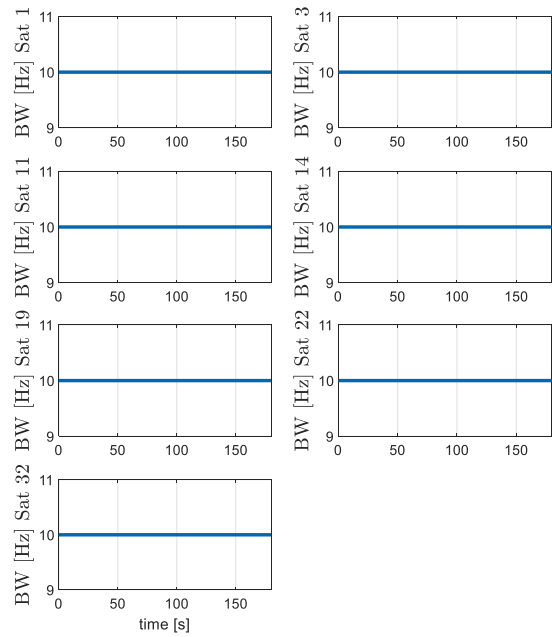
III-18 pseudorange error - high dynamic



III-20 positioning error - high dynamic



III-21 bandwidth - DA42

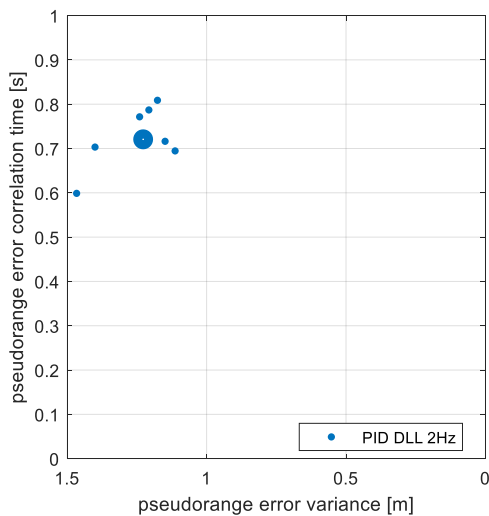


III-22 bandwidth - high dynamic

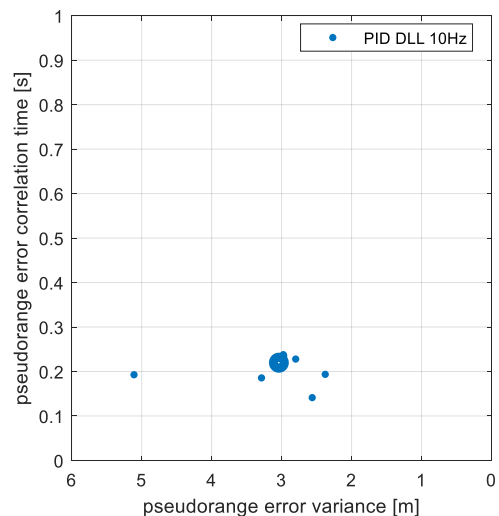
The figures above show the bandwidth of the fixed gain tracking loop for each satellite channel. All channels have the same bandwidth.

For a better comparison, the following figures show normalized pseudorange error variances versus correlation time constants. The most right upper corner would be the best location, having long correlation time and low error variance.

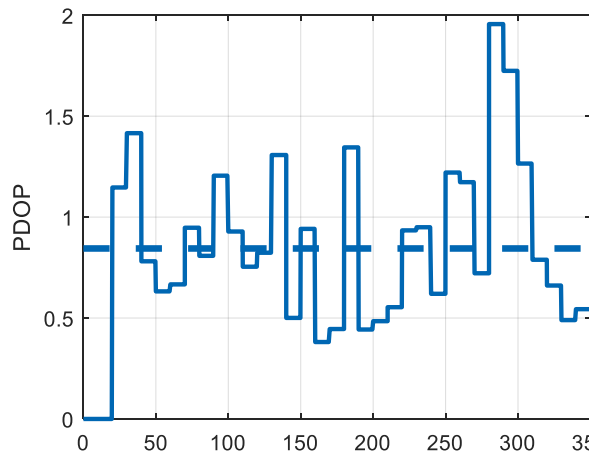
Longer correlation times are desirable from point of loss of lock perspective. Long correlated errors, even having the same error variance, can be handled with much smaller bandwidth, as short correlated errors. In case of long correlated errors, it takes much more time until the accumulated phase difference between the replicated and received code phase becomes critical for causing loss of lock.



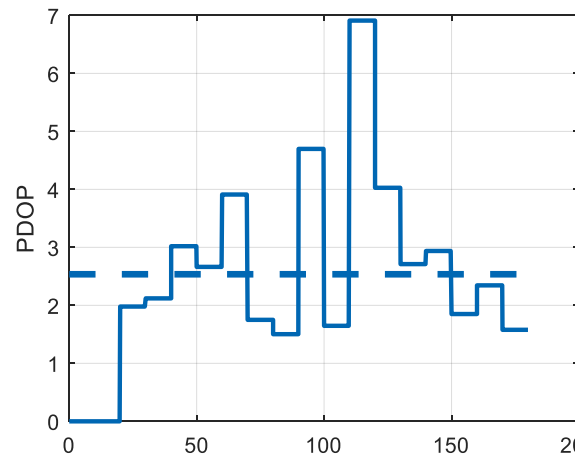
III-23 Comparing raw data - DA42



III-24 Comparing raw data - high dynamic



III-25PDOP - DA42



III-26PDOP - high dynamic

CONCLUSION

The figures above show the tracking behavior of a scalar fixed gain tracking loop. As scenarios, a low dynamic DA42 approach and a high dynamic fighter trajectory are considered.

The tuning is based on tracking bandwidth criteria. Selecting the appropriate bandwidth is always a tradeoff between minimum dynamic stress errors and minimum noise like errors. A too small bandwidth gives small noise like errors, but at the same time high dynamic stress. A too high bandwidth gives minimum dynamic stress errors but high noise like errors. Somewhere in between, there is an optimal bandwidth which gives the best possible trajectory following capability and at the same time, minimum noise like errors.

The bandwidth oriented tuning approach does not directly map mission parameters, like trajectory dynamic and signal to noise ratio, on tuning parameters. For tuning, experience plays a vital role.

The scalar optimal tracking loop, introduced in the next chapter, provide a huge advantage, because this tracking approach provides the possibility, to map mission characteristics on available tuning parameters.

Additionally, the fixed gain tracking loops do not provide some error covariance information about the pseudorange and range rates. But exactly this information is necessary in a following navigation filter or tightly coupled integration. In this chapter, a standalone error covariance calculation was introduced, providing the error covariance information for the raw data, which in turn can be used for the subsequent navigation filter or for tight integration filter. A disadvantage of this standalone error covariance propagation is that dynamic stress is not considered. The error covariance only considers errors based on signal to noise ratio. Nevertheless, the figures above on pseudorange errors confirm the validity of the derived algorithm.

Major pros and cons

Scalar fixed gain - unaided

- ➖ The loop-filter-order must be selected based on the assumed platform dynamic. A manual check is necessary to ensure there is no remaining steady state control deviation.
- ➖ No definite mapping of mission parameters on tuning figures is possible (Tuning is mostly based on designing an appropriate closed loop tracking bandwidth)
It is difficult to find the optimal bandwidth for minimum noise error and minimum dynamic stress at the same time.
- ➖ No error covariance information available. For error covariance propagation, a separate error model must be propagated in parallel. This error covariance information only includes noise caused errors and no errors due to dynamic.
- ➖ Online adaptive tuning based on actual signal to noise ratio measurements is not possible, because there is no direct mapping of signal to noise ratio on tuning parameters. It is only possible by using experience values.

III - 3 Scalar total state optimal filter tracking (unaided)

III - 3.1 Motivation

In this section, a scalar optimal filter-based tracking loop is developed and analyzed. The classical scalar fixed gain loop filter will be replaced by a total state Kalman filter.

There is already some literature on optimal filter tracking. In [35], [11] and [22], different forms of scalar optimal filter tracking architectures are derived, whereby the focus is on designing the dynamic model and implementing the measurement equation for the Kalman filter.

In [22] and [14], it will be shown that after settling, the Kalman filter based approach is equivalent to the classical fixed gain approach. For evidence, the settled Kalman gains are mapped on a classical fixed gain loop filter structure, where each Kalman gain is replaced by corresponding P and I gains.

In addition to available literature, this section derives some important advantages of scalar optimal filter-based tracking architectures, which are:

- The dynamic model within the optimal filter propagation equations gives inherently the correct filter order, provided the model is chosen according to the platform dynamic.
- A novel tuning approach is developed in this section, which will be called “mission matched tuning”. The platform dynamic characteristics and also the assumed noise environment, can directly be mapped on the tuning figures of the Kalman filter.

- Error variance information for carrier and code solution are inherently available, in contrast to the scalar fixed gain approach - where the error variance has to be calculated separately.
- Due to direct mapping of mission characteristics like platform dynamic or the actual signal to noise level on the tuning figures, an easy and fast online adaption of the tracking loop is possible.

If the mission dynamic behavior or even the mission trajectory is known, the dynamic model can be chosen in an appropriate way, which gives inherently the needed filter order to eliminate remaining control deviations and at the same time, gives minimum dynamic stress errors.

In contrast to the tuning methods used for classical fixed gain tracking loops, optimal filter tracking loops provide the possibility to map directly mission parameters, like the acceleration along the trajectory or expected or measured signal to noise ratios, directly on to tuning parameters of the Kalman filter. The difficulty, finding the optimal balance between minimum dynamic stress errors and minimum noise like errors, is solved inherently by this mission matched tuning approach. This mapping allows also an easy online adaption of the optimal filter according the actual noise or even jamming environment.

A further benefit of optimal tracking loops is the error covariance information for all raw data, which is provided inherently by the optimal filter state error covariance matrix. Contrary to the error covariance information of scalar fixed gain tracking loops, which has to be calculated separately, the error covariance of the optimal filter realization also considers dynamic stress errors, which cannot be considered in the error covariance calculation of fixed gain tracking loops.

This section introduces a tracking architecture, using optimal filters and a singer filter like dynamic model. Moreover, the novel mission matched tuning will be introduced in this chapter, where the optimal filter is tuned by a direct mapping of mission characteristics onto the Kalman figures.

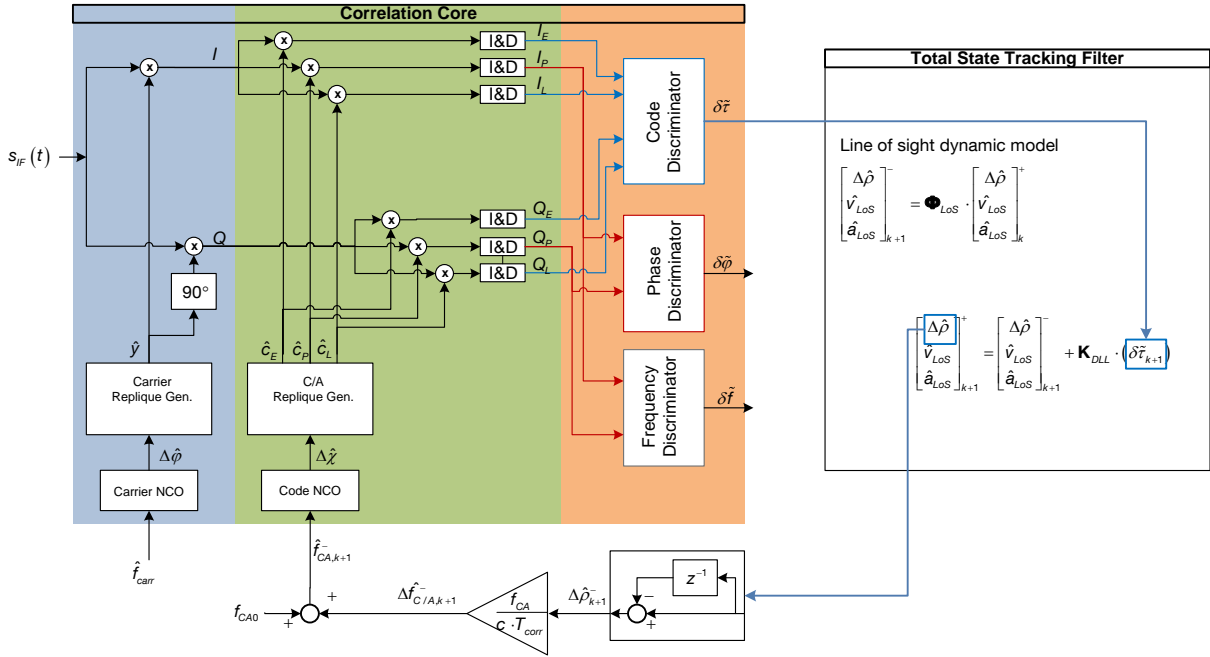
For that, at first a fixed tuning about a complete mission is applied. In a second step, a method for online adaptive tuning will be derived and discussed.

Besides the pure concept, also the exact implementation is discussed in order to avoid problems and some pitfalls.

For pre mission evaluation of the tracking loop and especially the tuning settings, an equivalent base band representation of this tracking loop is developed. For stability evaluation and closed loop tracking bandwidth calculation, a centralized state space representation of this tracking loop will be introduced.

III - 3.2 Tracking Architecture

This section introduces the architecture of an optimal scalar code tracking loop. As can be seen in the figure below, the classical loop filter is replaced by an optimal filter.



-III-27 Scalar optimal filter code tracking loop – frequency feedback

In this section, at first, the unaided total space approach is considered. As tracking filter a "singer filter" is used, which usually is applied in object tracking filters. In [36], this tracking filter is described in detail.

Depending on the line of sight dynamic, a suitable dynamic model must be chosen in order to make the difference between the model dynamic and the real line of sight dynamic as small as possible.

Because the main focus of this chapter is on comparing different tracking architectures, only code tracking will be considered.

The line of sight dynamic is propagated, using the following line of sight dynamic model

$$\dot{\mathbf{z}}_{LoS} = \mathbf{A}_{LoS} \cdot \mathbf{z}_{LoS} + \mathbf{n}_{LoS}$$

$$\begin{bmatrix} \Delta \dot{\hat{\rho}} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_{a,LoS}} \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \end{bmatrix}, \quad \Phi_{LoS} = \mathbf{I}^{3 \times 3} + \mathbf{A}_{LoS} \cdot \Delta T \quad (\text{III-58})$$

The line of sight acceleration is assumed to have zero mean with variations, modeled by a 1st order Gauss Markov process.

The propagation rate of the filter is equal to the output rate of the correlator, which works with a correlation time of τ_{corr} . Important to notice is that the Kalman filter is used in this application in a closed loop feedback system and the Kalman filter represents a kind of optimal loop controller.

The typical Kalman Filter equations are given below. (A detailed derivation of these equations can be found in [37], [38], [39])

$$\begin{aligned}
\text{Propagation: } \quad & \hat{\mathbf{z}}_{k+1}^- = \Phi \cdot \hat{\mathbf{z}}_k^+ \\
& \mathbf{P}_{k+1}^- = \Phi \cdot \mathbf{P}_k^+ \cdot \Phi^T + \mathbf{Q}_k \\
& \mathbf{K} = \mathbf{P}_{k+1}^- \cdot \mathbf{H}^T \cdot (\mathbf{H} \cdot \mathbf{P}_{k+1}^- \cdot \mathbf{H}^T + \mathbf{R})^{-1} \\
\text{Correction: } \quad & \hat{\mathbf{z}}_{k+1}^+ = \hat{\mathbf{z}}_{k+1}^- + \mathbf{K} \cdot (\mathbf{y}_{k+1} - \mathbf{H}^T \cdot \hat{\mathbf{z}}_{k+1}^-) \\
& \mathbf{P}_{k+1}^+ = (\mathbf{I}^{5 \times 5} - \mathbf{K} \cdot \mathbf{H}) \cdot \mathbf{P}_{k+1}^- \cdot (\mathbf{I}^{5 \times 5} - \mathbf{K} \cdot \mathbf{H})^T + \mathbf{K} \cdot \mathbf{R} \cdot \mathbf{K}^T
\end{aligned} \tag{III-59}$$

The propagated states $\hat{\mathbf{z}}_{LoS,k+1}^-$ are transformed into measurement respectively observation space by using

$$\Delta \hat{\tau}_{k+1}^- = \mathbf{h}_F^T \cdot \begin{bmatrix} \Delta \hat{\rho}^- \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ c & & \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho}^- \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix}_{k+1} \tag{III-60}$$

The observation model used in equation (III-60) uses the discriminator outputs as measurements.

A special feature in this Kalman filter application is the calculation of the innovation. Typically, the innovation is calculated within the update equation. Because here, the Kalman filter is used as closed loop “loop filter”, the innovation is calculated within the correlation core.

The estimated time progress $\Delta \hat{\tau}_{k+1}^-$ is applied in the feedback path to the correlation core.

This measured time error $\delta \tilde{\tau}_{k+1} = \Delta \tilde{\tau}_{k+1}^- - \Delta \hat{\tau}_{k+1}^-$ is already the innovation of the Kalman filter. Using this information, the Kalman filter update equation respectively the innovation gets

$$\begin{aligned}
\hat{\mathbf{z}}_{LoS,k+1}^+ &= \hat{\mathbf{z}}_{LoS,k+1}^- + \mathbf{K} \cdot (\delta \tilde{\tau}_{k+1}) \\
\delta \tilde{\tau}_{k+1} &= (\Delta \tilde{\tau}_{k+1}^- - \Delta \hat{\tau}_{k+1}^-) = \left(\Delta \tilde{\tau}_{k+1}^- - \begin{bmatrix} 1 & 0 & 0 \\ c & & \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho}^- \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix}_{k+1} \right)
\end{aligned} \tag{III-61}$$

The index F of the vector \mathbf{h}_F^T stands for “feedback”. In this example, the observation matrix \mathbf{H}^T used for calculating the Kalman gain, is identical to \mathbf{h}_F^T . If consider-states will be used like in the next section, it is necessary to differ between the transformation to observation space, used for Kalman gain calculation and the transformation, used for generating the real feedback.

The NCO is controlled by frequency. The NCO command is built out of the base C/A code frequency and the frequency correction being necessary to correct the phase difference between the received C/A code and the replicated one.

$$\hat{f}_{CA,k+1}^- = f_{C/A,0} + \Delta \hat{f}_{CA,k+1}^- \tag{III-62}$$

The frequency correction is calculated out of the estimated time progress according to the

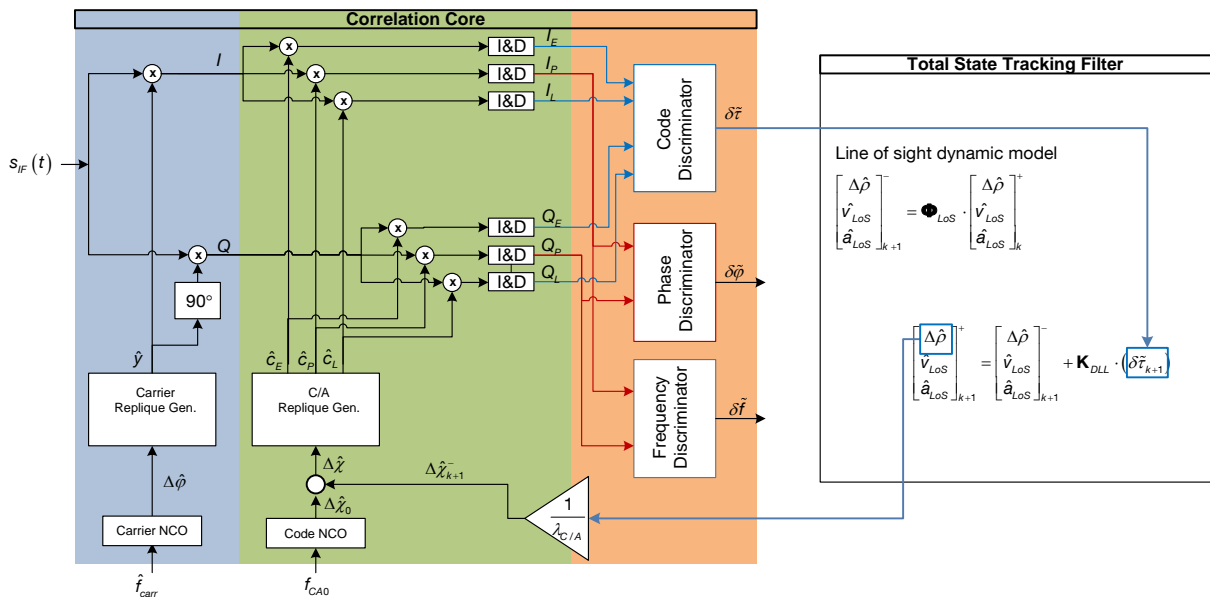
following equation.

$$\Delta \hat{f}_{CA,k+1}^- = \frac{f_{CA} \cdot (\Delta \hat{t}_{k+1}^- - \Delta \hat{t}_k^-)}{T_{corr}} \quad (\text{III-63})$$

The time progress $\Delta \hat{t}_{k+1}^-$ is the absolute time since the start of tracking. The frequency correction is calculated based upon the difference between the last and the actual time progress estimation.

In the architecture given in figure -III-27, as correlation core interface, the code doppler frequency was used. This estimated code doppler frequency is internally integrated by the NCO to the corresponding estimated code phase progress $\Delta \hat{\chi}_{k+1}^-$.

In case of software defined GPS receivers or even in future GPS receivers, where a direct interface to replica generator will be most likely available, the term $\Delta \hat{\chi}_{k+1}^-$ can directly be applied to control the phase of the replica generation, as it is shown in the following architecture.



III-28 Scalar Optimal Gain Total State DLL - Direct code phase interface

The tracking architecture in figure III-28 is equal to the architecture shown in figure -III-27. For simulation purpose and closed loop state space models to be introduced, the architecture with direct phase control will be used.

The tracking results respectively the raw data $\Delta \hat{\rho}$ and \hat{V}_{LoS} , can directly be taken from the states of the line of sight dynamic model. In order to get the absolute pseudorange, the initial determined pseudorange must be added, because the states only show the pseudorange change since the beginning of tracking.

$$\hat{\rho} = \hat{\rho}_0 + \Delta \hat{\rho} \quad (\text{III-64})$$

The initial pseudorange $\hat{\rho}_0$ is calculated by the navigation processor. A detailed description of how this initial pseudorange can be calculated, is given in [5].

An important advantage of using a Kalman filter within the code tracking loop, is the inherent availability of the actual state error covariance information given by the state error covariance matrix \mathbf{P} . The corresponding diagonal elements directly represent the pseudorange error covariance $\sigma_{\hat{\rho}}^2$ and the velocity error covariance $\sigma_{\hat{v},LoS}^2$.

III - 3.3 Error modeling

Error components

The code phase error measurement at the output of the discriminator contains a noise like error, caused by the received noise or even caused by jamming, the ionosphere and troposphere error, ephemeris error, satellite clock error, the receiver clock error and an error caused by dynamic stress, which can also be interpreted as a control deviation. Especially in unaided architectures like the one considered in this section, dynamic stress errors pose a major error component. The dynamic stress can be reduced by using a higher tracking bandwidth on the cost of higher errors caused by noise. The optimal tuning must be a compromise between minimum noise caused errors and minimum dynamic stress. This optimal tuning can be realized by a novel scenario matched tuning and is topic of the following section.

Within the correlation core, the code phase error is calculated according to

$$\delta\tilde{\chi}_{k+1} = \Delta\tilde{\chi}_{k+1} - \Delta\hat{\chi}_{k+1}^- \quad (\text{III-65})$$

The tracking error respectively discriminator output is converted to units of seconds. The code phase control is again applied in units of code phase.

$$\begin{aligned} \delta\tilde{\tau}_{k+1} &= \Delta\tilde{\tau}_{k+1} - \Delta\hat{\tau}_{k+1}^-, \quad \text{using } \delta\tau = \frac{1}{f_{CA}} \cdot \delta\chi \\ \Delta\hat{\chi}_{k+1}^- &= f_{CA} \cdot \Delta\hat{\tau}_{k+1}^- \end{aligned} \quad (\text{III-66})$$

The following expression gives the error components at the output of the discriminator which are considered. Ephemeris errors and further minor error components are not considered.

$$\begin{aligned} \delta\tilde{\chi} &= \Delta\tilde{\chi}_{k+1} - \Delta\hat{\chi}_{k+1}^- = \Delta\chi_{k+1} + I_{\chi,k+1} + T_{\chi,k+1} + f_{CA} \cdot \delta t_{Sv} + f_{CA} \cdot n_{\delta\tau} - \Delta\hat{\chi}_{k+1}^- - f_{CA} \cdot \delta t_{CLK} \\ \delta\tilde{\tau} &= \delta\tau + I_{t,k+1} + T_{t,k+1} - \delta t_{CLK} + \delta t_{Sv} + n_{\delta\tau} \end{aligned} \quad (\text{III-67})$$

The innovation, respectively the actual deviation between the received code phase and the replicated one - converted to time error - consists out of the given error components.

In the following, only the noise like errors, the receiver clock error and the error part $\delta\tau$, caused by dynamic stress, respectively the inability of the tracking loop to follow the line of sight dynamic, are considered.

All other error components, like ionosphere error, troposphere error and ephemeris error are estimated in the positioning filter and not in the scalar tracking stage. But nevertheless, the focus of this thesis is to compare different tracking architectures. For that, only the dominant parts of the user equivalent range error (UERE) are considered, respectively the parts having a tuning relevant impact.

A very important error component in GPS tracking is the dynamic stress, which happens if the tracking filter is not able to follow the line of sight dynamic in a proper way. This error type is a control deviation and can become very large in case of wrong tuning - it can even lead to loss of lock in high dynamic scenarios. Typical GPS literature has not the focus of a proper tracking loop tuning, that's why this important error type gets not the attention, which it should have.

Dynamic stress and dynamic model uncertainty

Knowing the expected platform dynamic, a dynamic model for the used optimal filter is selected.

In the following, a zero mean, line of sight acceleration is assumed. Within the dynamic model of the Kalman filter, the line of sight acceleration is modeled as a 1st order Gauss Markov process. The resulting model uncertainty enables the Kalman filter to estimate the correct line of sight figures.

For tuning, as it will be shown in the following section, the deviation of the line of sight acceleration from the constant mean value, is assumed to be a Gauss Markov process. Correlation time and variance are the two characteristic figures of the Gauss Markov process, which will be determined by analyzing the given real process.

The following equations show the derivation of the model error covariance matrix, given the line of sight dynamic model as well as the driving noise for the Gauss Markov process. The clock error is not yet considered in the dynamic model.

The discrete version of the model error covariance matrix can be calculated, by solving the line of sight dynamic differential equation, depending on the seed noise vector \mathbf{n}_{LoS} .

$$\begin{aligned}\dot{\mathbf{z}}_{LoS} &= \mathbf{A}_{LoS} \cdot \mathbf{z}_{LoS} + \mathbf{n}_{LoS} \\ \mathbf{z}_{LoS,k+1} &= e^{\mathbf{A}_{LoS} \cdot (t_{k+1}-t_k)} \cdot \mathbf{z}_{LoS,k} + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}_{LoS} \cdot (t_{k+1}-\tau)} \cdot \mathbf{n}_{LoS}(\tau) d\tau\end{aligned}\quad (III-68)$$

$$\text{using the assumptions: } \begin{aligned} \mathbf{n}_{LoS}(\tau) &= \mathbf{n}_{LoS} \text{ for } t_k < \tau < t_{k+1} \\ e^{\mathbf{A}_{LoS} \cdot (t_{k+1}-t_k)} &\approx \mathbf{I} + \mathbf{A}_{LoS} \cdot \Delta T = \mathbf{\Phi}_{LoS} \end{aligned}, \quad (III-69)$$

$$\begin{aligned}\mathbf{z}_{LoS,k+1} &= \mathbf{\Phi}_{LoS} \cdot \mathbf{z}_{LoS,k} + \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_{LoS}(t_{k+1}, \tau) \cdot \mathbf{n}_{LoS} d\tau = \mathbf{\Phi}_{LoS} \cdot \mathbf{z}_{LoS,k} + \mathbf{n}_{LoS,k} \\ \mathbf{Q}_k &= \mathcal{E} \left\{ \mathbf{n}_{LoS,k} \cdot \mathbf{n}_{LoS,k}^T \right\} = \mathcal{E} \left\{ \int_{\Delta T} e^{\mathbf{A}_{LoS} \cdot \tau} \cdot \mathbf{n}_{LoS}(\tau) d\tau \cdot \left(\int_{\Delta T} e^{\mathbf{A}_{LoS} \cdot \xi} \cdot \mathbf{n}_{LoS}(\xi) d\xi \right)^T \right\} \\ &= \mathcal{E} \left\{ \int_{\Delta T} \int_{\Delta T} e^{\mathbf{A}_{LoS} \cdot \tau} \cdot \mathbf{n}_{LoS}(\tau) \cdot \mathbf{n}_{LoS}^T(\xi) \cdot \delta(\tau - \xi) \cdot \left(e^{\mathbf{A}_{LoS} \cdot \xi} \right)^T d\xi d\tau \right\} \\ &= \int_{\Delta T} e^{\mathbf{A}_{LoS} \cdot (\tau)} \cdot \mathcal{E} \left\{ \mathbf{n}_{LoS}(\tau) \cdot \mathbf{n}_{LoS}^T(\tau) \right\} \cdot \left(e^{\mathbf{A}_{LoS} \cdot (\xi)} \right)^T d\tau = \mathbf{Q} \cdot \int_{\Delta T} e^{\mathbf{A}_{LoS} \cdot (\tau)} \cdot \left(e^{\mathbf{A}_{LoS} \cdot (\xi)} \right)^T d\tau\end{aligned}\quad (III-70)$$

Using the approximation from equation (III-69), the discrete error covariance matrix gets:

$$\mathbf{Q}_k = \mathbf{Q} \cdot \int_{\Delta T} \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & t \\ 0 & 0 & 1 - \frac{t}{\tau_{a,LoS}} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ t & 1 & 0 \\ 0 & t & 1 - \frac{t}{\tau_{a,LoS}} \end{bmatrix} d\tau = \mathbf{Q} \cdot \int_{\Delta T} \begin{bmatrix} 1+t^2 & t & 0 \\ t & 1+t^2 & t - \frac{t^2}{\tau_{a,LoS}} \\ 0 & t - \frac{t^2}{\tau_{a,LoS}} & \left(1 - \frac{t}{\tau_{a,LoS}}\right)^2 \end{bmatrix} d\tau = .. \quad (III-71)$$

$$\begin{aligned} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{n,aLoS}^2 \end{bmatrix} \cdot \begin{bmatrix} \Delta T + \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} & 0 \\ \frac{\Delta T^2}{2} & \Delta T + \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} - \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}} \\ 0 & \frac{\Delta T^2}{2} - \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}} & \Delta T - \frac{\Delta T^2}{\tau_{a,LoS}} + \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}^2} \end{bmatrix} = \\ \mathbf{Q}_k &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sigma_{n,aLoS}^2 \cdot \left(\frac{\Delta T^2}{2} - \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}} \right) & \sigma_{n,aLoS}^2 \cdot \left(\Delta T - \frac{\Delta T^2}{\tau_{a,LoS}} + \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}^2} \right) \end{bmatrix} \quad (III-72)\end{aligned}$$

If the assumed error variance $\sigma_{n,aLoS}^2$ is large, the Kalman filter pays more attention to the

measurement and gives only little trust to the model based predicted line of sight states. This in turn leads inherently to higher Kalman gains - which are necessary to put more weight on the measurement. These higher Kalman gains increase the overall closed loop tracking bandwidth and help the tracking filter to follow high line of sight dynamic. It is comparable to a gain increase in a classical PID loop filter-based tracking loop.

The red colored parameters show the direct link between the Kalman filter tuning parameters and mission characteristics, as it will be shown in the section on "scenario matched tuning".

Noise like measurement error

The noise like part of the tracking error $n_{\delta r}$, is a result of the received noise at the antenna together with some minor noise sources, like additional amplifier noise within the analog electronics stage of the receiver and the quantization noise.

The variance $\sigma_{n,\delta r}^2$ of this noise can be calculated as a function of the signal to noise ratio and jamming to signal ration, which will be derived in detail in a later chapter. The Kalman filter considers this noise like error within the measurement error variance matrix.

$$\mathbf{R}_{DLL} = \sigma_{n,\delta r}^2 \quad (\text{III-73})$$

Important to know, only if this noise like measurement error is white, the Kalman filter is an optimal filter. If this error is colored, whitening measures must be implemented.

Clock error

In this scalar tracking approach, there is no possibility to estimate the clock error based on geometrical relations. If the state dynamic model without clock error will be used, the clock error is mapped onto the estimated pseudorange.

The only alternative would be to model the clock error as a consider state, by using the following dynamic model and observation matrix.

$$\begin{bmatrix} \Delta \dot{\rho} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \\ c \cdot \Delta \dot{t}_{clk} \\ c \cdot \Delta \dot{d}_{clk} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{a,LoS}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \\ c \cdot \Delta t_{clk} \\ c \cdot \Delta d_{clk} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \\ n_{\Delta tclk} \\ n_{\Delta dclk} \end{bmatrix} \quad (\text{III-74})$$

$$\mathbf{h}^T = \begin{bmatrix} 1 \\ \frac{1}{c} & 0 & 0 & \frac{1}{c} & 0 \end{bmatrix} \quad (\text{III-75})$$

Modeling the clock error as a consider state which is mapped onto the measurement, makes the measurement more uncertain in the frequency range of the clock error.

In high dynamic scenarios, where the line of sight dynamic has higher frequencies than the clock error dynamic, there is no benefit of this approach. Only in low dynamic or even static scenarios, this approach might improve the performance.

III - 3.4 Novel scenario matched adaptive tuning

In the chapter on classical scalar tracking loops, the main focus of tuning was the closed loop tracking bandwidth and also the avoidance of remaining tracking errors due to a wrong loop filter order. There is no direct link between the line of sight dynamic or the real signal to noise ratios and the optimal closed loop tracking bandwidth and filter order.

Especially in case of pre known flight trajectories, like approaches to airports or planned missions in case of military operations, the line of sight dynamic is well known in advance. Also, assumptions on the received signal to noise ratio can be made or even some worst case boundaries on the received noise power can be estimated very good in advance.

Classical scalar tracking loops do not benefit from this information.

Optimal scalar tracking loops however use all this information directly for tuning. This information can directly be assigned to the available tuning parameters of the scalar optimal filter tracking architecture.

Selection of an appropriate line of sight dynamic model

At first, the line of sight dynamic model, used for state propagation in the Kalman filter, has to be selected. This model should correspond to the real line of sight dynamic of the platform. In [36], a line of sight dynamic model for radar-based flight object tracking is suggested. This dynamic model is also used to describe the line of sight dynamic between the DA42 or the fighter and the respective satellite.

Tuning the model error covariance matrix

As it was already mentioned in chapter III - 3.3 , the line of sight dynamic model assumes that the line of sight acceleration behaves like a zero mean Gauss Markov process. In order to describe this acceleration error process, respectively its dynamic behavior as a Gauss Markov process, the variance and the correlation time constant of the processes must be determined.

Both parameters can be found in the autocorrelation of the gauss Markov process, as the following equation shows [38, p. 33].

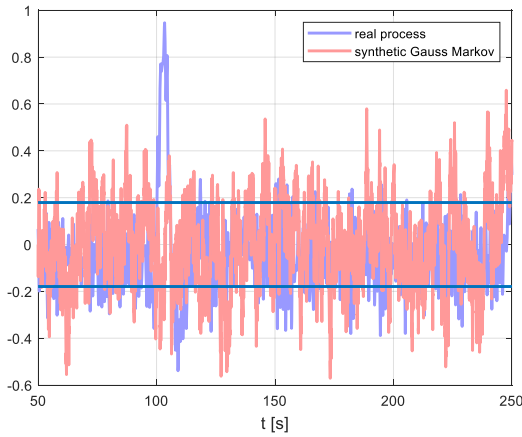
$$R_{xx}(t) = \sigma_{a,LoS}^2 \cdot e^{\frac{-|t|}{\tau_{a,LoS}}} \quad (\text{III-76})$$

The mean of the acceleration is assumed to be zero $\mathcal{E}\{\mathbf{a}_{LoS}\} = \mathbf{0}$.

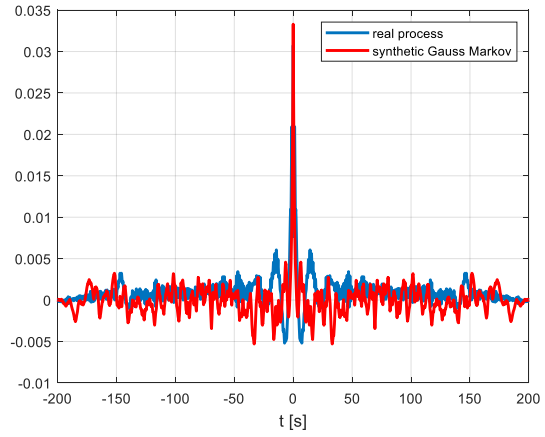
In order to fit the line of sight dynamic model to the real line of sight dynamic, the variance and correlation time constant of the real acceleration error process are determined.

In the following figures, an example acceleration process and the synthetic Gauss Markov model are shown. The left figure shows the process in time space. The right figure shows the correlation

of the real process and the synthetic one.

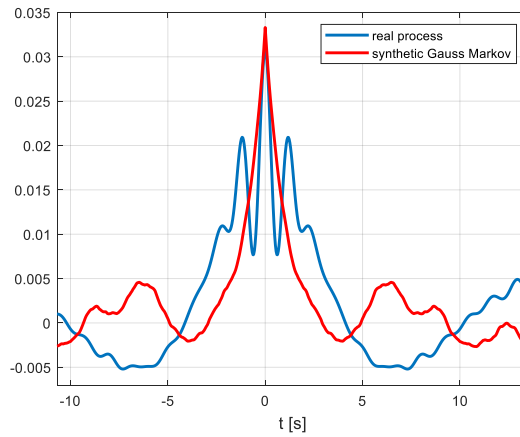


III-29 LoS acceleration error process - DA42 approach



III-30 LoS acceleration error correlation – DA42 approach

In order to show that the synthetically generated Gauss Markov process has the same correlation time constant as the real process, the following figure shows a zoomed version of the correlation.



III-31 LoS acceleration error correlation - zoomed

A detailed derivation of the Gauss Markov process can be found in Appendix VII - 1.2 .

Knowing the Gauss Markov process parameters, the necessary variance of the seed noise gets according to (VII-12) as

$$\sigma_{n,aLoS}^2 = \frac{\sigma_{a,LoS}^2 \cdot 2}{\tau_{a,LoS}} \quad (\text{III-77})$$

With the knowledge of the parameters and the already introduced clock error model, the model error covariance matrix for the scalar total state optimal tracking gets

$$\mathbf{w} = \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \end{bmatrix}, \quad \mathbf{Q} = \mathcal{E}\{\mathbf{w} \cdot \mathbf{w}^T\} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{a,LoS}^2}{\tau_{a,LoS}} \end{bmatrix} \quad (\text{III-78})$$

If the clock error is considered within the scalar tracking loop, the error covariance gets

$$\mathbf{w} = \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \\ n_{\delta t,k} \\ n_{\delta f,k} \end{bmatrix}, \quad \mathbf{Q} = \mathcal{E}\{\mathbf{w} \cdot \mathbf{w}^T\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{a,LoS}^2}{\tau_{a,LoS}} & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\delta t}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\delta f}^2 \end{bmatrix} \quad (\text{III-79})$$

Considering (III-72), the discrete model error covariance matrix can be written as following.

$$\mathbf{Q}_k = \mathcal{E}\{\mathbf{w}_k \cdot \mathbf{w}_k^T\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{n,aLoS}^2 \cdot \left(\frac{\Delta T^2}{2} - \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}} \right) & \sigma_{n,aLoS}^2 \cdot \left(\Delta T - \frac{\Delta T^2}{\tau_{a,LoS}} + \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}^2} \right) & 0 & 0 \\ 0 & 0 & 0 & \Delta T \cdot \sigma_{\delta t}^2 & 0 \\ 0 & 0 & 0 & 0 & \Delta T \cdot \sigma_{\delta f}^2 \end{bmatrix} \quad (\text{III-80})$$

It is assumed, that the clock error is not correlated with the line of sight dynamic error. The time ΔT , at which the Kalman filter is iterated, should correspond to the correlation time T_{corr} .

Tuning the measurement error variance matrix

Like it was already mentioned, the noise like measurement error variance is a function of the signal to noise ratio, the used discriminator type, the early minus late spacing within the correlation core and also the correlation time. The measurement error variance can be calculated analytically using the following equation. In this example, an early minus late power discriminator is used [2].

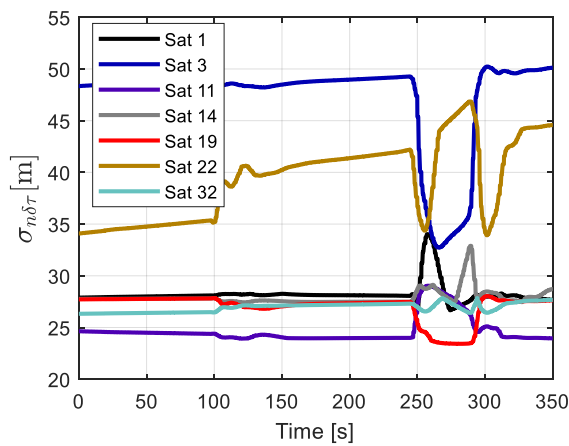
$$\sigma_{n,\delta r}^2 = \frac{T_{CAchip}^2}{4 \cdot T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}} \right)} \cdot \left(1 + \frac{2}{T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}} \right)} \right) \quad (\text{III-81})$$

The signal to noise ratio at IF interface can be predicted using models which will be derived in chapter IV - 2, where the whole line of sight characteristic, possible jammer constellations, the whole receiver analog electronic and also the antenna reception pattern will be considered.

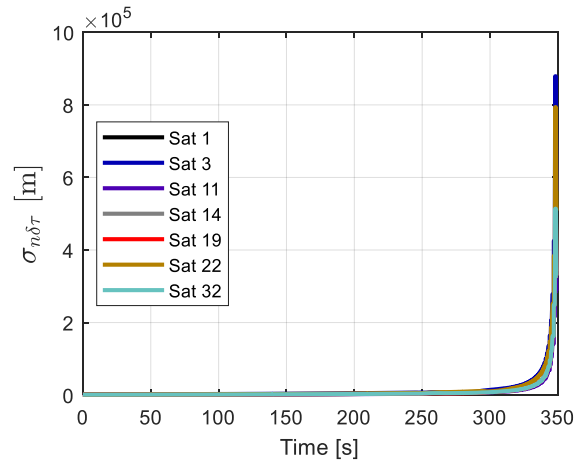
Another possibility is the online measurement of the signal to noise ratio with an instantaneous adaption of the measurement error variance within flight.

For tuning, the DA42 approach and the fighter trajectory will be used. For both trajectories, the measurement error variance $\sigma_{n,\delta r}^2(t, \mathbf{x}(t))$ will be predicted using the model based C/N prediction, introduced in chapter IV - 2.5 .

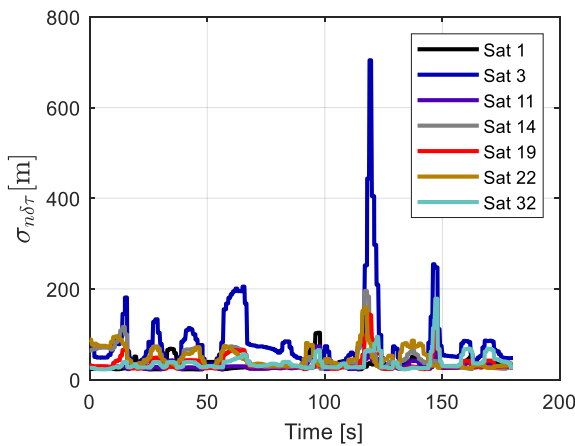
The following two figures show the measurement error variance at the discriminator output, given the DA42 approach and the high dynamic trajectory. (The examples show the unjammed and jammed error variance for the satellites in view, while being on mission)



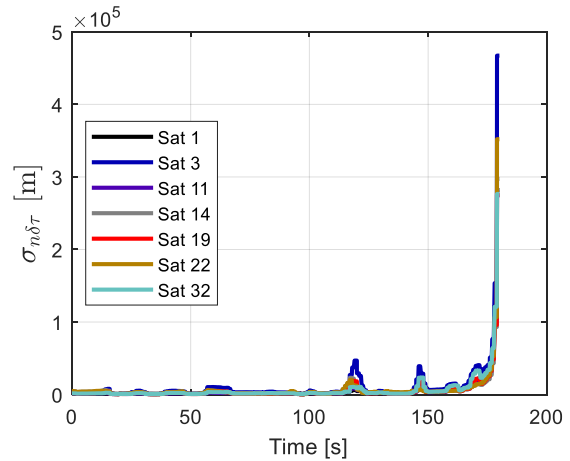
III-32 Discr. error variance - DA42 - unjammed



III-33 Discr. error variance - DA42 - jammed



III-34 Discr. error variance - high dynamic - unjammed



III-35 Discr. error variance - high dynamic - jammed

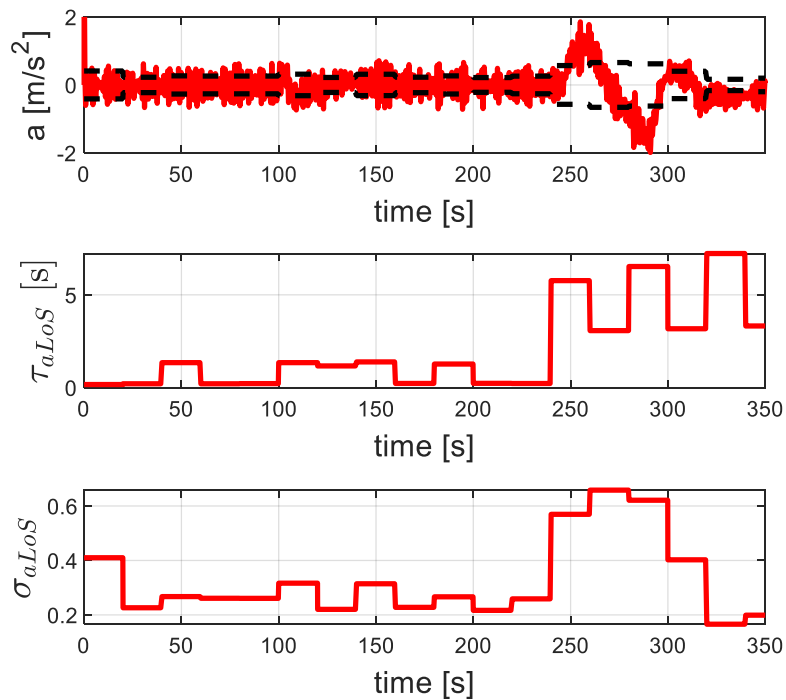
As can be seen from figures III-32 to III-35, the measurement error variance at the output of the discriminator, changes along mission. For optimal tuning, this time dependent error variance has to be considered within the measurement error variance matrix \mathbf{R}_{DLL} of the Kalman filter. As a result, \mathbf{R}_{DLL} will become time dependent.

Online adaption of line of sight dynamic model

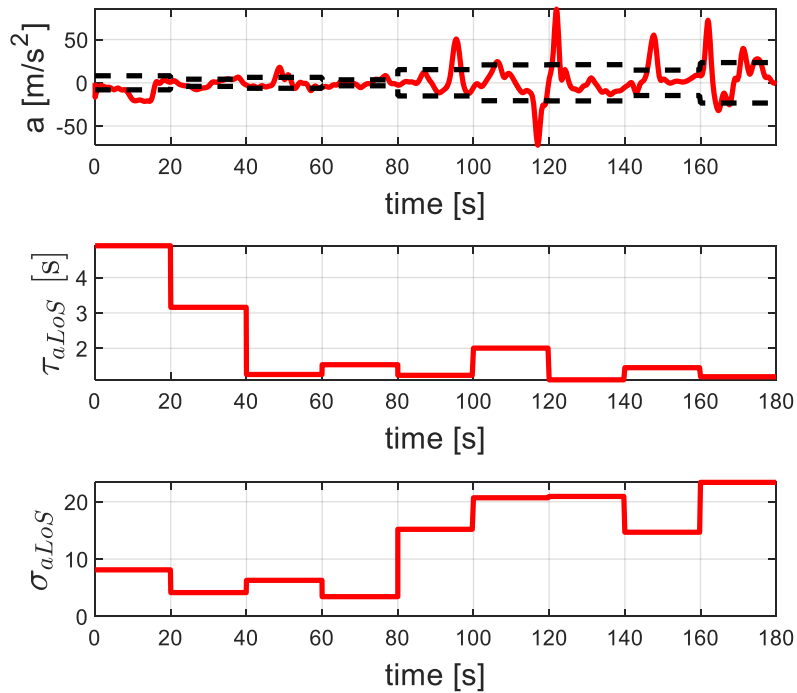
Besides the measurement error variance, also the line of sight acceleration varies over time. Up to now, for tuning, the acceleration was assumed having a zero mean. The deviation from this mean value was modeled as a Gauss Markov process 1st order.

The variance was calculated by considering the acceleration behavior along the whole mission. A better approach is to calculate the variance of the acceleration on many small sections along the mission trajectory.

The following figures show as an example this procedure for satellite 19, given the DA42 approach and the high dynamic fighter trajectory.



III-36 Adaptive tuning parameter DA42 approach



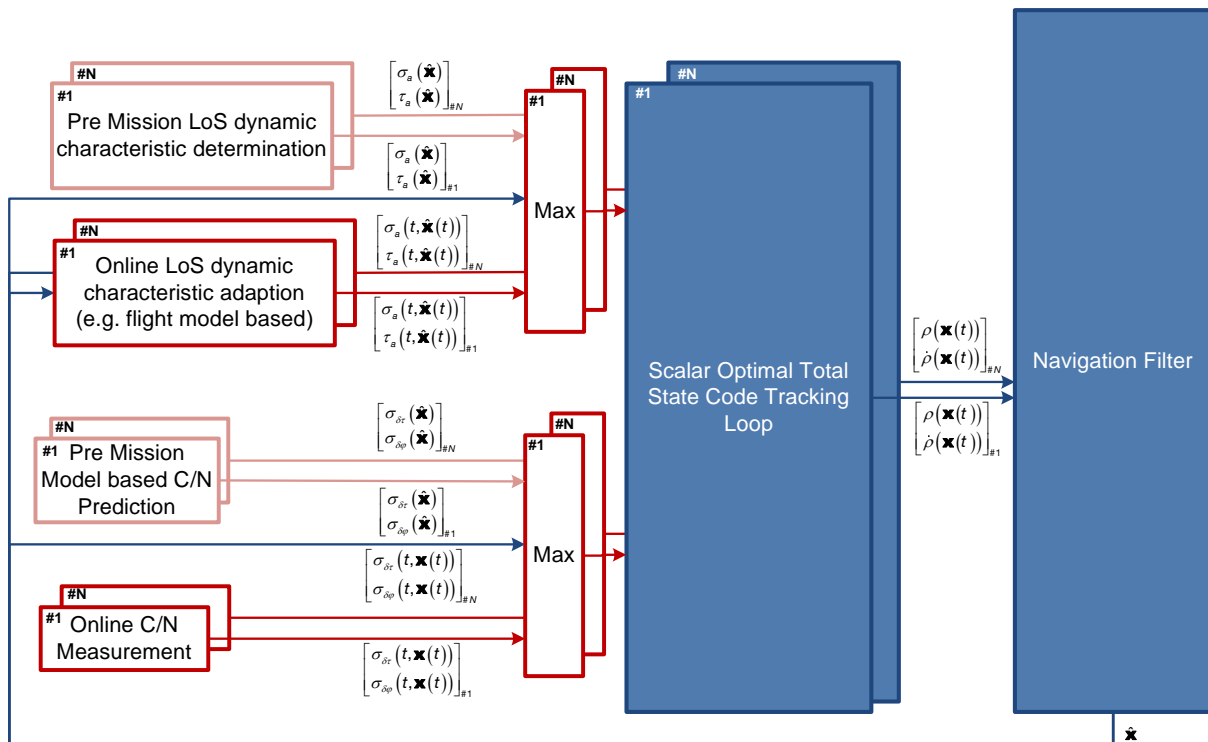
III-37 Adaptive tuning parameter Fighter

The selection of the right section length is an important criterion, to find the optimal parameter for loop design. In this example the section time was figured out by manual search, which is possible because the mission is already known. In other cases, there will be the possibility to a flight model based tuning approach. The flight computer control commands could be used to predict the short time dynamic change of the aircraft, using a flight model. The predicted dynamic changes could be used in turn for LoS variance calculations.

This variable acceleration variance is used within the model error covariance matrix \mathbf{Q} , which now becomes also time dependent.

In case of fixed tuning parameters, the Kalman filter becomes steady state, after an initial settling phase. In case of variable tuning parameter, there will be no steady state Kalman gain.

The following architecture proposes a concept for realization of such an online adaptive tuning scheme. The tuning figures of the Kalman filter within the scalar optimal total state code tracking loop are changed online. Because the Kalman filter acts within the tracking loop as a loop controller, the following architecture represents an optimal adaptive control loop.



III-38 Architecture for online adaptive tuning

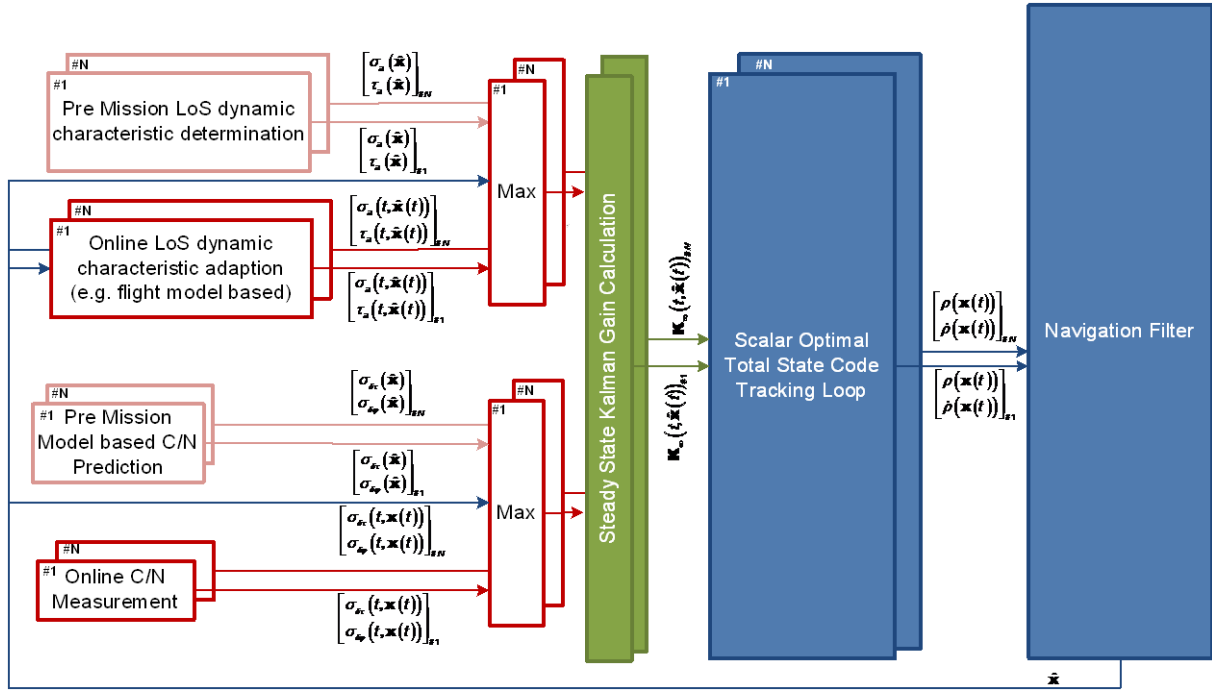
At first, pre mission, the line of sight dynamic tuning parameters are calculated, respectively predicted. Additionally, while being on the mission, the tuning parameters are also determined online, based on actual mission parameters.

The measurement error variances are determined from online measurements of signal to noise ratio. The acceleration variances and correlation times can be adapted by online acceleration measurements, or can be predicted, based on actual flight control commands, by using an aerodynamic model of the aircraft.

Scenario matched online adaptive tuning - steady state

In case of the mentioned online adaption of model dynamic and model error covariance matrix \mathbf{Q} , as well as the online adaption of the measurement error variance, the Kalman filter almost never reaches steady state. Especially in high dynamic scenarios, the tuning parameters are adapted all the time, leaving the Kalman filter always in a settling mode. The optimal Kalman gains, being necessary corresponding to the line of sight dynamic and the actual measurement error variance at the valid at distinct point on trajectory, will not be reached.

The following tuning architecture suggests a method to get the optimal settled Kalman gains on each point on trajectory, valid for the actual line of sight dynamic and actual measurement error variance.



III-39 Architecture for online adaptive tuning – steady state Kalman gain initialization

At every time step, at first the actual steady state Kalman gains are calculated by solving the Matrix Riccati differential equation.

Starting with the discrete propagation of the state error covariance equation according to [40, p. 301]

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k \Phi_k^T + \Phi_k^T \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (\text{III-82})$$

and replacing the Kalman gain by the corresponding Kalman gain update equation

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k^T \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (\text{III-83})$$

the discrete state error covariance equation gets

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k \Phi_k^T + \Phi_k^T \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k^T \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (\text{III-84})$$

The steady state solution is reached, if the state error covariance does not change from time step k to time step $k+1$. Using this assumption, the state error covariance \mathbf{P}_{k+1}^- can be replaced by \mathbf{P}_k^- .

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_k \Phi_k^T + \Phi_k^T \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k^T \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (\text{III-85})$$

This form is also called the discrete algebraic Riccati equation (DARE).

There are several approaches for solving the DARE, like given in [41]. In the following, the DARE is solved by first converting it into time continuous form and calculating the continuous steady state solution according to [33, p. 246].

$$\dot{\mathbf{P}} = -\mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{P} + \mathbf{A} \mathbf{P} + \mathbf{P} \mathbf{A}^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T \quad (\text{III-86})$$

And the steady state solution is given by solving the continuous algebraic Riccati equation (CARE).

$$\mathbf{0} = -\mathbf{P}_\infty \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{P}_\infty + \mathbf{A} \mathbf{P}_\infty + \mathbf{P}_\infty \mathbf{A}^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T \quad (\text{III-87})$$

Important to notice is the difference between the discrete and continuous versions of the following matrices.

$$\mathbf{Q}_k = \mathbf{Q} \cdot \Delta T, \quad \mathbf{R}_k = \frac{1}{\Delta T} \cdot \mathbf{R} \quad (\text{III-88})$$

According to [33, p. 246] and [42], the state error covariance matrix can be decomposed as

$$\mathbf{P}(t) = \mathbf{N}(t) \mathbf{D}(t)^{-1} \quad (\text{III-89})$$

Using (III-90), the continuous matrix Riccati differential can be written according to the next matrix differential equation.

$$\begin{bmatrix} \dot{\mathbf{N}}(t) \\ \dot{\mathbf{D}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \mathbf{Q} \mathbf{G}^T \\ \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} & -\mathbf{A}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{N}(t) \\ \mathbf{D}(t) \end{bmatrix} = \boldsymbol{\Psi} \cdot \begin{bmatrix} \mathbf{N}(t) \\ \mathbf{D}(t) \end{bmatrix}, \quad \begin{bmatrix} \mathbf{N}_0 \\ \mathbf{D}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{I} \end{bmatrix} \quad (\text{III-90})$$

The general solution of this matrix differential equation is

$$\begin{bmatrix} \mathbf{N}(t) \\ \mathbf{D}(t) \end{bmatrix} = \mathbf{e}^{\boldsymbol{\Psi}(t-t_0)} \cdot \begin{bmatrix} \mathbf{N}_0 \\ \mathbf{D}_0 \end{bmatrix} \quad (\text{III-91})$$

For solution, the matrix $\boldsymbol{\Psi}$ is written in eigenspace representation.

$$\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Lambda} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}^{-1} \quad (\text{III-92})$$

With this eigenspace approach, the solution gets

$$\begin{bmatrix} \mathbf{N}(t) \\ \mathbf{D}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}^{\boldsymbol{\Lambda}(t-t_0)} & \mathbf{0} \\ \mathbf{0} & \mathbf{e}^{-\boldsymbol{\Lambda}(t-t_0)} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{N}_0 \\ \mathbf{D}_0 \end{bmatrix} \quad (\text{III-93})$$

The eigenvalues with $\Re\{\lambda\} > 0$ are collected in $\boldsymbol{\Lambda}$, whereby $-\boldsymbol{\Lambda}$ contains the eigenvalues with a negative real part. The eigenvectors are rearranged so that

$$\begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{V}_{21} \end{bmatrix} = [\mathbf{v}_{\lambda_1}, \dots, \mathbf{v}_{\lambda_n}] \quad (\text{III-94})$$

contain the eigenvectors belonging to the eigenvalues with $\Re\{\lambda\} > 0$.

The steady state solution is reached by letting $t \rightarrow \infty$ in equation (III-93). After some algebra, which is explained in detail in [33, p. 248], the steady state error covariance matrix gets

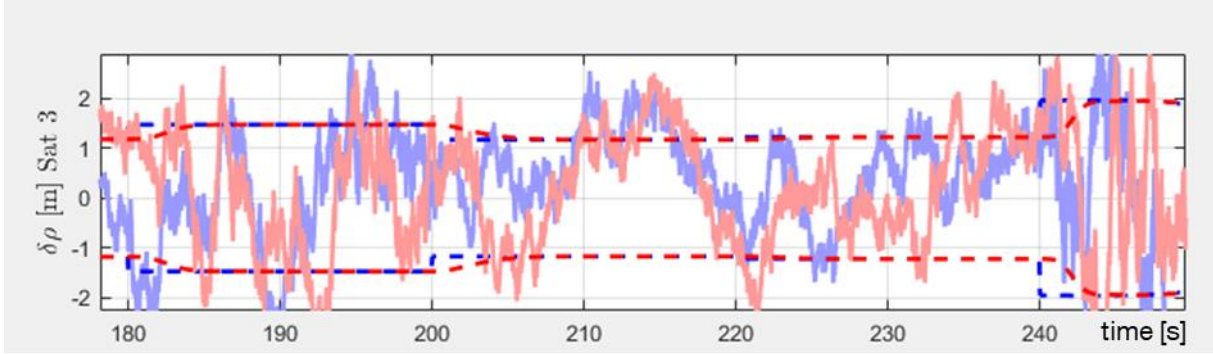
$$\mathbf{P}_\infty = \mathbf{V}_{11} \mathbf{V}_{21}^{-1} \quad (\text{III-95})$$

The corresponding steady state Kalman gain gets

$$\mathbf{K}_\infty = \mathbf{P}_\infty \mathbf{H}^T (\mathbf{H}^T \mathbf{P}_\infty \mathbf{H}^T + \mathbf{R})^{-1} \quad (\text{III-96})$$

The tracking behavior comparison between this steady state calculated Kalman gain and the normal settling Kalman gain, is given at the end of this chapter. The following figure shall only give a first

impression for the different tracking behaviors. The figure is a zoomed snapshot of the tracking behavior simulation, given the DA42 approach. The blue line represents the behavior in case of steady state Kalman gain calculation. The red line represents the behavior in case of free settling. Considering the standard deviation, it gets obvious that after every change of some tuning figures, the steady state standard deviation changes promptly. The other standard deviation becomes the same value as the steady state one, after some settling time.



III-40 Comparing steady state and free settling Kalman gain tracking behavior

III - 3.5 Equivalent base band dynamic state space simulation

III - 3.5.1 Distributed state space model

In figure III-28, the tracking architecture of the unaided optimal filter-based code tracking loop in total state form, is shown.

For pre mission tracking simulation, in this section, an equivalent base band realization of this tracking loop will be derived.

In the first step, the correlation core as well as the loop filter will be described in separated, but coupled state space representations.

The feedback from loop filter to correlation core is realized as direct phase control. The corresponding correlation core model gets:

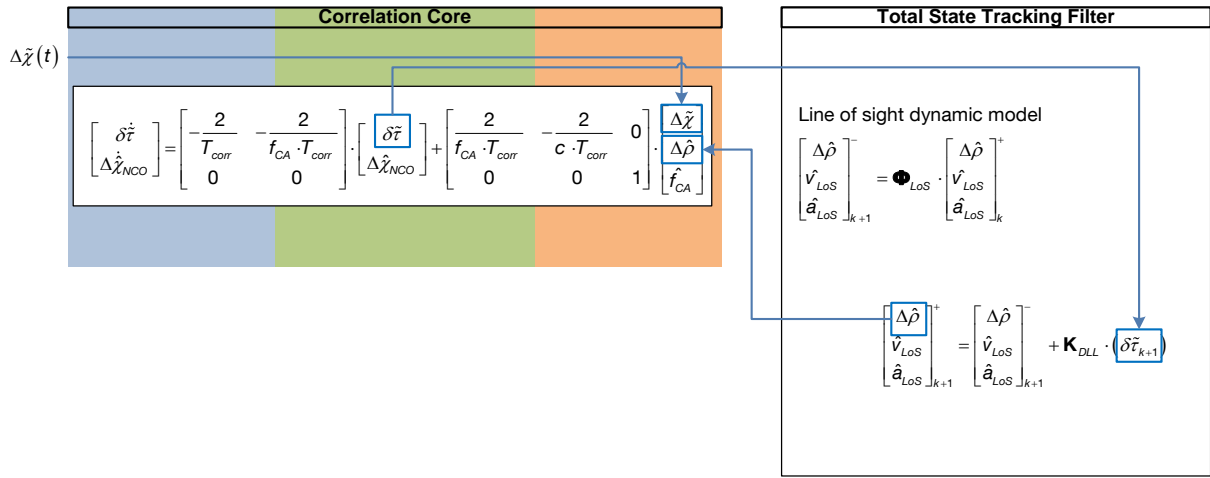
$$\begin{bmatrix} \dot{\delta\tilde{\tau}} \\ \Delta\dot{\hat{\chi}}_{NCO} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2}{f_{CA} \cdot T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi}_{NCO} \end{bmatrix} + \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} & -\frac{2}{c \cdot T_{corr}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta\tilde{\chi}_{SV} \\ \Delta\hat{\rho} \\ \hat{f}_{CA} \end{bmatrix} \quad (III-97)$$

The second component of the distributed state space realization is the loop filter itself.

For equivalent base band simulation, the loop filter – which is a Kalman filter – can be implemented by using the normal Kalman filter equations.

$$\begin{aligned}
& \text{state propagation: } \mathbf{z}_{LF,k+1}^- = \Phi_{LF} \cdot \mathbf{z}_{LF,k}^- \\
& \text{state error covariance propagation: } \mathbf{P}_{LF,k+1}^- = \Phi_{LF} \mathbf{P}_{LF,k}^+ \Phi_{LF}^T + \mathbf{Q}_{LF} \\
& \text{Kalman gain calculation: } \mathbf{K}_{LF,k+1} = \mathbf{P}_{LF,k+1}^- \mathbf{H}_{LF}^T \cdot (\mathbf{H}_{LF} \mathbf{P}_{LF,k+1}^- \mathbf{H}_{LF}^T + \mathbf{R}_{LF})^{-1} \quad (\text{III-98}) \\
& \text{state update: } \mathbf{z}_{LF,k+1}^+ = \mathbf{z}_{LF,k+1}^- + \mathbf{K}_{LF,k+1} \cdot (\delta\tau_{LF,k+1}) \\
& \text{state error covariance update: } \mathbf{P}_{LF,k+1}^+ = (\mathbf{I} - \mathbf{K}_{LF,k+1} \mathbf{H}_{LF}) \cdot \mathbf{P}_{LF,k+1}^-
\end{aligned}$$

The following figure shows the distributed state space representation, together with corresponding couplings between the two components. Within the tracking filter component, only the relevant equations for coupling are shown. It is important to keep in mind, within the tracking filter component, all equations according to (III-98) are calculated at every time step.



III-41 Distributed state space representation

In case of non-adaptive optimal gain tracking, the matrices Φ_{LF} , \mathbf{R}_{LF} and \mathbf{Q}_{LF} are independent of time. If adaptive tracking is used, these three matrices become time dependent $\Phi_{LF}(t)$, $\mathbf{R}_{LF}(t)$ and $\mathbf{Q}_{LF}(t)$. These three matrices are adapted, based on actual line of sight tuning parameters according to figure III-39.

The equivalent base band input code phase dynamic $\Delta\tilde{\chi}(t)$ is generated, based on the line of sight Doppler frequency and corresponds the real code phase progress, a real tracking loop would receive. A detailed derivation of equivalent base band code phase dynamic is given in IV - 2.8 .

III - 3.5.2 Error stimulation within the distributed state space form

Two errors are considered for this tracking loop.

The first error being considered is the code phase error measurement noise $n_{\delta\tau}$. This error component is caused by the received noise and antenna noise.

This error component is stimulated directly at the output of the discriminator by adding the corresponding code phase noise to the discriminator output.

The other major error component is the clock error. A detailed derivation of the following clock

error model is given in appendix VII - 1.1 .

$$\begin{bmatrix} \delta \dot{t}_{clk} \\ \dot{d}_{clk} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta t_{clk} \\ d_{clk} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \eta_{\delta t} \\ \eta_{\delta d} \end{bmatrix} \quad (III-99)$$

The clock error is basically a result of a drifting base frequency of the oscillator within the GPS receiver and therefore a frequency error of the NCO used for generating the code repique.

$$\delta \chi_{clk} = \int \delta f_{clk}(t) dt \quad (III-100)$$

Given the clock error, the corresponding NCO frequency error can be calculated according to following equation.

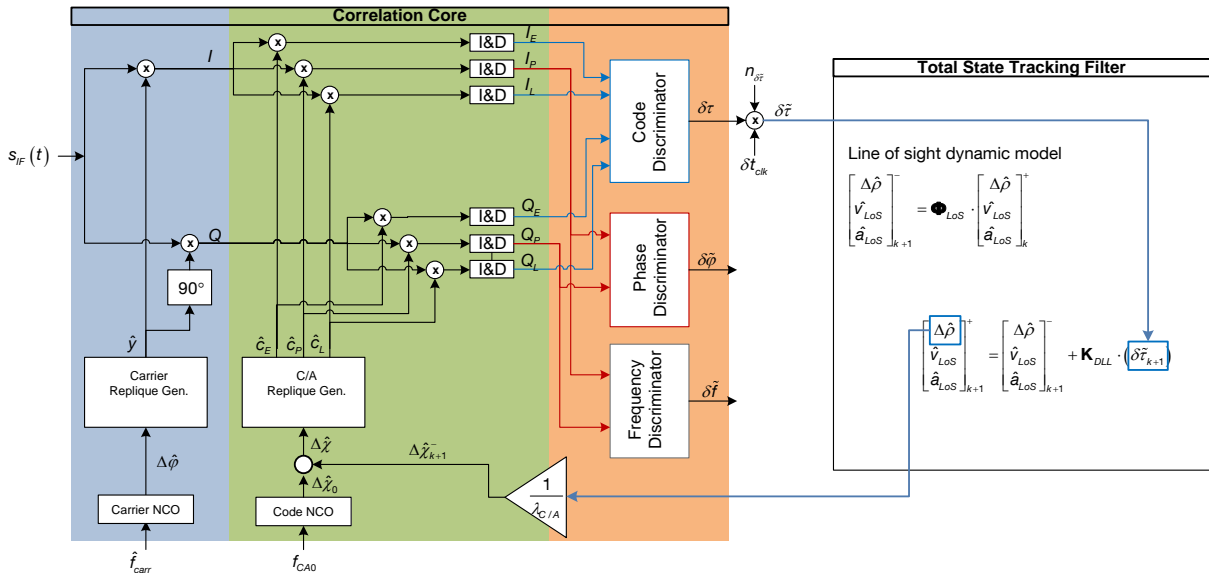
$$\delta f_{clk} = \delta \dot{\chi}_{clk} = \frac{\delta \dot{t}_{clk}}{T_{C/A}} = \frac{\delta t_{clk}(k) - \delta t_{clk}(k-1)}{T_{corr} \cdot T_{C/A}} \quad (III-101)$$

For stimulation the clock error within the equivalent base band simulation, there are two possibilities.

The first one is, adding the calculated frequency error to the NCO frequency command according to next equation.

$$\hat{f}_{C/A} = f_{C/A,0} + \Delta f_{Doppler} + \delta f_{clk} \quad (III-102)$$

The other option is, adding the clock error δt_{clk} directly to the discriminator measurement, as shown in figure III-42.



III-42 Error stimulation - DLL

III - 3.5.3 Centralized state space form

In the previous chapter, the state space representation for the tracking loop was derived, whereby the correlation core state space model and the loop filter state space model were handled as two separate state space forms.

For calculation of the closed loop tracking bandwidth or the eigenvalue-based tracking stability analysis, a centralized state space representation of the closed loop tracking loop is necessary.

For this purpose, the separated state space model is written in a centralized closed loop state space matrix. The couplings between the two separated state space forms are realized within the system matrix.

The following equation shows the centralized state space representation of the scalar optimal total state tracking architecture.

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{u}$$

$$\begin{bmatrix} \delta \dot{\tau} \\ \Delta \dot{\hat{\chi}}_{NCO} \\ \Delta \dot{\hat{\rho}} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2}{f_{CA} \cdot T_{corr}} & -\frac{2}{c \cdot T_{corr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{k_{11}(t)}{T_{corr}} & 0 & 0 & 1 & 0 \\ \frac{k_{21}(t)}{T_{corr}} & 0 & 0 & 0 & 1 \\ \frac{k_{31}(t)}{T_{corr}} & 0 & 0 & 0 & -\frac{1}{\tau_{aLoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi}$$
(III-103)

The Kalman gains need to be normed by correlation time respectively the rate by which the discrete version of the matrix differential equation is solved. The reason becomes clear by considering the discrete version of equation (III-103). For the discrete version it is assumed, that the integration period equals the correlation time.

$$\mathbf{z}_{k+1} = (\mathbf{I} + \mathbf{A} \cdot T_{corr}) \cdot \mathbf{z}_k + \mathbf{B} \cdot T_{corr} \cdot \mathbf{u}_k$$

$$\begin{bmatrix} \delta \tilde{\tau} \\ \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix}_{k+1} = \begin{bmatrix} -1 & \frac{-2}{f_{CA}} & -\frac{2}{c} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ k_{11,k} & 0 & 1 & T_{corr} & 0 \\ k_{21,k} & 0 & 0 & 1 & 0 \\ k_{31,k} & 0 & 0 & 0 & 1 - \frac{T_{corr}}{\tau_{aLoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix}_k + \begin{bmatrix} \frac{2}{f_{CA}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi}_k$$
(III-104)

Within the update equation of the Kalman filter, there is no multiplication of the Kalman gains with the sample time. By solving the matrix differential equation (III-103) in discrete form, the Kalman

gains will be multiplied by the correlation time. To avoid this multiplication, the Kalman gains need to be divided by the correlation time.

Using the closed loop state space matrices introduced so far, the noise free pseudorange behavior can be evaluated and also the dynamic caused tracking errors.

Error stimulation in centralized state space form

In order to evaluate the pseudorange error variance being caused by the received noise, the noise has to be considered within the centralized state space matrix. Also, the clock error can be stimulated.

The following matrix differential equation shows the noise and clock error stimulation within the centralized state space form. The clock error states are implemented as consider states within the state propagation of the Kalman filter. For state error covariance propagation, these consider states are added to the measurement, using the following transformation

$$\mathbf{z} = \begin{bmatrix} \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \\ \delta \hat{t}_{clk} \\ \delta \hat{f}_{clk} \end{bmatrix}, \quad \mathbf{h}_{DLL}^T = \begin{bmatrix} \frac{1}{c} & 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{III-105})$$

The real NCO control, derived from the Kalman filter state vector, does not include this consider state.

$$\Delta \hat{\tau} = \begin{bmatrix} \frac{1}{c} & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \\ \delta \hat{t}_{clk} \\ \delta \hat{f}_{clk} \end{bmatrix} \quad (\text{III-106})$$

Therefore, the consider states do not appear within the closed loop steady state matrix, even if the state propagation model of the corresponding Kalman filter is extended by the consider states. The effect of the consider states is already included in the Kalman gains, used in the closed loop state space matrix.

For simulation, additionally to the received equivalent base band code phase progress $\Delta \tilde{\chi}$, also the noise and clock errors must be considered within the steady state space representation according to the following equation.

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{u} + \mathbf{G} \cdot \mathbf{n} = \mathbf{A} \cdot \mathbf{z} + \mathbf{b}_{\Delta \tilde{\chi}} \cdot \Delta \tilde{\chi} + \begin{bmatrix} \mathbf{g}_{n_{\delta \tau}} & \mathbf{g}_{\delta t, clk} \end{bmatrix} \cdot \begin{bmatrix} n_{\delta \tau} \\ \delta t, clk \end{bmatrix} \quad (\text{III-107})$$

$$\begin{bmatrix} \delta \dot{\tau} \\ \Delta \dot{\hat{\chi}}_{NCO} \\ \Delta \dot{\hat{\rho}} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2}{f_{CA} \cdot T_{corr}} & -\frac{2}{c \cdot T_{corr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{k_{11}}{T_{corr}} & 0 & 0 & 1 & 0 \\ \frac{k_{21}}{T_{corr}} & 0 & 0 & 0 & 1 \\ \frac{k_{31}}{T_{corr}} & 0 & 0 & 0 & -\frac{1}{\tau_{aLoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} \dots$$

$$+ \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{k_{11}}{T_{corr}} & \frac{k_{11}}{T_{corr}} \\ \frac{k_{21}}{T_{corr}} & \frac{k_{21}}{T_{corr}} \\ \frac{k_{31}}{T_{corr}} & \frac{k_{31}}{T_{corr}} \end{bmatrix} \cdot \begin{bmatrix} n_{\delta \tau} \\ \delta t, clk \end{bmatrix}$$

(III-108)

The centralized state space approach offers now the possibility to simulate the tracking behavior pre mission, like the distributed state space simulation. At this point, there is one major difference regarding the Kalman gains. In case of the distributed state space simulation, the Kalman gains are calculated online in the context of the step by step execution of the Kalman filter equations. The centralized state space form does not calculate the Kalman gains. The Kalman gains must be calculated in advance by using the distributed state space form.

But what is more important, this closed loop state space matrices can be used, to calculate the closed loop tracking bandwidth and the closed loop tracking eigenvalues.

The closed loop tracking bandwidth is a SISO bandwidth of a defined path through the given centralized state space system. The following SISO bandwidths are of interest

$$\Delta \tilde{\chi} \rightarrow \Delta \hat{\rho} : \quad \mathbf{G}_{\Delta \rho} = \mathbf{c}_{\Delta \rho}^T \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}_{\Delta \tilde{\chi}}$$

$$\Delta \tilde{\chi} \rightarrow \hat{v}_{LoS} : \quad \mathbf{G}_{\hat{v}_{LoS}} = \mathbf{c}_{\hat{v}_{LoS}}^T \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}_{\Delta \tilde{\chi}}$$

$$\mathbf{c}_{\Delta \rho}^T = [0 \quad 0 \quad 1 \quad 0 \quad 0]$$

$$\mathbf{c}_{\hat{v}_{LoS}}^T = [0 \quad 0 \quad 0 \quad 1 \quad 0]$$

$$\mathbf{b}_{\Delta \tilde{\chi}}^T = \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} & 0 & 0 & 0 & 0 \end{bmatrix}$$

(III-109)

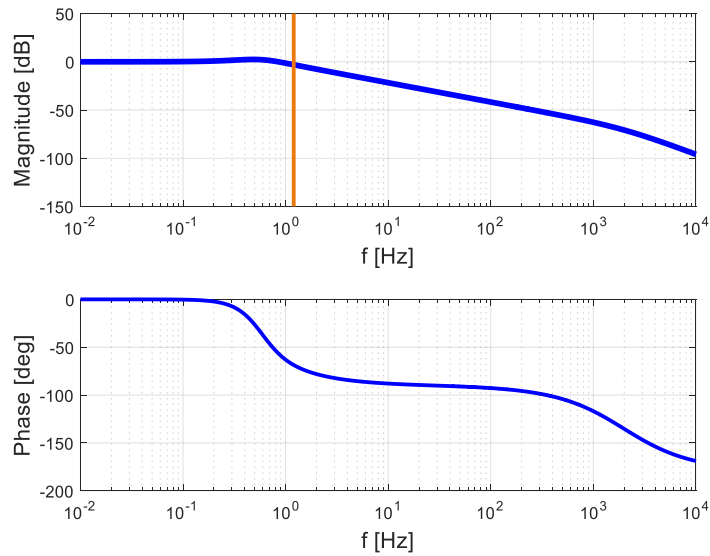
The bandwidth is defined as that frequency, where the closed loop amplification is 3 dB less than at zero frequency. For Bode diagrams, usually the gain at zero frequency is 0 dB. To get this, the output of the transfer function must be of the same physical unity than the input. As closed loop tracking transfer function therefore, the following equation is used for Bode diagrams.

$$\Delta\tilde{\chi} \rightarrow \Delta\hat{\chi}: \quad \mathbf{G}_{\Delta\rho} = \begin{bmatrix} 0 & 0 & \frac{1}{\lambda_{C/A}} & 0 & 0 \end{bmatrix} \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}_{\Delta\tilde{z}} \quad (\text{III-110})$$

The following equation gives the analytic solution of the transfer function, depending on the Kalman gains.

$$\mathbf{G}_{\Delta\rho}(s) = \frac{2 \cdot k_{11}(t) \cdot s^2 + (2 \cdot k_{21}(t) + k_{11}(t)) \cdot s + (2 \cdot k_{31}(t) + k_{21}(t))}{b_8 \cdot s^4 + b_7 \cdot s^3 + b_6 \cdot s^2 + b_5 \cdot k_{11}(t) \cdot s^2 + (b_4 \cdot k_{21}(t) + b_3 \cdot k_{11}(t)) \cdot s + (b_2 \cdot k_{31}(t) + b_1 \cdot k_{21}(t))} \quad (\text{III-111})$$

The Laplace variable s in the transfer function above can be replaced by $j\omega$ if the system is stable.



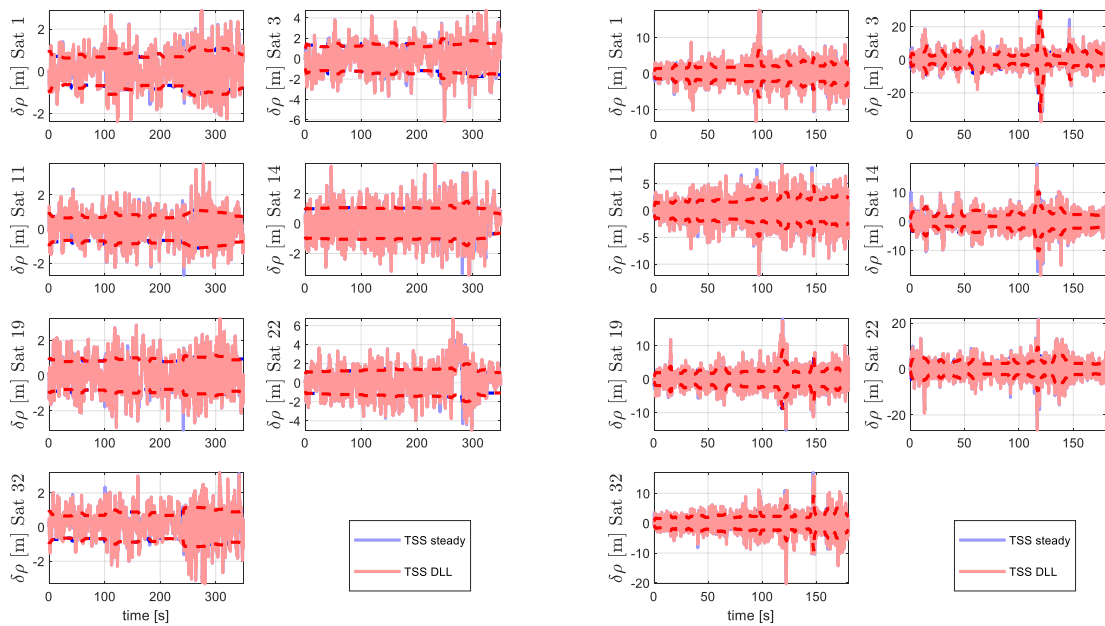
III-43 Closed loop tracking bandwidth

The given Bode diagram example shows the closed loop tracking characteristic at time $t = 60\text{s}$, in case of the DA42 approach.

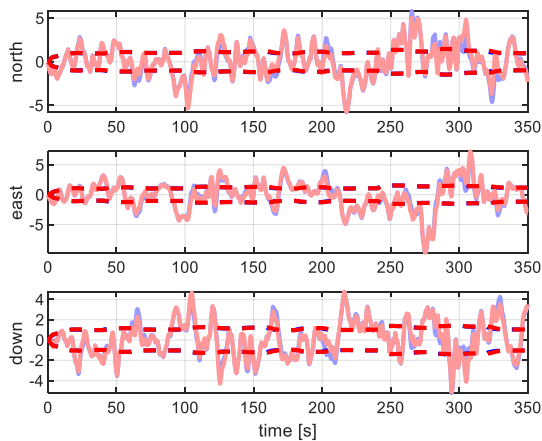
III - 3.6 Steady state adaptive tuning vs. continuous adaptive tuning

In this chapter on unaided scalar total state optimal gain tracking, two different tunings were considered. The natural online adaptive tuning with settling Kalman gains and the tuning which uses the steady state Kalman gains at every time point. (For that, at every point on trajectory or with a defined sample rate, the steady state Kalman gains are calculated and used in the tracking filter). These two approaches will be compared in this section.

UNJAMMED

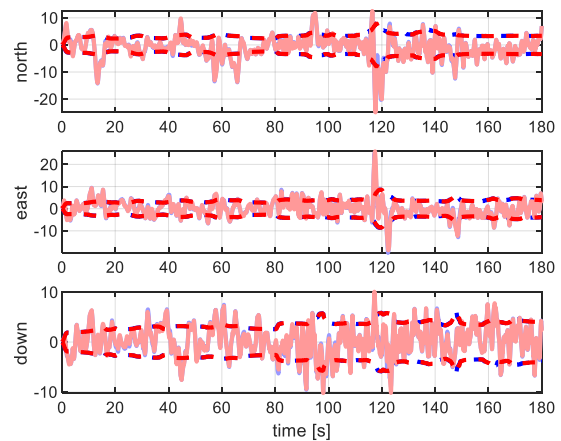


III-44 pseudorange error - DA42

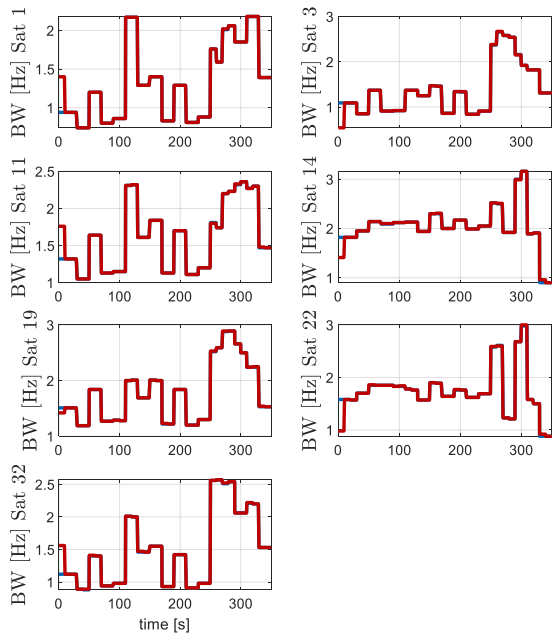


III-46 positioning error - DA42

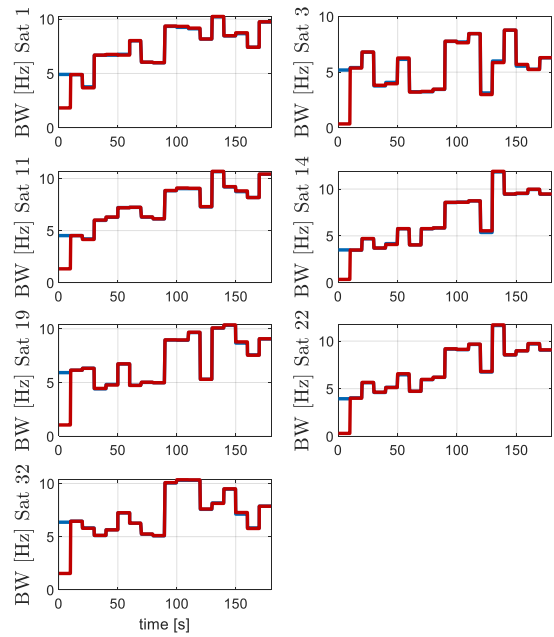
III-45 pseudorange error - high dynamic



III-47 positioning error - high dynamic



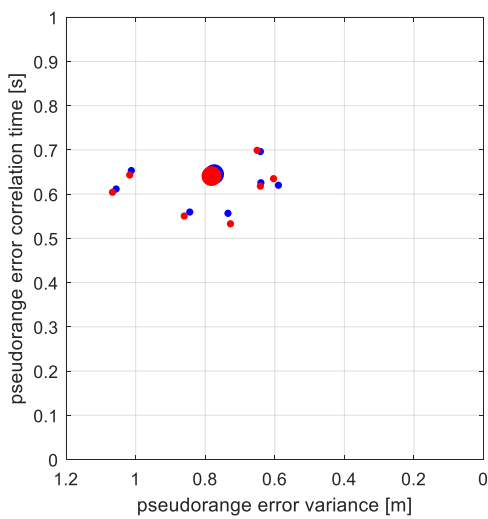
III-48 bandwidth - DA42



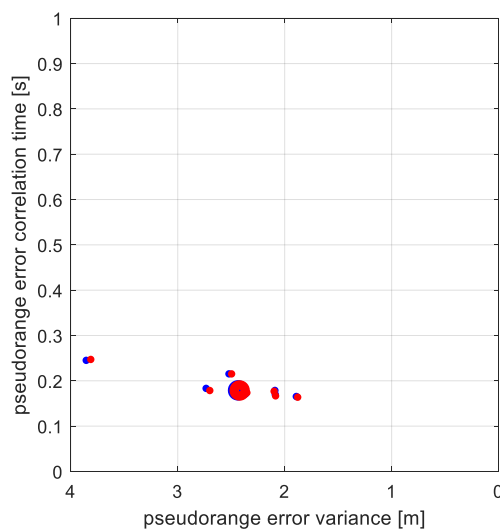
III-49 bandwidth - high dynamic

For a better comparison, the following figures show normalized pseudorange error variances and correlation time constants. The most right upper corner would be the best location, having long correlation time and low error variance.

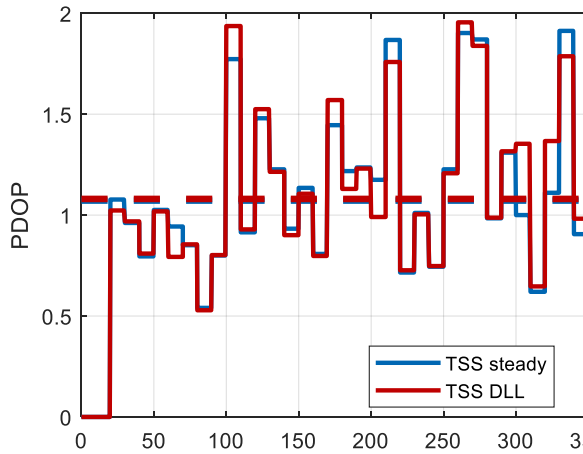
Longer correlation times are desirable from point of loss of lock perspective. Long correlated errors, even having the same error variance, can be handled with much smaller bandwidth, as short correlated errors. In case of long correlated errors, it takes much more time until the accumulated phase difference between the replicated and received code phase, becomes a critical one for causing loss of lock.



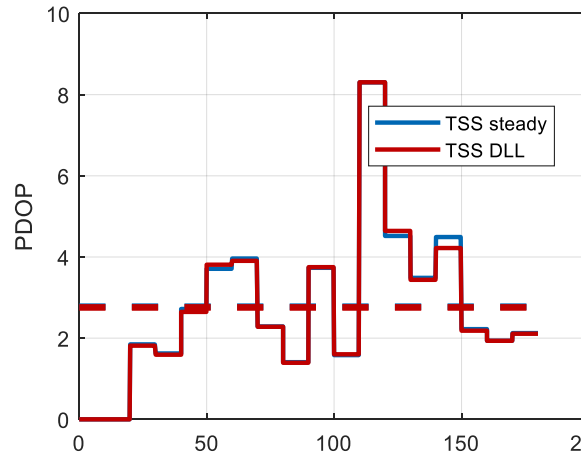
III-50 Comparing raw date - DA42



III-51 Comparing raw date - high dynamic



III-52 PDOP - DA42



III-53 PDOP - high dynamic

CONCLUSION

Comparing:

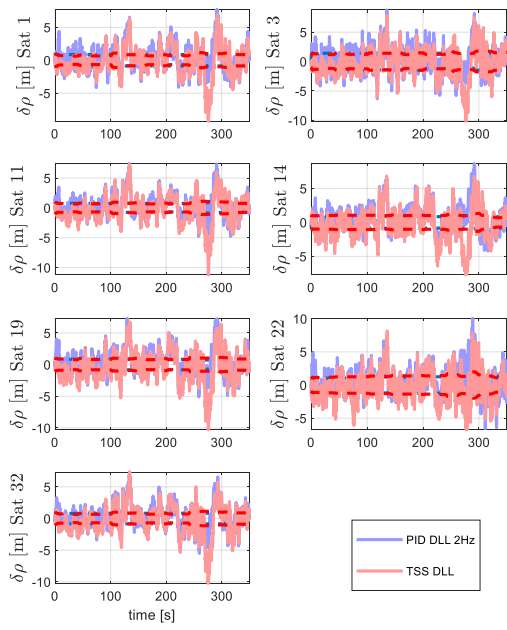
- Unaided total state scalar optimal tracking
- Unaided total state scalar optimal tracking – moving steady state Kalman gains

As the simulations show, a steady state Kalman gain initialization at every point on trajectory, considering the actual tuning figures, provides no benefit.

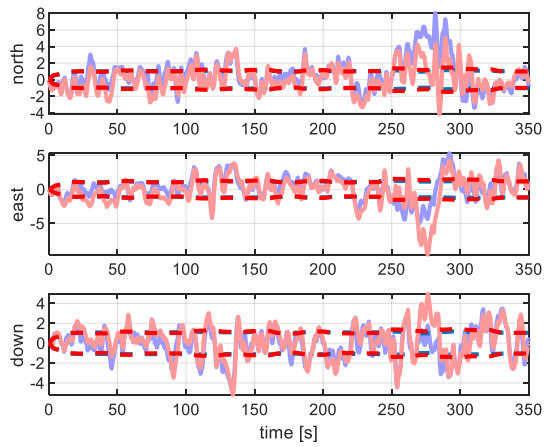
III - 3.7 Classical scalar vs. optimal scalar tracking

UNJAMMED

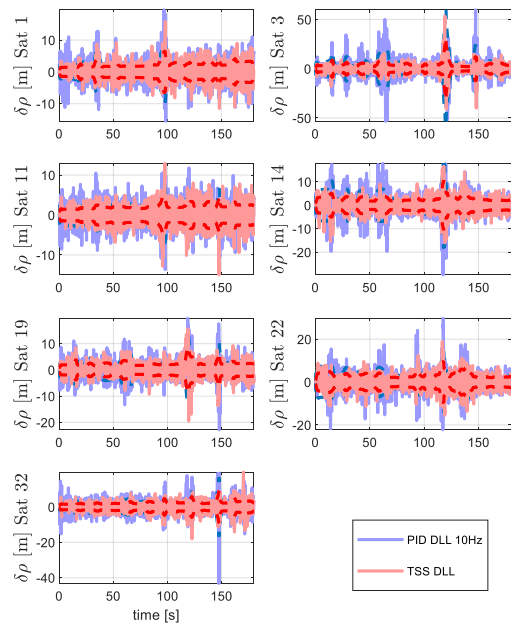
In this section, the scalar fixed gain tracking architecture will be compared with the scalar optimal total state tracking architecture. For tuning the scalar optimal tracking filter, scenario matched adaptive tuning is used. The tuning of the fixed gain tracking loops is based on closed loop tracking bandwidth. In order to make both tracking loops comparable, as tuning relevant closed loop tracking bandwidth for the fixed gain tracking loops, the average closed loop tracking bandwidth of the optimal filter-based tracking loop is used.



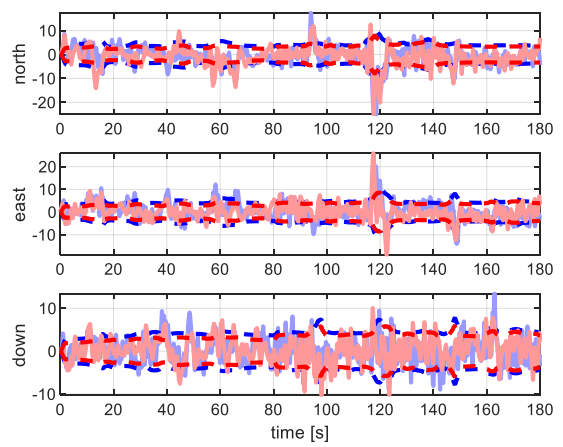
III-54 pseudorange error - DA42



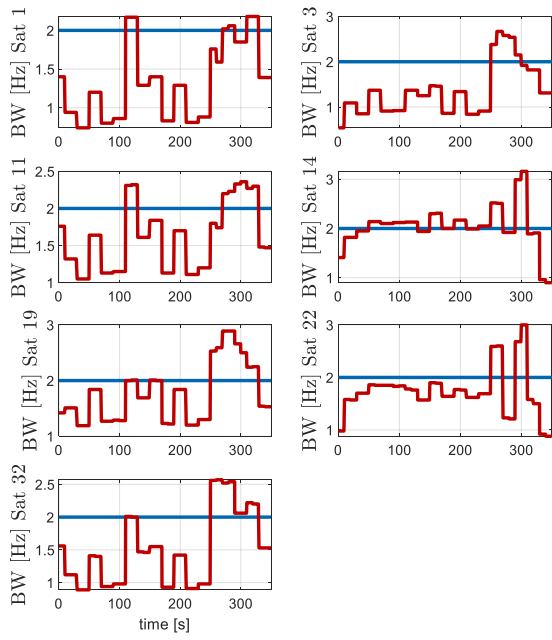
III-56 positioning error - DA42



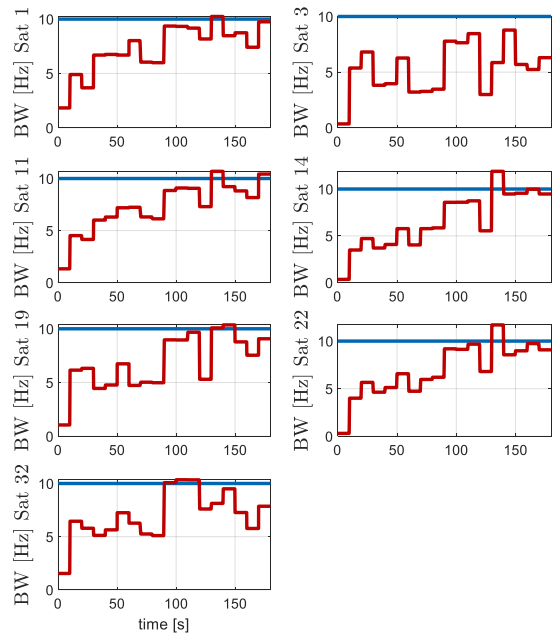
III-55 pseudorange error - high dynamic



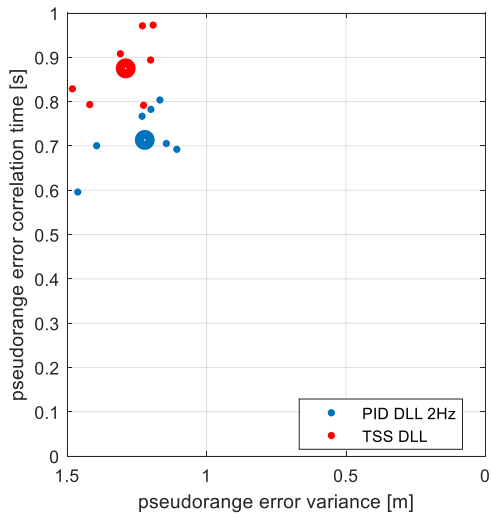
III-57 positioning error - high dynamic



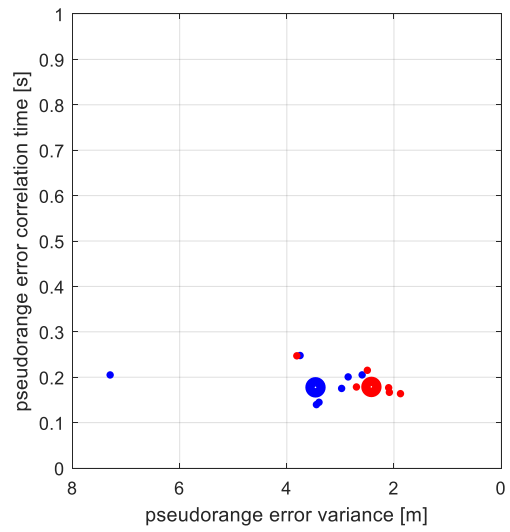
III-58 bandwidth - DA42



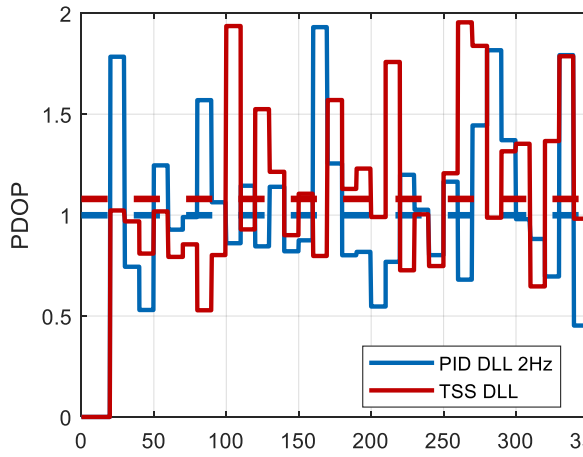
III-59 bandwidth - high dynamic



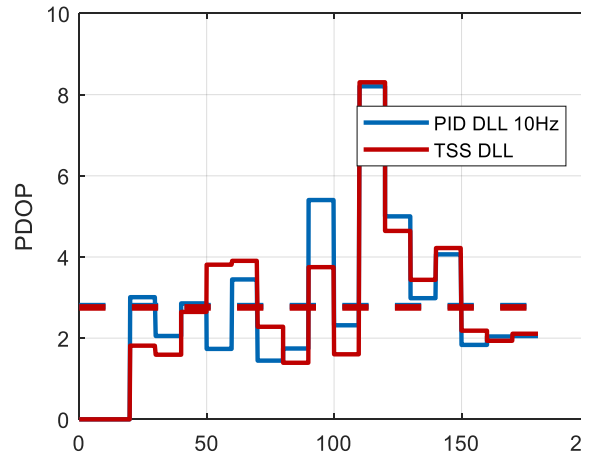
III-60 Comparing raw date - DA42



III-61 Comparing raw date - high dynamic



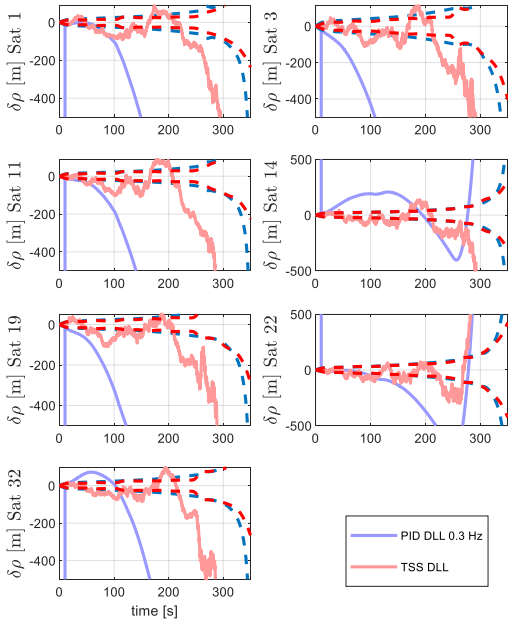
III-62 PDOP - DA42



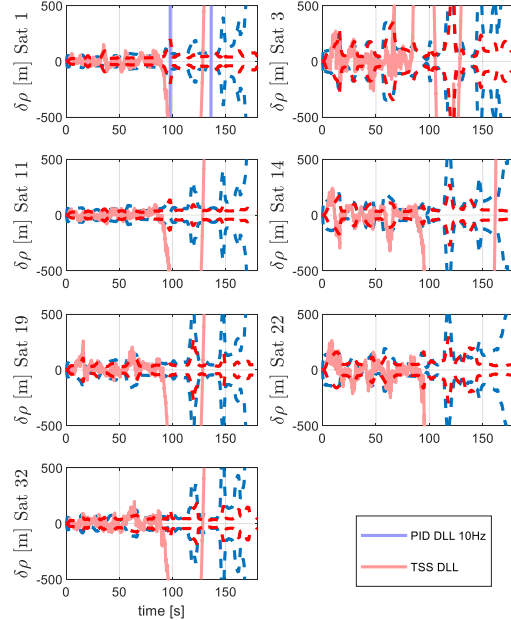
III-63 PDOP - high dynamic

JAMMED

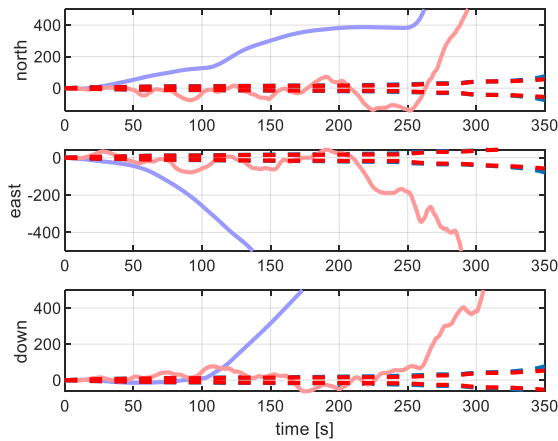
In this section, the scalar fixed gain tracking architecture will be compared with the scalar optimal tracking loop, given a jammed environment. As jammer, a 1kW white noise jammer is used, having a jamming bandwidth of 2 MHz, together with an IF bandwidth of 10 MHz. A detailed jammer link budget analysis is given for the DA42 and the fighter scenario in chapter IV - 2.6



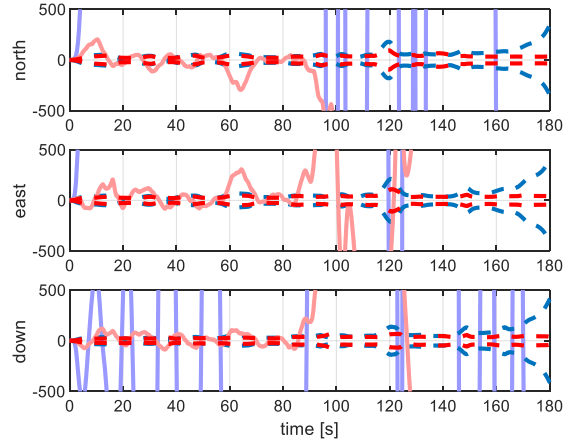
III-64 pseudorange error jammed - DA42



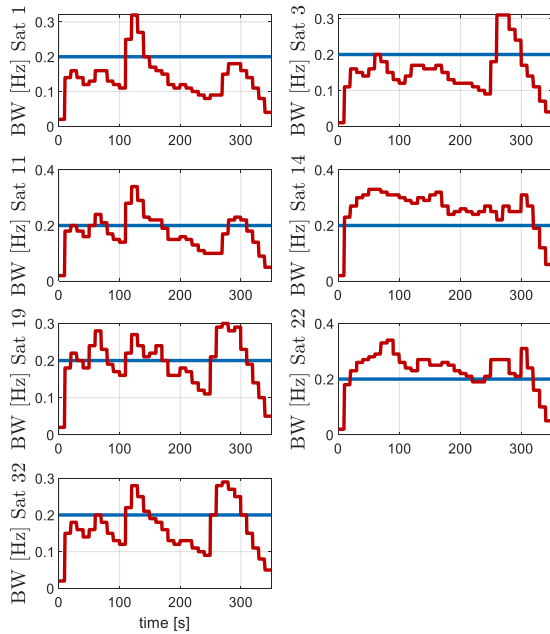
III-65 pseudorange error - high dynamic



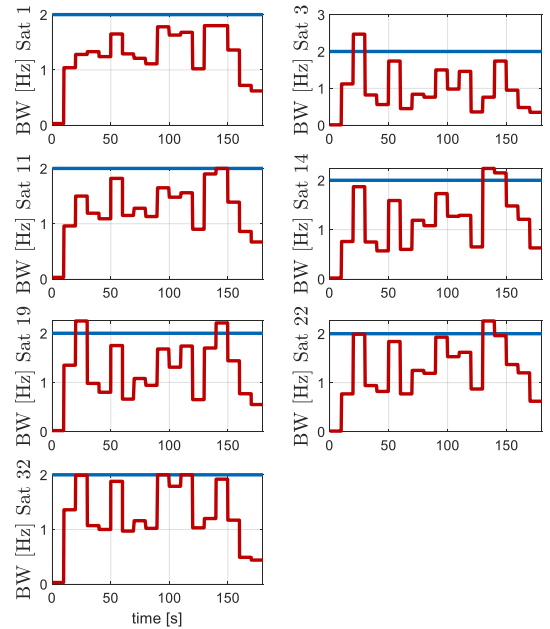
III-66 positioning error jammed - DA42



III-67 positioning error - high dynamic



III-68 bandwidth - DA42



III-69 bandwidth - high dynamic

CONCLUSION

Comparing:

- Total state scalar optimal tracking - unaided
- Scalar fixed gain tracking - unaided

Comparing the scalar fixed gain tracking, using a classical loop filter and the total state optimal tracking loop, using a Kalman filter as loop filter, there is only a small difference regarding tracking performance. The reason is the used tuning bandwidth for the classical loop filter. As tuning bandwidth, the mean bandwidth of the optimal gain tracking loop was selected.

This result is not surprising, because after settling, the Kalman filter becomes steady state and can be interpreted as a fixed gain classical loop filter.

It is even possible to represent a settled Kalman filter as a fixed gain loop structure, using a corresponding set of proportional and integral gains. This relationship is shown in [14] and [22].

The real benefit is the applicable scenario matched tuning in case of total state scalar tracking. If scenario parameters are available, together with some pre knowledge about the platform dynamic, the dynamic model and Kalman tuning factors can be derived.

In case of classical fixed gain tracking loops, one major task is to define the general filter structure and filter order. Given the information regarding the expected mission dynamic and selecting an appropriate dynamic model for the Kalman filter, the optimal filter order is given inherently, preventing remaining control deviations.

Online adaptive tuning, considering real signal to noise measurements, is easy to implement in case of optimal filter-based tracking. Signal to noise measurements can directly be used as input for the measurement error covariance matrix. In case of classical fixed gain tracking, there is no possibility to map directly real signal to noise measurements on loop filter tuning parameters.

In case of jamming, the scalar optimal tracking approach shows a better performance, because of a tuning setting, which matches better the real environment. The jamming simulations show, that the predicted error variance information are only valid, as long as the tracking loop is locked.

The following table summarizes the pros and cons of the total state scalar unaided tracking approach.

Major pros and cons of scalar optimal tracking

Scalar fixed gain – unaided

- The tracking filter order is an inherent result of the assumed platform dynamic by using the corresponding dynamic model in the Kalman filter propagation equation.
- Application of scenario matched tuning possible. All platform dynamic and noise environment characteristics can be mapped bijective on the Kalman filter tuning parameters. The closed loop tracking bandwidth is a direct consequence of the selected dynamic model and the applied tuning parameters.
Especially in case of planned missions, the dynamic behavior is known and a corresponding dynamic model can be selected. All deviations from this dynamic model are also known pre mission and can be used to model the errors within the Kalman filter. This approach gives an optimal tuning
- Inherently available tracking error covariance information by the Kalman filter state error covariance.
The error covariance includes noise caused errors and also dynamic caused errors.
- Easy online adaptive tuning. Changing platform dynamic characteristics and a changing signal to noise environment changes automatically the tuning, by mapping these values on the Kalman filter tuning parameters

III - 4 Scalar optimal filter carrier smoothing

III - 4.1 Motivation

In the previous section, a Kalman filter was introduced and applied as an optimal loop filter in a code tracking loop.

One main application field of Kalman filter is sensor fusion, which makes the application of the Kalman filter perfect for carrier aided code tracking architectures.

The following literature gives examples of carrier aided code tracking architectures, where always classical loop filter structures are discussed: [43], [44] and [45]. In [46], an optimal filter based carrier tracking architecture is considered, but without any coupling to a code tracking loop.

This section provides the following contributions

Contribution 1: An optimal filter-based carrier aided code tracking loop will be developed, where the applied Kalman filter works as a loop filter and additionally, realizes the fusion of code and carrier tracking results. In contrast to fixed coupling between carrier and code tracking loop in classical architectures, here an optimal adaptive coupling between carrier and code tracking loop is realized.

Contribution 2: The already introduced novel scenario matched tuning is in this section extended to carrier tracking and is applied to carrier aided code tracking.

Contribution 3: Because classical fixed gain carrier aided code solutions do not provide inherently tracking error covariance information, an error covariance model will be developed, which can be propagated standalone, in parallel to the tracking itself. Moreover, it will be shown that optimal filter-based carrier aided code tracking do provide inherently tracking error covariance information, considering the coupled solution.

III - 4.2 Tracking architecture – classical fixed gain carrier smoothing

In this section, the principle structure of classical carrier smoothing, using fixed gain loop filters, will be explained. A similar structure for dual frequency tracking can be found in [47] and [48].

Principally, there are two loops – the code tracking and the carrier tracking loop – which are coupled, using a fixed factor $\frac{f_{CA0}}{f_{IF0}}$. This coupling does not consider any actual error covariance information from code and carrier tracking loop.

The coupling factor can be derived by using the following relations.

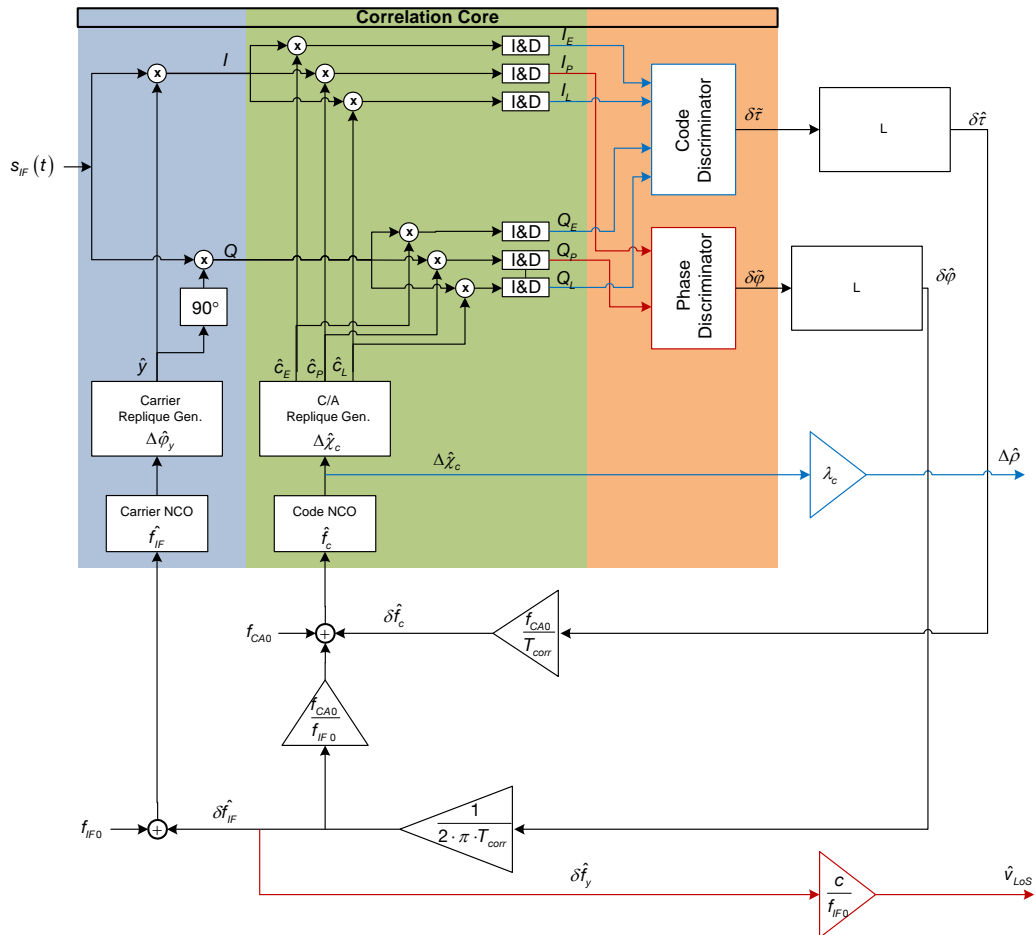
$$\begin{aligned}\Delta\chi &= \Delta f_{C/A} \cdot T_{corr} \\ \Delta\varphi &= \Delta f_{IF} \cdot 2 \cdot \pi \cdot T_{corr}\end{aligned}\tag{III-112}$$

$$\begin{aligned}1 &\leftrightarrow 2 \cdot \pi \\ \Delta\chi &\leftrightarrow \Delta\varphi\end{aligned}\quad \Delta\chi = \frac{1}{2 \cdot \pi} \cdot \Delta\varphi\tag{III-113}$$

Of course, the coupling could be made adaptive, but there is no error information available which can directly be mapped on the coupling factor.

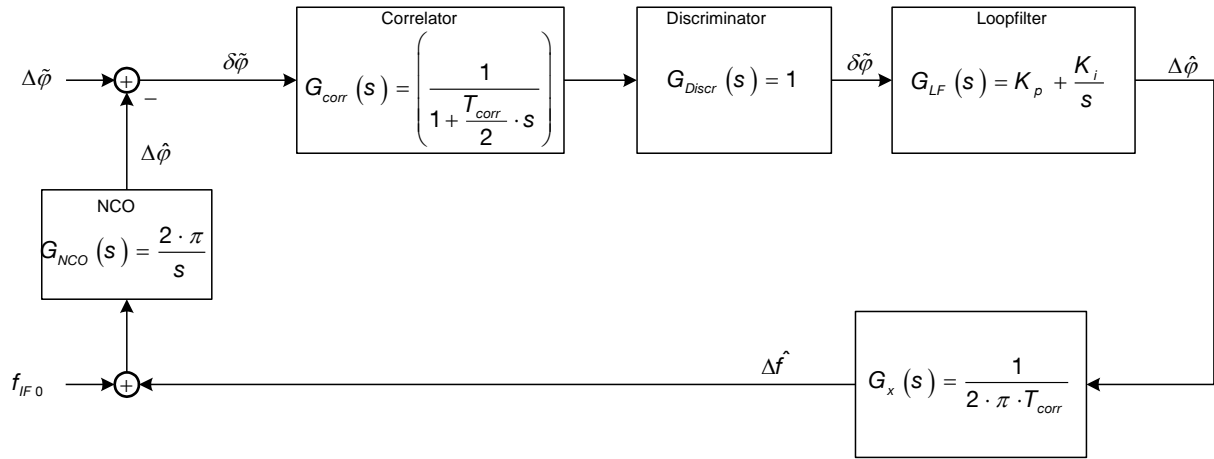
In order to compare optimal filter-based carrier smoothing and fixed gain carrier smoothing, this section introduces at first an equivalent base band model of a carrier tracking loop and the tuning of the carrier tracking loop.

Because there is no inherently provided error covariance information of the carrier tracking loop, similar to equation (III-49), a separate error covariance propagation will be derived.



III-70 Classical scalar PLL aided DLL

For loop tuning, the DLL and PLL are considered separately. For code tracking loop, the already introduced tuning from chapter III - 3.4 is used. A similar approach will be used in the following for the carrier tracking loop.



III-71 Laplace representation - PLL

For the code tracking loop, a closed loop tracking bandwidth of 2 Hz was selected. For carrier tracking, simulations show, that even for the DA42 approach, a 2 Hz bandwidth is too small in order to follow the line of sight dynamic. A closed loop tracking bandwidth of 4 Hz is necessary.

Using the same approach as in(III-18), the absolute value of the closed loop tracking bandwidth is requested to be $\frac{1}{\sqrt{2}}$, at the frequency 4 Hz.

$$|G_{PLL}(j\omega)| = \frac{|K_p \cdot j\omega + K_i|}{|-2 \cdot \pi \cdot T_{corr} \cdot \omega^2 + K_p \cdot j\omega + K_i|} = \frac{\sqrt{K_p^2 \cdot \omega^2 + K_i^2}}{\sqrt{K_p^2 \cdot \omega^2 + (K_i - 2 \cdot \pi \cdot T_{corr} \cdot \omega^2)^2}} = \frac{1}{\sqrt{2}} \quad (\text{III-114})$$

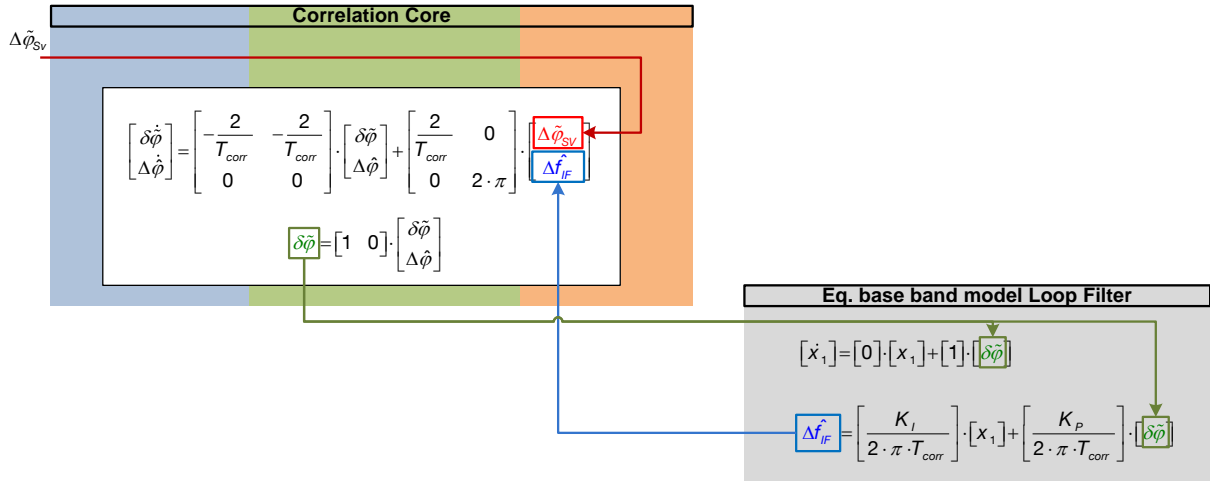
The distributed state space simulation of the carrier tracking loop consists also out of the correlation core and the loop filter.

Because in the previous chapter only code tracking was considered, in the following equations, the corresponding equivalent base band equation for a distributed state space representation of the carrier tracking loop is given.

The correlation core model for a carrier tracking loop gets, according to chapter IV - 2.9 .

$$\begin{bmatrix} \delta\dot{\varphi} \\ \Delta\dot{\hat{\varphi}} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2}{T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\varphi} \\ \Delta\hat{\varphi} \end{bmatrix} + \begin{bmatrix} \frac{2}{T_{corr}} & 0 \\ 0 & 2 \cdot \pi \end{bmatrix} \cdot \begin{bmatrix} \Delta\tilde{\varphi}_{SV} \\ \Delta\hat{f}_{IF} \end{bmatrix} \quad (\text{III-115})$$

The distributed state space model of the loop filter is equivalent to the code tracking loop. The coupling between the loop filter and the correlation core is different, as can be seen in the following figure.



III-72 Distributed state space - PLL

For tracking error covariance information, similar to equation (III-48), the error covariance information for carrier tracking can be calculated, by using the centralized state space representation of the given carrier tracking loop.

$$\delta \dot{\mathbf{z}} = \mathbf{A} \cdot \delta \mathbf{z} + \mathbf{G} \cdot n_{\delta\phi}$$

$$\begin{bmatrix} \delta \dot{\tilde{\phi}} \\ \Delta \dot{\hat{\phi}} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2}{T_{corr}} & 0 \\ \frac{K_p}{T_{corr}} & 0 & \frac{K_i}{T_{corr}} \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\phi} \\ \Delta \hat{\phi} \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_p}{T_{corr}} \\ 1 \end{bmatrix} \cdot n_{\delta\phi} \quad (III-116)$$

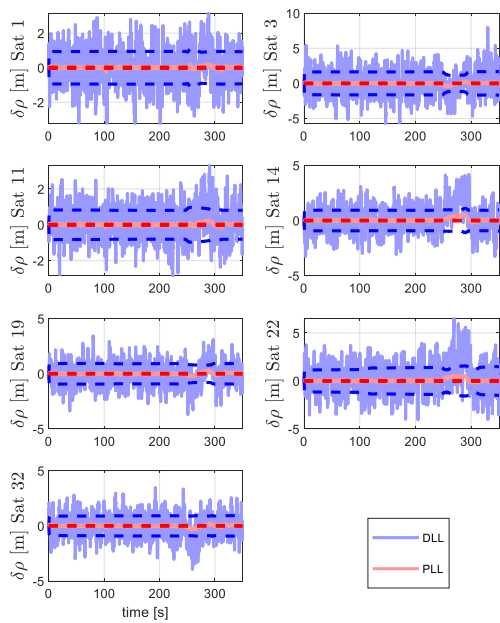
The corresponding state uncertainty matrix \mathbf{Q} gets

$$\mathbf{Q}_k = \mathbf{G} \cdot \mathcal{E} \left\{ \mathbf{n}_{\delta\phi} \cdot \mathbf{n}_{\delta\phi}^T \right\} \cdot \mathbf{G}^T \cdot T_{corr}^2 = \begin{bmatrix} 0 \\ \frac{K_p}{T_{corr}} \\ 1 \end{bmatrix} \cdot \sigma_{n_{\delta\phi}}^2 \cdot \begin{bmatrix} 0 & \frac{K_p}{T_{corr}} & 1 \end{bmatrix} \cdot T_{corr}^2 \quad (III-117)$$

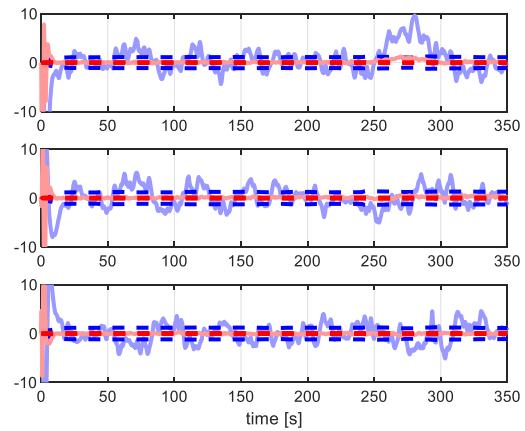
The state error covariance can be propagated according to the next equation.

$$\mathbf{P}_{k+1} \approx (\mathbf{1} + \mathbf{A} \cdot T_{corr}) \cdot \mathbf{P}_k \cdot (\mathbf{1} + \mathbf{A} \cdot T_{corr})^T + \mathbf{Q}_k \quad (III-118)$$

The following figures compare the carrier solution with the pure code solution.



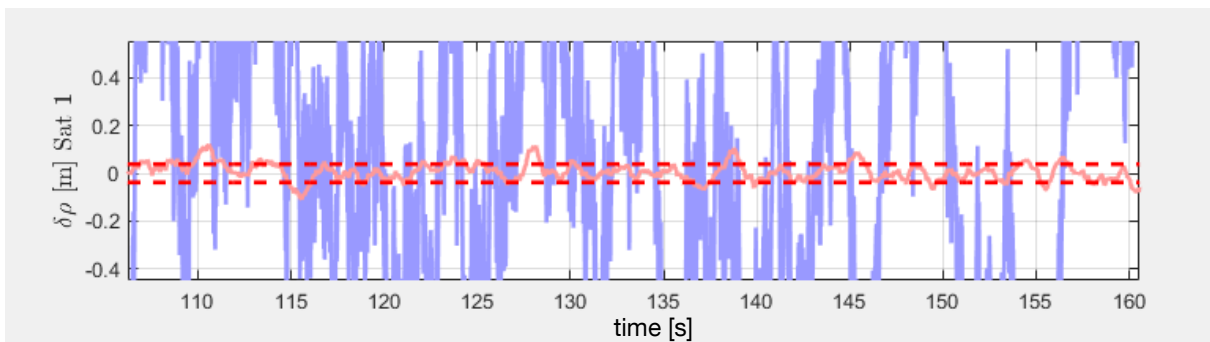
III-73 Raw data error - DLL versus PLL



III-74 Positioning error - DLL versus PLL

As the figures above show, the error variance of the carrier solution is much lower than the error variance of the code solution, despite the closed loop tracking bandwidth of the carrier tracking loop is 4 Hz compared to the closed loop tracking bandwidth of the code solution with 2 Hz.

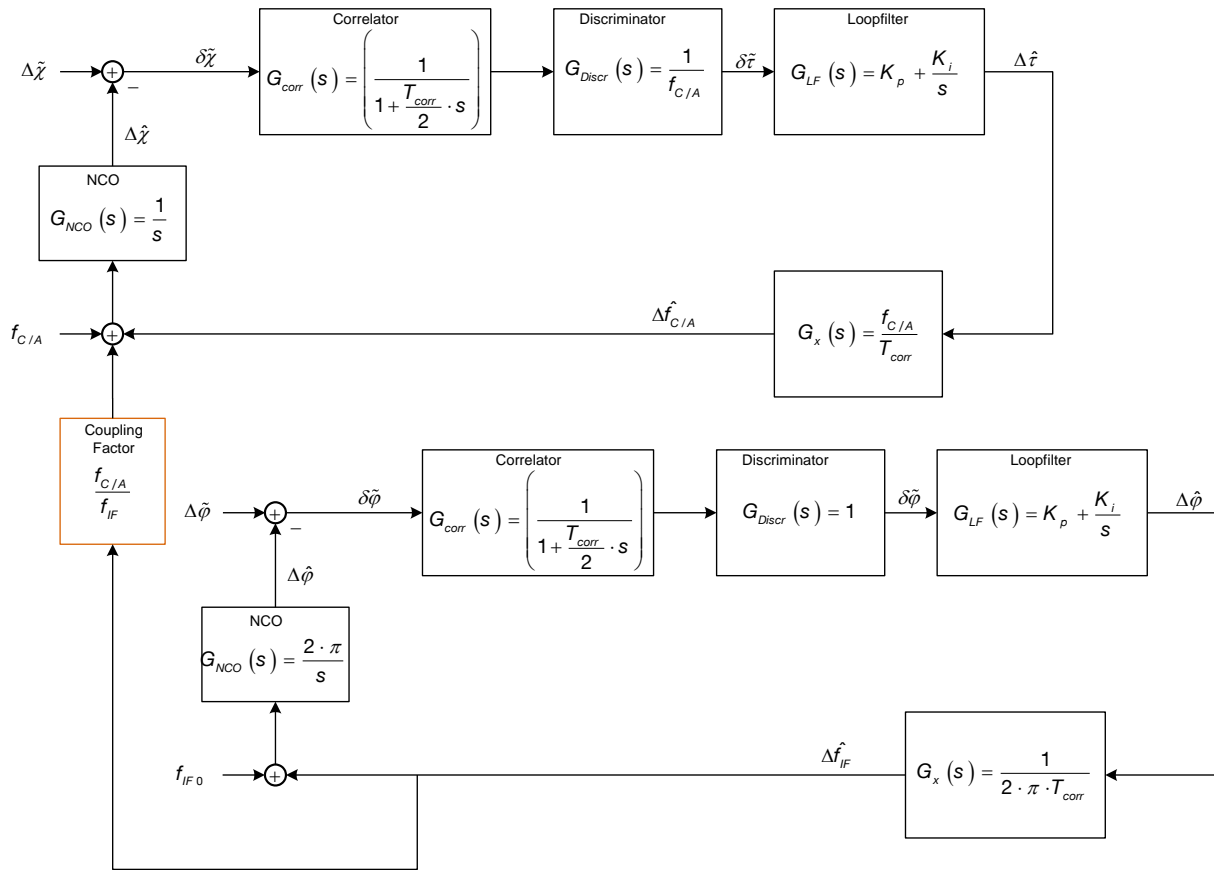
The following figure shows the zoomed version of the pseudorange error from one satellite.



III-75 - Raw data error - DLL versus PLL - zoomed

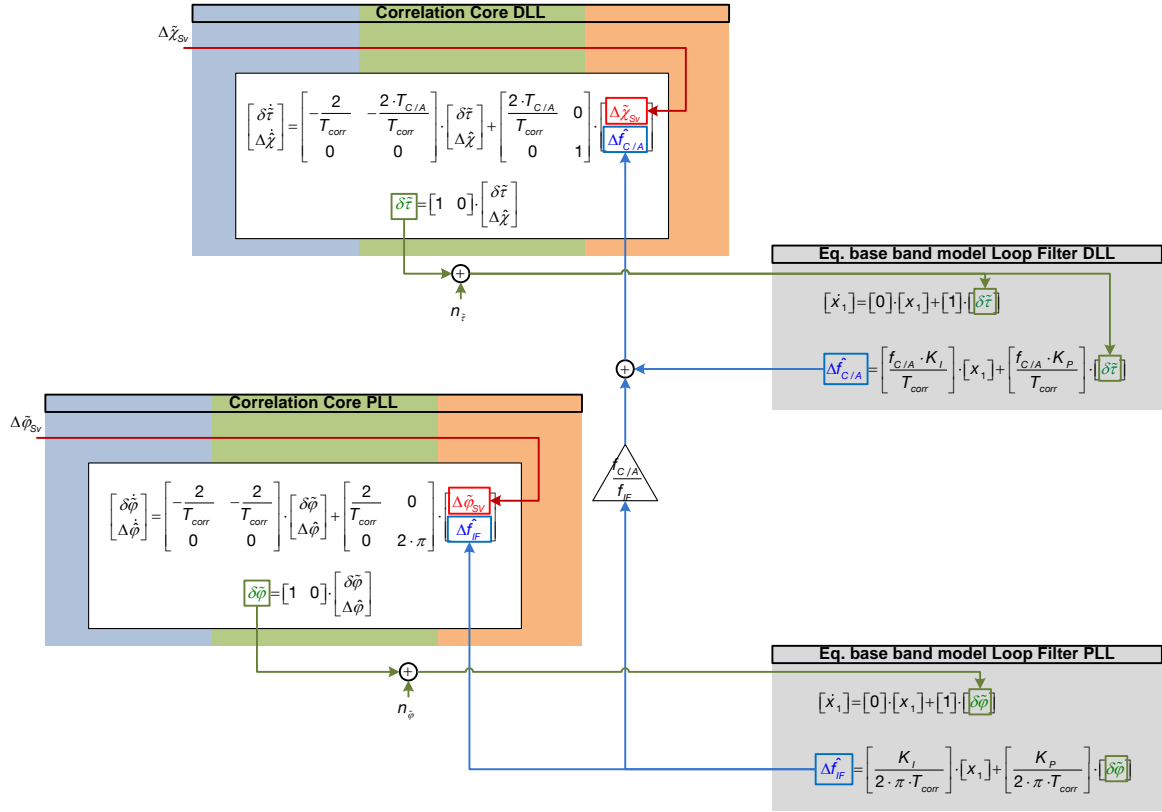
Especially the zoomed figure shows that the separate pseudorange error covariance prediction provides correct results. Nevertheless, the error variance needs to be calculated in parallel. Only noise caused errors are considered, dynamic stress is not included in the error covariance propagation.

Figure III-70 shows the tracking architecture, realizing classical fixed gain carrier aided code tracking. For simulation and also for ongoing analysis, an equivalent base band Laplace representation and distributed state space representation for this carrier aided code tracking will be introduced.



III-76 Fixed gain carrier aided code tracking - Laplace representation

The corresponding distributed state space realization of a carrier aided code tracking loop in equivalent base band, is shown in the following architecture.



III-77 Distributed state space realization - carrier aided code tracking (fixed gain)

Classical fixed gain carrier aided code tracking lacks the same disadvantage as classical fixed gain code tracking. There is no inherently available error covariance, which is necessary for subsequent positioning solution.

In the section on fixed gain scalar code tracking, already the centralized state space form was used for a separate error covariance calculation. This approach is also applied in this case. Therefore, in the following, the centralized state space representation for carrier aided code tracking is derived.

$$\begin{bmatrix} \delta \dot{\tau} \\ \Delta \dot{\hat{\chi}} \\ \dot{x}_{DLL} \\ \delta \dot{\varphi} \\ \Delta \dot{\hat{\phi}} \\ \dot{x}_{PLL} \end{bmatrix} = \begin{bmatrix} \frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 & 0 & 0 & 0 \\ a_{DLL} \cdot \frac{f_{C/A} \cdot K_{p,DLL}}{T_{corr}} & 0 & \frac{a_{DLL} \cdot f_{C/A} \cdot K_{i,DLL}}{T_{corr}} & \frac{a_{PLL} \cdot 2 \cdot \pi \cdot f_{C/A} \cdot K_{p,PLL}}{f_{IF} \cdot T_{corr}} & 0 & \frac{a_{PLL} \cdot 2 \cdot \pi \cdot K_{i,PLL}}{f_{IF} \cdot T_{corr}} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{2}{T_{corr}} & -\frac{2}{T_{corr}} & 0 \\ 0 & 0 & 0 & \frac{K_{p,PLL}}{T_{corr}} & 0 & \frac{K_{i,PLL}}{T_{corr}} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi} \\ x_{DLL} \\ \delta \tilde{\varphi} \\ \Delta \hat{\phi} \\ x_{PLL} \end{bmatrix} + \begin{bmatrix} \frac{2}{T_{corr} \cdot f_{C/A}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{2}{T_{corr}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \tilde{\chi} \\ \Delta \tilde{\varphi} \end{bmatrix} \quad (III-119)$$

For a separate error covariance propagation, the inputs $\Delta\tilde{\chi}$ and $\Delta\tilde{\varphi}$ are assumed to be zero. The whole system is driven by the discriminator measurement noise $n_{\tilde{\tau}}$ and $n_{\tilde{\varphi}}$. The only inputs to the system are the phase discriminator noise and the code discriminator noise.

$$\delta\dot{\mathbf{z}} = \mathbf{A} \cdot \delta\mathbf{z} + \mathbf{G} \cdot \mathbf{n}$$

$$\begin{bmatrix} \delta\dot{\tilde{\tau}} \\ \Delta\dot{\tilde{\chi}} \\ \dot{x}_{DLL} \\ \delta\dot{\tilde{\varphi}} \\ \Delta\dot{\tilde{\varphi}} \\ \dot{x}_{PLL} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 & 0 & 0 & 0 \\ \frac{f_{C/A} \cdot K_{p,DLL}}{T_{corr}} & 0 & \frac{f_{C/A} \cdot K_{i,DLL}}{T_{corr}} & \frac{f_{C/A} \cdot K_{p,PLL}}{f_{IF} \cdot T_{corr}} & 0 & \frac{f_{C/A} \cdot K_{i,PLL}}{f_{IF} \cdot T_{corr}} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{2}{T_{corr}} & -\frac{2}{T_{corr}} & 0 \\ 0 & 0 & 0 & \frac{K_{p,PLL}}{T_{corr}} & 0 & \frac{K_{i,PLL}}{T_{corr}} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\tilde{\chi} \\ x_{DLL} \\ \delta\tilde{\varphi} \\ \Delta\tilde{\varphi} \\ x_{PLL} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{f_{C/A} \cdot K_{p,DLL}}{T_{corr}} & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \frac{K_{p,PLL}}{T_{corr}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} n_{\tilde{\tau}} \\ n_{\tilde{\varphi}} \end{bmatrix} \quad (III-120)$$

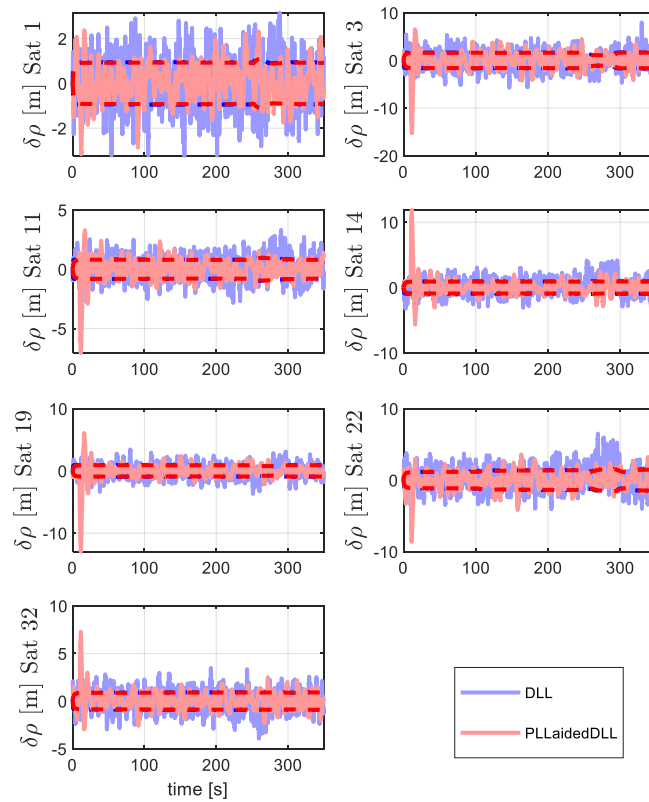
The state error uncertainty matrix \mathbf{Q}_k is calculated according to next equation.

$$\mathbf{Q}_k = \mathbf{G} \cdot \mathcal{E}\{\mathbf{n} \cdot \mathbf{n}^T\} \cdot \mathbf{G}^T \cdot T_{corr}^2 = \begin{bmatrix} 0 & 0 \\ \frac{f_{C/A} \cdot K_{p,DLL}}{T_{corr}} & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \frac{K_{p,PLL}}{T_{corr}} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_{n,\delta\tau}^2 & 0 \\ 0 & \sigma_{n,\delta\varphi}^2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 \\ \frac{f_{C/A} \cdot K_{p,DLL}}{T_{corr}} & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \frac{K_{p,PLL}}{T_{corr}} \\ 0 & 1 \end{bmatrix}^T \quad (III-121)$$

Using equation (III-54), the state error covariance propagation can be derived.

Regarding tuning, in case of carrier aiding, the closed loop tracking bandwidth of the code tracking loop must be reduced. In this example, the bandwidth is reduced to 1 Hz, in contrast to 2 Hz for the unaided code tracking loop.

The following figure compares the pure code tracking with the carrier aided code tracking.



III-78 PLL aided DLL - pseudoranges - DA42

The red curves represent the carrier aided code tracking. The error variance of the carrier aided code tracking solution is smaller, compared to the pure code tracking solution. Despite the tracking bandwidth of the code tracking loop was reduced from 2 Hz to 1 Hz, the dynamic stress errors are even smaller than in the pure code tracking solution, which still uses a closed loop tracking bandwidth of 2 Hz.

The closed loop tracking bandwidth of the code tracking loop could be reduced to a much lower bandwidth than the applied 1 Hz. This would give even better results for the carrier aided code architecture. Selecting the optimal code tracking bandwidth cannot be derived analytically. It is much more an experience-based approach.

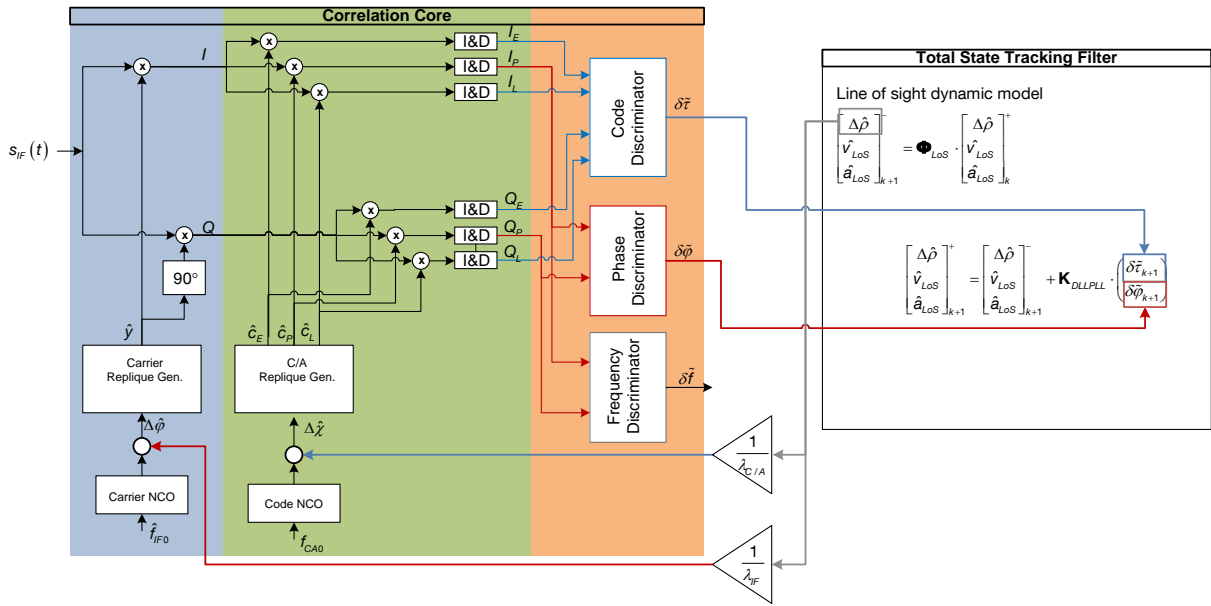
In the next section, an optimal filter-based approach is used for carrier smoothing. There, the closed loop tracking bandwidth is inherently a function of the actual carrier tracking error variance, which leads automatically to an optimal selection of the closed loop code tracking bandwidth.

III - 4.3 Tracking architecture – optimal filter carrier smoothing

In this section, carrier smoothing by using a combined optimal filter, will be developed. In the previous section on fixed gain carrier aided code tracking, the carrier tracking loop and code tracking loop are basically independent. The carrier loop is coupled with the code tracking loop by using a fixed coupling. The coupling does not consider mutual error covariance information from code or carrier tracking loop.

The optimal filter carrier aided code tracking loop adapts the coupling based on the inherently available error covariance information from code and carrier tracking loop. Moreover, similar to the already introduced pure optimal filter-based code tracking loop, scenario matched tuning can be applied.

The following figure shows the corresponding tracking architecture.



III-79 Scalar total state optimal PLL aided DLL

The line of sight dynamic model for the carrier tracking loop is equivalent to the line of sight dynamic of the code dynamic. Also, the filter states are equal, because they represent the geometric line of sight dynamic.

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{n}$$

$$\begin{bmatrix} \Delta \dot{\rho} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_{a,LoS}} \end{bmatrix} \cdot \begin{bmatrix} \Delta \rho \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \end{bmatrix}, \quad \Phi = \mathbf{I}^{3 \times 3} + \mathbf{A} \cdot \Delta T \quad (III-122)$$

The only difference is the observation space transformation, given in the next equation.

$$\Delta \chi = \frac{\Delta \rho}{\lambda_{C/A}} = \frac{\Delta \rho \cdot f_{C/A}}{c} \rightarrow \Delta \tau = \frac{\Delta \chi}{f_{C/A}} \quad (III-123)$$

$$\Delta\varphi = \frac{2 \cdot \pi \cdot \Delta\rho}{\lambda_{IF}} = \frac{2 \cdot \pi \cdot f_{IF0} \cdot \Delta\rho}{c} \quad (\text{III-124})$$

In matrix form this transformation gets

$$\begin{bmatrix} \Delta\hat{\tau} \\ \Delta\hat{\varphi} \end{bmatrix} = \mathbf{H}^T \cdot \begin{bmatrix} \Delta\hat{\rho} \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{\tau}^T \\ \mathbf{h}_{\varphi}^T \end{bmatrix} \cdot \begin{bmatrix} \Delta\hat{\rho} \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{1}{c} & 0 & 0 \\ \frac{2 \cdot \pi \cdot f_{IF0}}{c} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta\hat{\rho} \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix} \quad (\text{III-125})$$

The update equation of the Kalman filter gets

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K}_{k+1}^{3 \times 2} \cdot \begin{bmatrix} \delta\tilde{\tau}_{k+1} \\ \delta\tilde{\varphi}_{k+1} \end{bmatrix} \quad (\text{III-126})$$

For tuning, again the scenario matched tuning is applied, which was already introduced in the previous chapter on scalar optimal code tracking. Additionally, to the code phase error measurement variance, the carrier phase error measurement variance is necessary for setting up the measurement error covariance matrix.

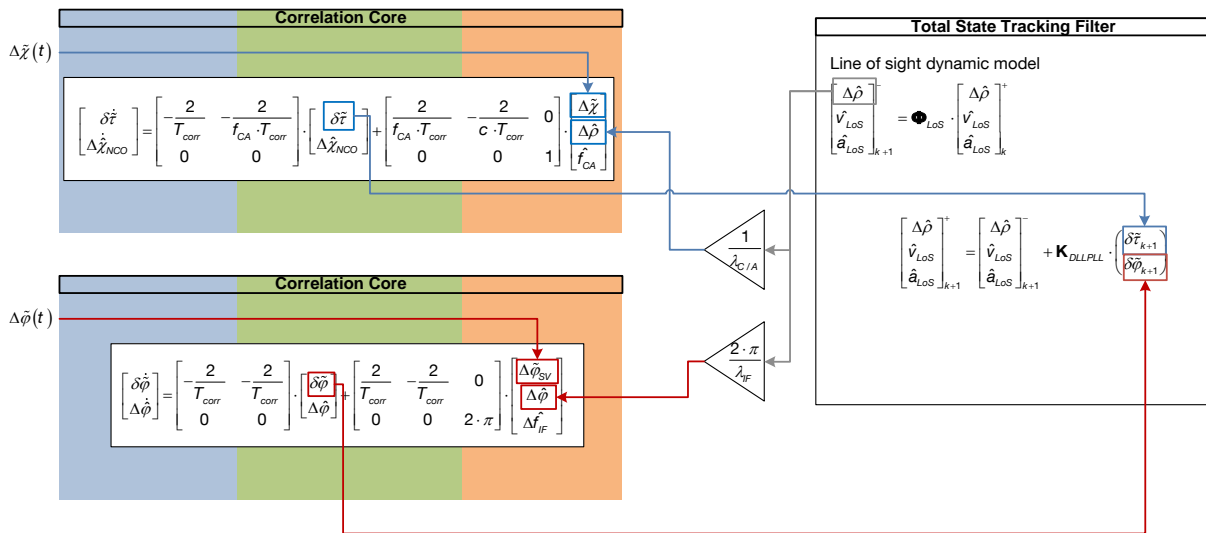
$$\mathbf{R} = \begin{bmatrix} \sigma_{\delta\tau}^2 & 0 \\ 0 & \sigma_{\delta\varphi}^2 \end{bmatrix} \quad (\text{III-127})$$

The model uncertainty matrix \mathbf{Q} is equivalent to the already introduced matrix in the context of pure code tracking.

The combination of carrier measurements and code measurements is based on the measurement error variance information and also the actual state error covariance.

III - 4.4 Equivalent base band dynamic state space simulation

For equivalent base band simulation, a distributed state space representation is introduced, which is shown in the following figure.

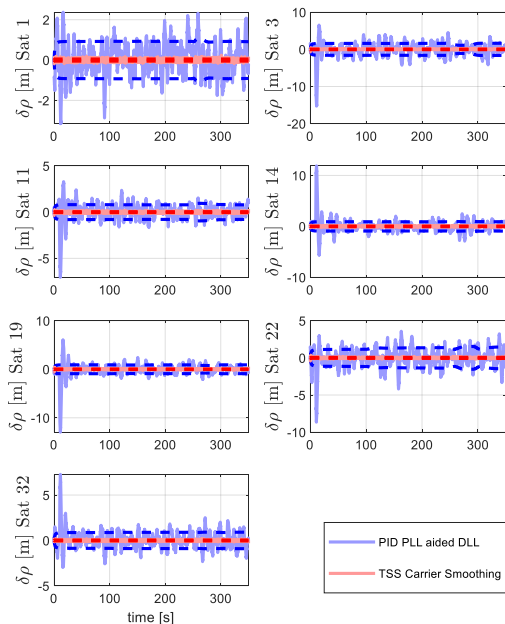


III-80 Distributed state space representation - Optimal carrier aided code tracking loop

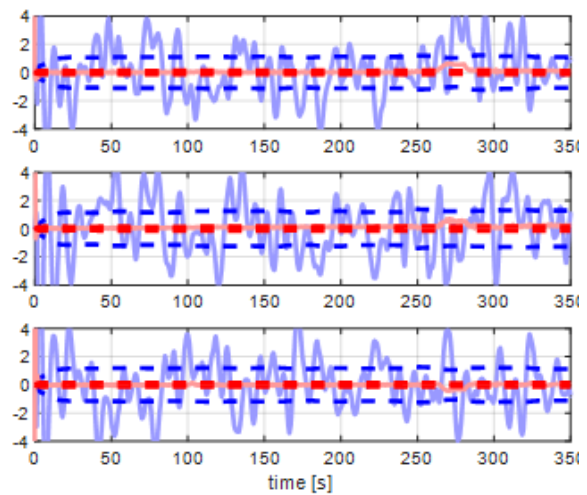
III - 4.5 Classical fixed gain vs. optimal filter carrier smoothing

This section is on comparing optimal filter-based carrier smoothing and classical fixed gain carrier aided code tracking.

As example, the DA4 approach is used. For tuning the Kalman filter, the already introduced scenario matched tuning from the previous section is applied.



III-81 pseudorange error - DA42

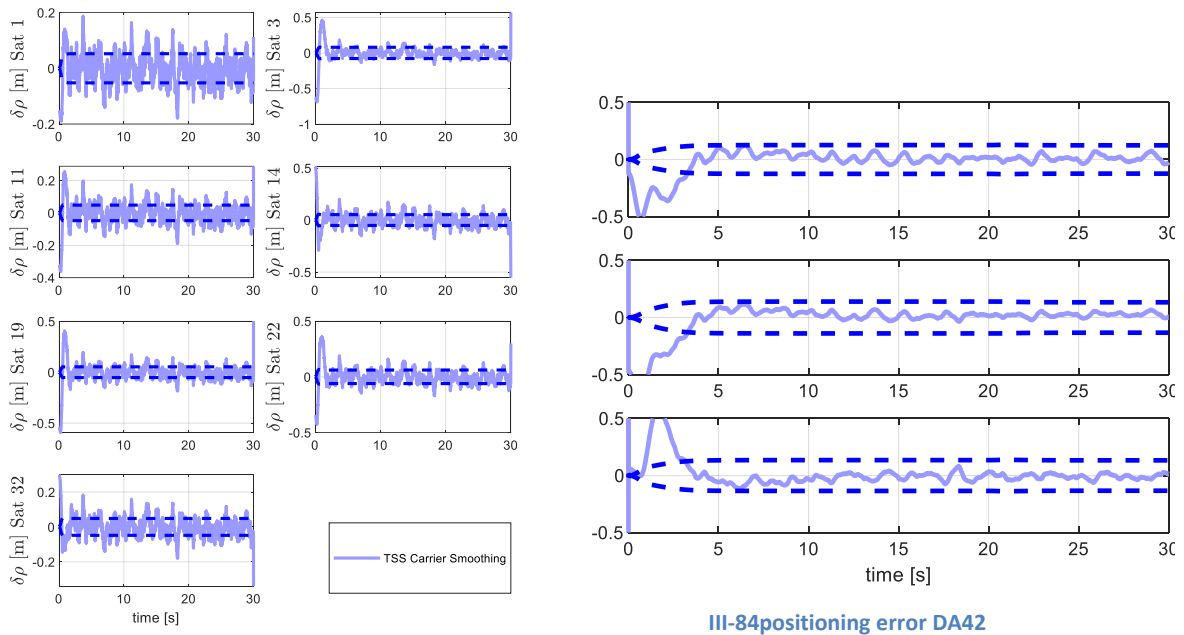


III-82 positioning error DA42

The figures show that optimal filter-based carrier smoothing provides far more better tracking

results. The error variance is much smaller than in case of classical PLL aided DLL tracking. A similar result would be also possible with PLL aided DLL tracking, but finding the correct tuning respectively tracking bandwidth for PLL and DLL is a tricky task. The optimal filter-based carrier smoothing provides automatically the correct tracking bandwidth, by using scenario matched tuning and considering the actual error variance in the carrier path for appropriate aiding of the code path. And especially this inherent optimal tuning, given the actual mission and assumed or measured signal to noise ratio, is the advantage of the optimal filter carrier smoothing.

In order to get an impression of the quality of optimal filter-based carrier smoothing, the following figures show only this approach.



III-83 pseudorange error - DA42

CONCLUSION

Fixed carrier aided code tracking	Optimal filter carrier smoothing
<p>➖ Coupling of carrier and code tracking loop using a fixed factor. The coupling does not consider actual error variance information in code and carrier tracking loop.</p>	<p>➕ The Kalman filter realizes the coupling between code and carrier tracking loop as a kind of signal fusion. The measurements at the output of the code discriminator and carrier discriminator are combined, based on the corresponding measurement error covariance information.</p>

-	<p>Carrier and code tracking loop are basically tuned separately with focus on closed loop tracking bandwidth.</p> <p>But to get a significant improvement in case of carrier aiding, the tuning of the code tracking loop has to adapted manually by reducing the closed loop tracking bandwidth.</p>	+	<p>For tuning, the scenario matched tuning regarding line of sight dynamic model and measurement error covariance is applied. Based on that tuning information, the Kalman filter inherently realizes the optimal tuning and also closed loop tracking bandwidth</p>
-	<p>No error covariance information available. For error covariance propagation, a separate error model has to be propagated in parallel. This error covariance information only includes noise caused errors and no errors due to dynamic.</p>	+	<p>Inherently available tracking error covariance information by the Kalman filter state error covariance.</p> <p>The error covariance includes noise caused errors and also dynamic caused errors.</p>
		+	<p>Easy online adaptive tuning through scenario matched tuning. Changing platform dynamic characteristics and a changing signal to noise ratio, automatically leads to adaption of the tuning settings.</p>

III - 5 Scalar error state optimal filter tracking (unaided)

III - 5.1 Motivation

In section III - 3 an unaided scalar optimal tracking architecture in total state representation was developed. The whole platform dynamic was modeled within the dynamic model of the Kalman filter. This section pursues the question, if there is some benefit, using a scalar error state approach, even if there is no external aiding.

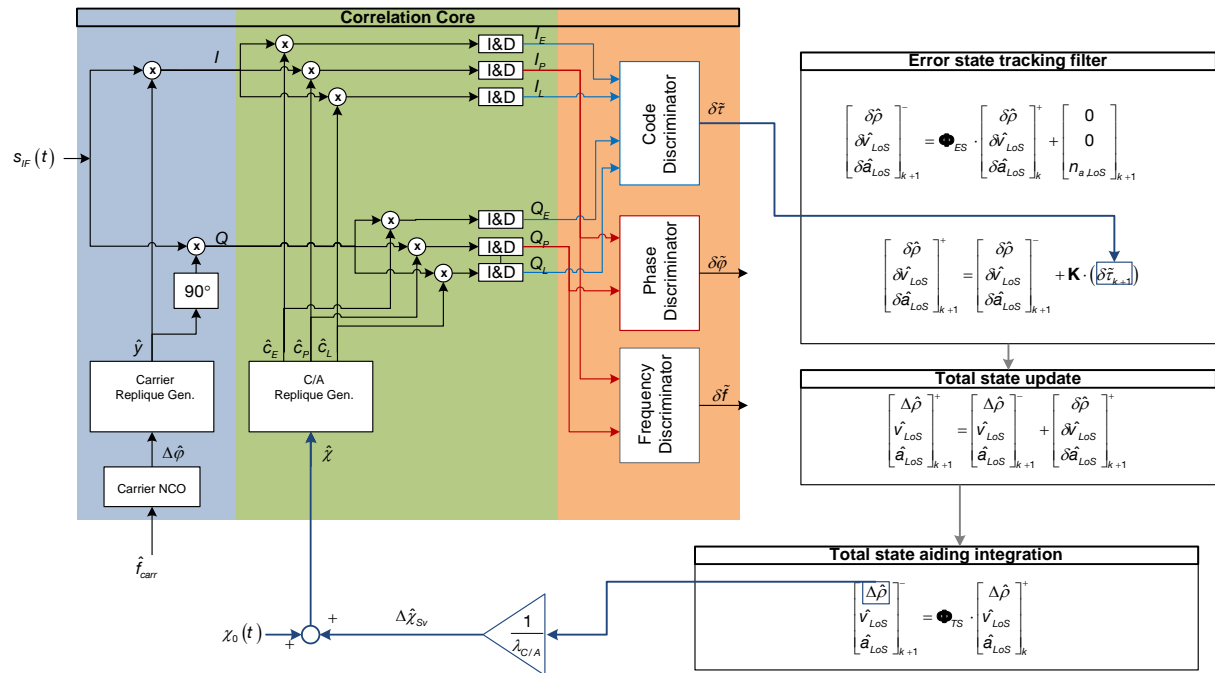
The contributions of this section are:

Contribution 1: Development of an error state optimal filter-based tracking architecture

Contribution 2: Evaluation if an error state realization in case of unaided scalar tracking provides some advantages.

III - 5.2 Tracking Architecture

For the error state approach, within the dynamic model of the Kalman filter, only the error shall be modeled. Therefore, the platform dynamic propagation must be realized outside the filter.



III-85 Scalar error state - unaided

As total state platform dynamic model, the following differential equation system is used.

$$\dot{\mathbf{z}} = \mathbf{A}_{TS} \cdot \mathbf{z}$$

$$\begin{bmatrix} \Delta \dot{\hat{\rho}} \\ \dot{\hat{\mathbf{v}}}_{LoS} \\ \dot{\hat{\mathbf{a}}}_{LoS} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \hat{\rho} \\ \hat{\mathbf{v}}_{LoS} \\ \hat{\mathbf{a}}_{LoS} \end{bmatrix} \quad (III-128)$$

The acceleration is assumed to be constant.

All deviations from this constant acceleration assumption are in this case the error dynamic, the error state Kalman filter has to deal with.

The remaining error components are summarized in the following equation.

$$\delta \tilde{\tau} = \delta t_{clk} - \delta t_{Sv} + I + T + n_{\delta \tau} + \delta \tau_{dyn} \quad (III-129)$$

$$\begin{aligned}
\delta t_{clk} &:= \text{receiver clock error [s]} \\
\delta t_{sv} &:= \text{satellite clock error [s]} \\
I &:= \text{ionosphere error [s]} \\
T &:= \text{troposphere error [s]} \\
n_{\delta\tau} &:= \text{noise error [s]} \\
\delta\tau_{dyn} &:= \text{dynamic model error [s]}
\end{aligned}
\tag{III-130}$$

The noise error together with the dynamic model error are by far the biggest error components and at the same time, having the highest dynamic.

The dynamic and error variance of the other error components is much lower. That's why the error state filter only cares about these two error components.

At this point, the clock error is not treated by the error state Kalman filter. On the one hand, the clock error is not observable at a scalar tracking stage. On the other hand, the dynamic and error variance of the clock error is much lower compared to the other error components, at least in case of unaided architecture.

The following equation gives the error state dynamic model.

$$\begin{aligned}
\delta \dot{\mathbf{z}} &= \mathbf{A}_{ES} \cdot \delta \mathbf{z} + \mathbf{n}_{ES} \\
\begin{bmatrix} \delta \dot{\rho} \\ \delta \dot{\nu}_{LoS} \\ \delta \dot{\mathbf{a}}_{LoS} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_{a,LoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta \hat{\rho} \\ \delta \hat{\nu}_{LoS} \\ \delta \hat{\mathbf{a}}_{LoS} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \end{bmatrix}
\end{aligned}
\tag{III-131}$$

The system dynamic matrix used for discrete realization gets

$$\Phi_{ES} = \mathbf{I}^{3 \times 3} + \mathbf{A}_{ES} \cdot T_{corr}
\tag{III-132}$$

The used error dynamic model is the same as it was already used for scalar total state tracking. The acceleration error is modeled as Gauss Markov process 1st order with constant mean.

In section III - 3 on scalar total state tracking, the acceleration was also modeled as a Gauss Markov process 1st order with constant mean.

The main difference between the total state and error state approach is the validity in the Kalman filter, assuming that the measurement error dynamic equals a zero mean Gauss Markov process. In case of the total state approach, this assumption is not fully true, whereas in case of the error state approach it is.

The correlation time and variance of the modeled acceleration error are determined by using the already introduced scenario matched tuning. The discriminator error variance $\sigma_{\delta\tau}^2$ is determined

by using $\left. \frac{C}{N} \right|_{IF}$ from the link budget model according to IV - 2.7 .

The measurement error variance matrix for error state scalar optimal filter gets

$$\mathbf{R}_{DLL} = \sigma_{n,\delta r}^2 \quad (\text{III-133})$$

For calculating the model error covariance matrix, the dynamic error of the applied line of sight dynamic model must be considered.

Despite here an error state approach is considered, the error dynamic of the acceleration equals the total state line of sight acceleration discussed in section III - 3 , because within the external total state dynamic model \mathbf{A}_{TS} , the acceleration is modeled as constant. All deviations of the acceleration from this constant value manifest within the error state dynamic model.

Therefore, the acceleration error $\delta \hat{\mathbf{a}}_{LoS}$ can be modeled as Gauss Markov process, similar to 0 with equal error variance and correlation time constant.

The model error covariance matrix gets

$$\mathbf{Q}_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sigma_{n,aLoS}^2 \cdot \left(\frac{\Delta T^2}{2} - \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}} \right) & \sigma_{n,aLoS}^2 \cdot \left(\Delta T - \frac{\Delta T^2}{\tau_{a,LoS}} + \frac{\Delta T^3}{3 \cdot \tau_{a,LoS}^2} \right) \end{bmatrix} \quad (\text{III-134})$$

The observation matrix for the error state filter can be written as

$$\mathbf{h}_F^T = \begin{bmatrix} 1 \\ c \\ 0 \\ 0 \end{bmatrix} \quad (\text{III-135})$$

For setting up a closed loop system, the estimated error state must be used to update the total states as given below.

$$\mathbf{z} = \mathbf{z} + \delta \mathbf{z} \quad (\text{III-136})$$

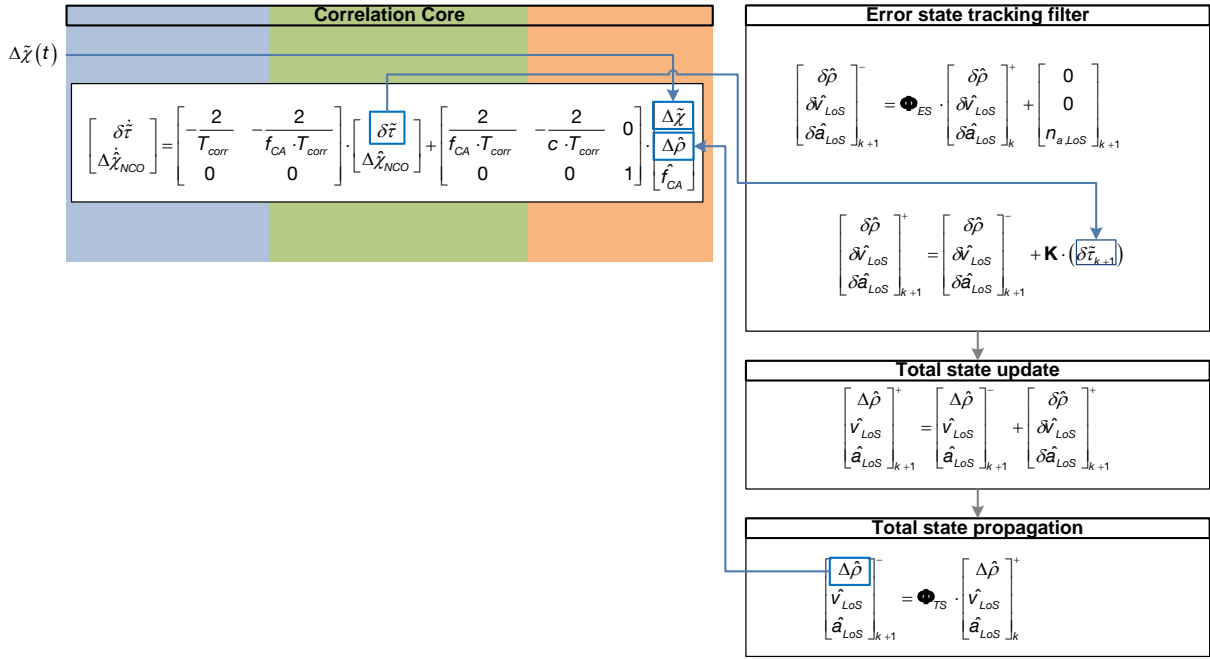
After updating the total states, the error states are set to zero.

$$\delta \mathbf{z} = \mathbf{0} \quad (\text{III-137})$$

III - 5.3 Equivalent base band dynamic state space simulation

III - 5.3.1 Distributed state space realization

The distributed state space realization consists in this case out of 4 components. These are the correlation core in state space form, the error state filter, the total state propagation and the total state update. The following architecture shows these 4 components together with their interconnections.



III - 5.3.2 Centralized state space representation and bandwidth calculation

For closed loop bandwidth calculation, a centralized state space realization of the above system is derived.

For that, the 4 separated components above are combined into one centralized state space matrix.

$$\dot{\mathbf{z}}_{CS} = \mathbf{A}_{CS} \cdot \mathbf{z}_{CS} + \mathbf{B}_{CS} \cdot u_{CS} \quad (\text{III-138})$$

$$\begin{bmatrix} \delta \dot{\tau} \\ \Delta \dot{\chi}_{NCO} \\ \delta \dot{\rho} \\ \delta \dot{v}_{LoS} \\ \delta \dot{a}_{LoS} \\ \Delta \dot{\rho} \\ \dot{v}_{LoS} \\ \dot{a}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{2}{T_{corr}} & -\frac{2}{f_{CA} \cdot T_{corr}} & 0 & 0 & 0 & -\frac{2}{c \cdot T_{corr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k_{11}}{T_{corr}} & 0 & \frac{-1}{T_{corr}} & 1 & 0 & 0 & 0 & 0 \\ \frac{k_{21}}{T_{corr}} & 0 & 0 & \frac{-1}{T_{corr}} & 1 & 0 & 0 & 0 \\ \frac{k_{31}}{T_{corr}} & 0 & 0 & 0 & \frac{-1}{\tau_{aLoS}} - \frac{1}{T_{corr}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{T_{corr}} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{T_{corr}} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{1}{T_{corr}} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi}_{NCO} \\ \delta \hat{\rho} \\ \delta \hat{v}_{LoS} \\ \delta \hat{a}_{LoS} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi} \quad (\text{III-139})$$

The state update of the total states and the state reset of the error states after the update, are two particularities which need to be modeled within the centralized state space matrix.

The blue matrix entries in previous equation realize the state update. The red entries realize the error state reset.

The derived centralized state space form can be used for calculating the closed loop tracking bandwidth.

Because the error states are reset after updating the total states, it is possible to get a shorter form of the centralized state space representation, without mentioning the error states. The effect of the error states is already included in the Kalman gains.

$$\Delta \dot{\hat{\rho}} = \hat{v}_{LoS} + \frac{1}{T_{corr}} \cdot \delta \hat{\rho} = \hat{v}_{LoS} + \frac{k_{11}}{T_{corr}} \cdot \delta \tilde{\tau} \quad (\text{III-140})$$

The equation above shows the example relation between the total state pseudorange change and the error state pseudorange. Because the error states are reset, there is no error state propagation and $\delta \tilde{\tau}$ can directly be used to update the pseudorange estimation. The reduced centralized state space representation gets:

$$\begin{bmatrix} \delta \dot{\tilde{\tau}} \\ \Delta \dot{\hat{\chi}}_{NCO} \\ \Delta \dot{\hat{\rho}} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{2}{T_{corr}} & -\frac{2}{f_{CA} \cdot T_{corr}} & -\frac{2}{c \cdot T_{corr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{k_{11}}{T_{corr}} & 0 & 0 & 1 & 0 \\ \frac{k_{21}}{T_{corr}} & 0 & 0 & 0 & 1 \\ \frac{k_{31}}{T_{corr}} & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} \quad (\text{III-141})$$

The interesting path through the system is

$$\Delta \tilde{\chi} \rightarrow \Delta \hat{\rho} \quad (\text{III-142})$$

In order to norm the bode diagram, the output variable is transformed to code phase.

$$\Delta \tilde{\chi} \rightarrow \Delta \hat{\rho} \cdot \frac{f_{C/A}}{c} \quad (\text{III-143})$$

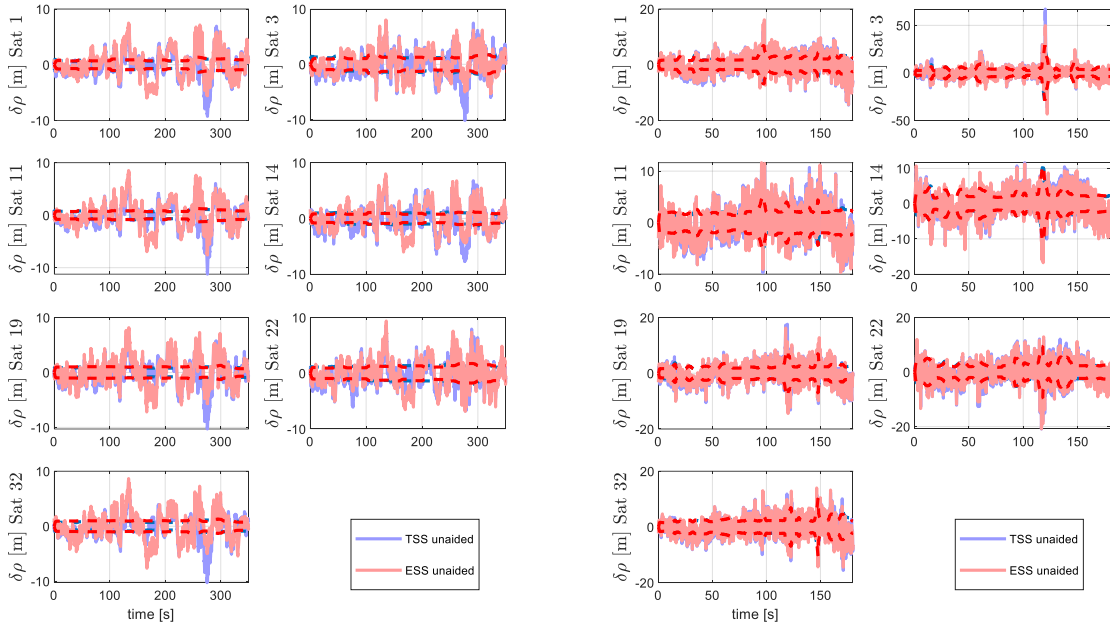
The transfer function is calculated according to the next equation

$$\begin{aligned} \mathbf{G}_{\Delta \rho} &= \mathbf{c}_{\Delta \rho}^T \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A}_{CS})^{-1} \cdot \mathbf{b}_{\Delta \tilde{\chi}} \\ \mathbf{c}_{\Delta \rho}^T &= \begin{bmatrix} 0 & 0 & \frac{1}{\lambda_{C/A}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{b}_{\Delta \tilde{\chi}}^T &= \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (\text{III-144})$$

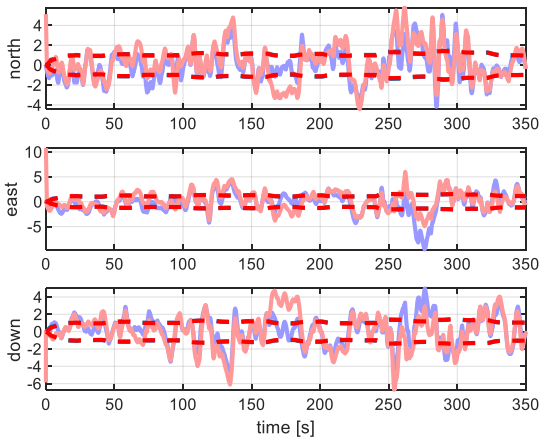
III - 5.4 Scalar total state vs. scalar error state tracking

The following simulation scenarios compare the scalar total state unaided architecture and the scalar error state unaided architecture.

UNJAMMED

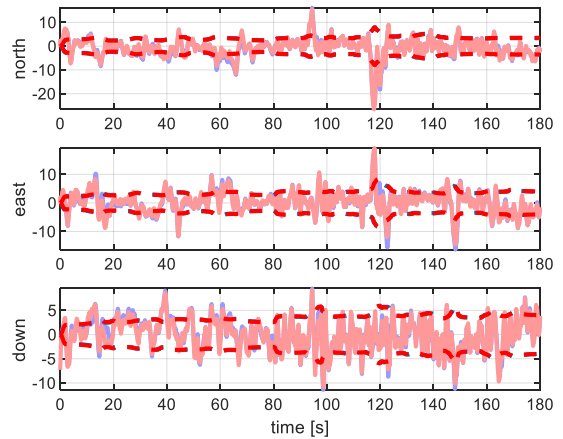


III-87 pseudorange error - DA42 - unjammed

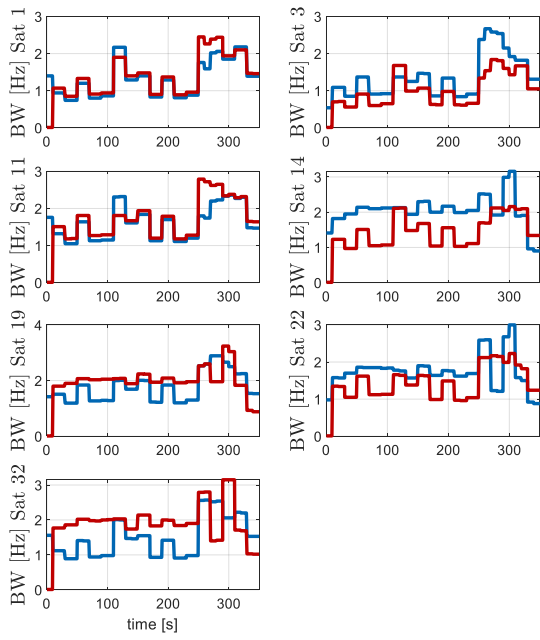


III-89 positioning error - DA42 - unjammed

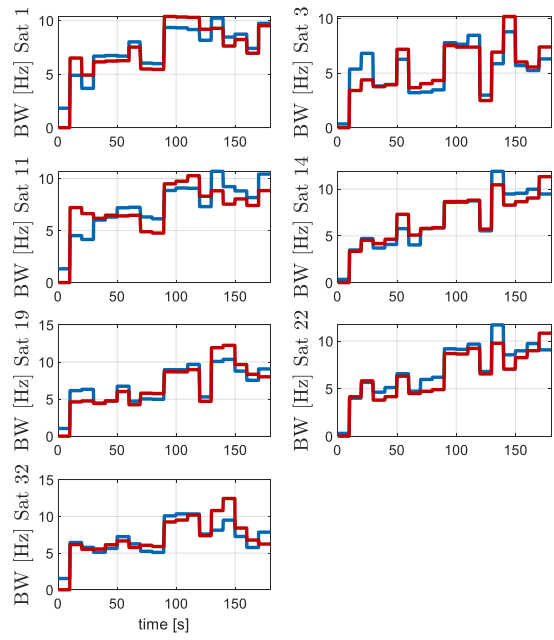
III-88 pseudorange error - high dynamic - unjammed



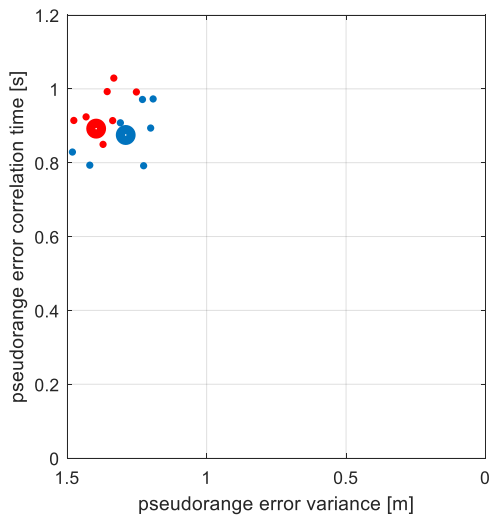
III-90 positioning error - high dynamic - unjammed



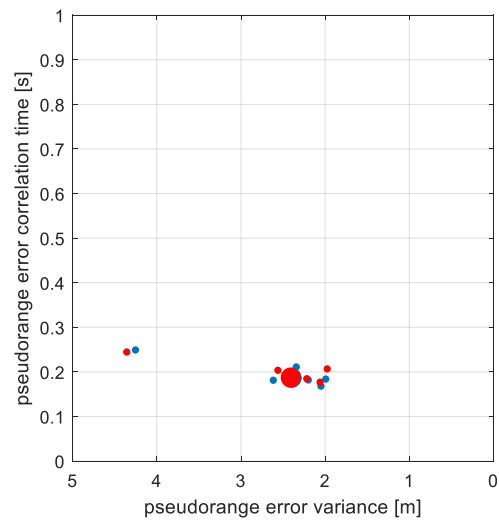
III-91 bandwidth - DA42 - unjammed



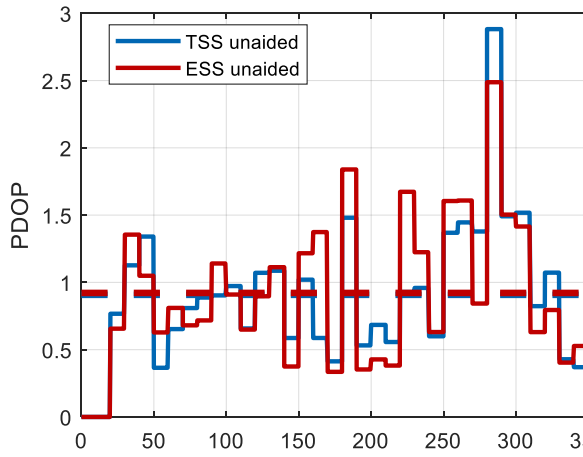
III-92 bandwidth - high dynamic - unjammed



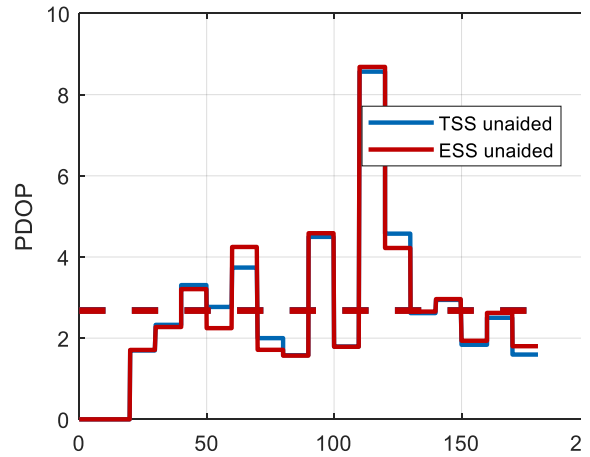
III-93 Comparing raw date - DA42 - unjammed



III-94 Comparing raw date - high dynamic - unjammed

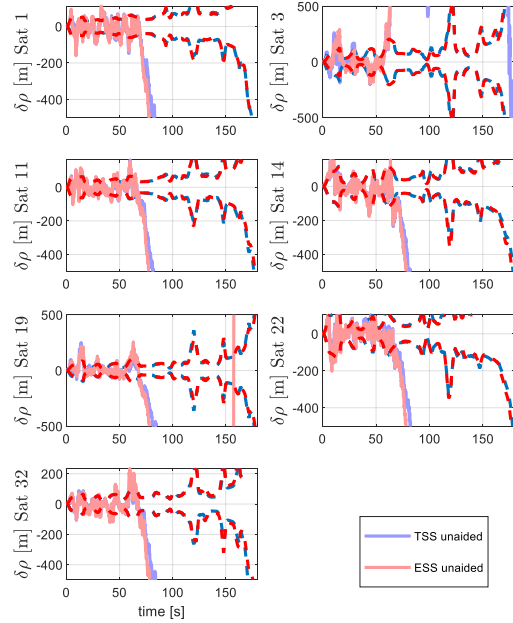
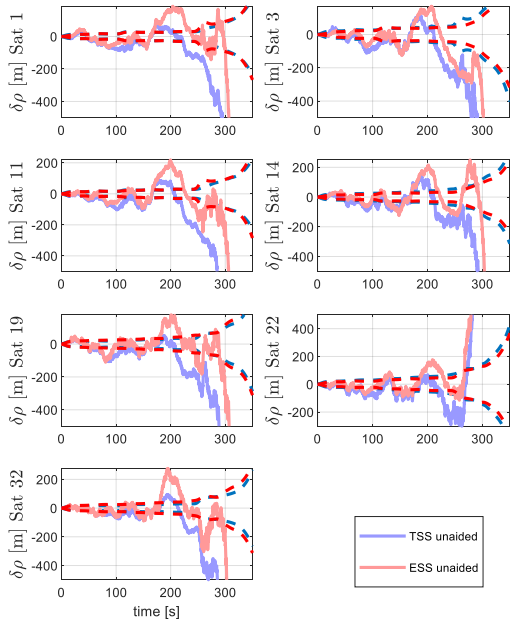


III-95 PDOP - DA42 - unjammed



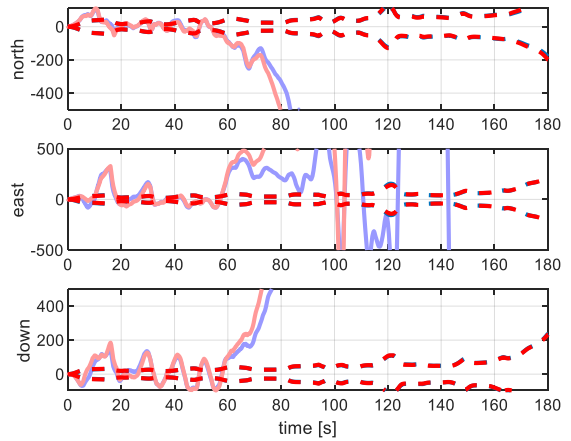
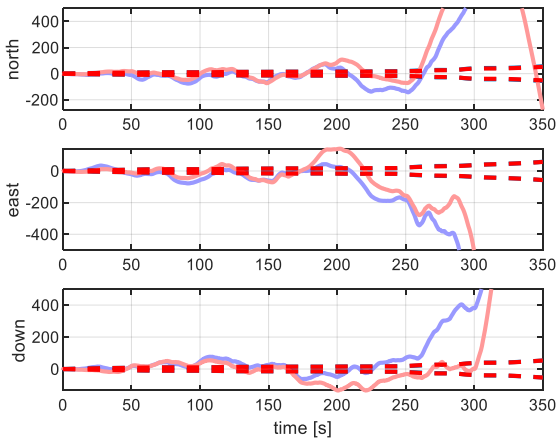
III-96 PDOP - high dynamic - unjammed

JAMMED



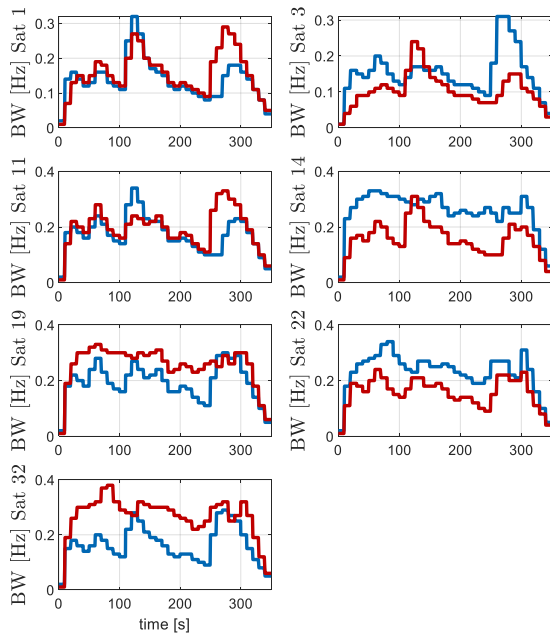
III-97 pseudorange error - DA42 - jammed

III-98 pseudorange error - high dynamic - jammed

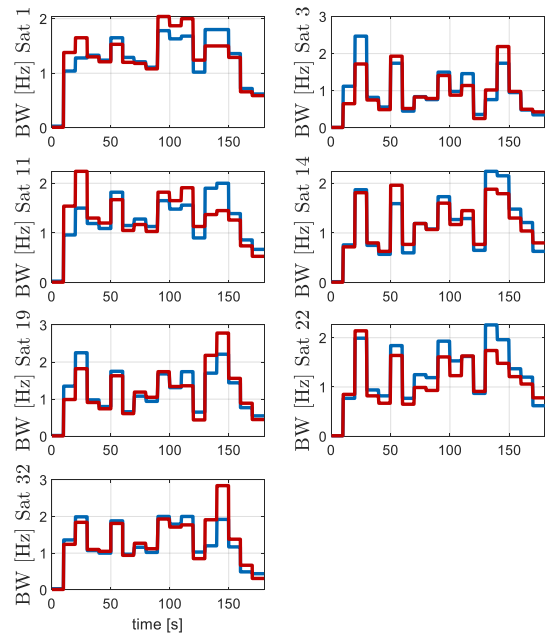


III-99 positioning error - DA42

III-100 positioning error - high dynamic



III-101 bandwidth - DA42



III-102 bandwidth - high dynamic

CONCLUSION

This section shows, for scalar unaided tracking, an error state realization provides no advantages compared to the total state approach. The error state tracking filter still needs to track the full line of sight dynamic. The additional implemented total state dynamic propagation, outside of the error state filter, only assumes a constant line of sight acceleration. All deviations from this constant line of sight acceleration need to be tracked by the error state filter. Without aiding, there is no possibility to predict the actual line of sight dynamic.

III - 6 Positioning solution using optimal filter

III - 6.1 Motivation

This thesis compares different tracking architectures, which are scalar architectures and also vector tracking architectures. Vector tracking architectures provide in case of position state formulation already the final positioning solution. In order to compare scalar tracking architectures with vector tracking architectures, an optimal filter-based positioning solution for scalar tracking architectures is developed in this section. For the positioning solution in the previous section, already the approach from this section was used.

A positioning solution would be also possible by using a Newton method based solving of the nonlinear relation between pseudoranges and position [2, p. 258]. But optimal filter-based positioning solution shows a better performance compared to classical positioning solution. Moreover, the optimal filter-based approach is more comparable to vector tracking which delivers inherently a positioning solution – also by using optimal filters.

There are already many papers on optimal filter-based positioning solution, like [49], [50], [51], [52] and [53].

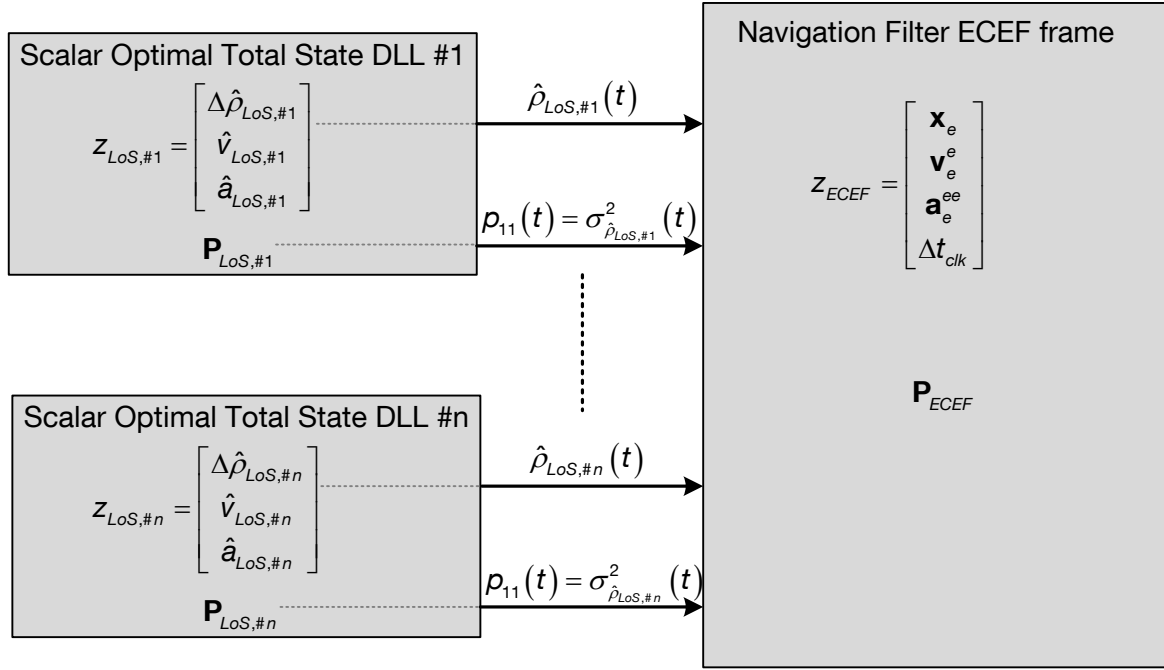
The contribution of this section is the coupling of optimal scalar filters with an optimal positioning filter. The provided raw data error variances of the scalar optimal filter state error covariance matrices are directly used within the measurement error variance of the positioning filter.

Additionally, the novel concept of scenario matched tuning, which was introduced in the context of scalar optimal filter tracking, will be also applied to positioning solution.

III - 6.2 Filter architecture

As positioning filter, a Kalman Filter is used to calculate the positioning solution and estimate the clock error, by using the available raw data from scalar tracking filters. This filtered solution approach offers a huge advantage, especially if it is used in combination with scalar optimal filter-based tracking loops, as they were introduced above.

The following figure shows the corresponding architecture.



III-103 Filtered navigation solution

Each scalar optimal tracking filter provides the actual pseudorange together with the inherently available error covariance information. (This error covariance information can be taken from the state error covariance matrix \mathbf{P} of the local Kalman filter).

Filtered navigation solution in ECEF frame

In the following, the equations for filtered navigation solution based on a Kalman filter, will be derived. The navigation solution can be calculated by using the available pseudoranges from the local filters. If a frequency aided code tracking loop or carrier aided code tracking loop is used, the navigation solution can also be calculated by using the pseudoranges and the range rates. The following equations assume a pure code tracking approach. Because this thesis is about comparing different architectures, only code tracking is considered.

The dynamic model used within the navigation filter is given according to [2, p. 264]. The dynamic model is written in coordinates of ECEF frame.

$$\dot{\mathbf{z}}_{ECEF} = \mathbf{A}_{ECEF} \cdot \mathbf{z}_{ECEF} + \mathbf{n}_{ECEF} \quad (\text{III-145})$$

$$\mathbf{z}_{ECEF} = \begin{bmatrix} \mathbf{x}_e \\ \mathbf{v}_e^e \\ \mathbf{a}_e^{ee} \\ c \cdot \delta t_{clk} \\ c \cdot \delta f_{clk} \end{bmatrix} \quad \mathbf{A}_{ECEF} = \begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_{33} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 1 \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 0 \end{bmatrix} \quad \mathbf{n}_{ECEF} = \begin{bmatrix} \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{n}_{\mathbf{a}_e^{ee}}^{3 \times 1} \\ n_{c \cdot \delta t, clk} \\ n_{c \cdot \delta f, clk} \end{bmatrix} \quad (\text{III-146})$$

The platform acceleration in ECEF coordinate axis will be modeled as a Gauss Markov process with constant mean. Furthermore, the acceleration in x, y and z direction of the ECEF coordinate axis will be assumed to be uncorrelated.

$$\mathbf{A}_{33} = \begin{bmatrix} -\frac{1}{\tau_{a_{e,x}^{ee}}} & 0 & 0 \\ 0 & -\frac{1}{\tau_{a_{e,y}^{ee}}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{a_{e,z}^{ee}}} \end{bmatrix} \quad (\text{III-147})$$

The clock error is modeled as a second order process according to equation (VII-2).

Scenario matched tuning

The driving noise for the Gauss Markov process $\mathbf{n}_{a_{e}^{ee}}$, together with the correlation time constants in equation (III-146) are derived by applying the already introduced scenario matched tuning.

The mean value of the accelerations towards the coordinate axis of the e-frame is calculated. The deviation from the mean values is modeled as a Gauss Markov process. The correlation time and variance of each process are determined by analyzing the pre mission available platform acceleration dynamic.

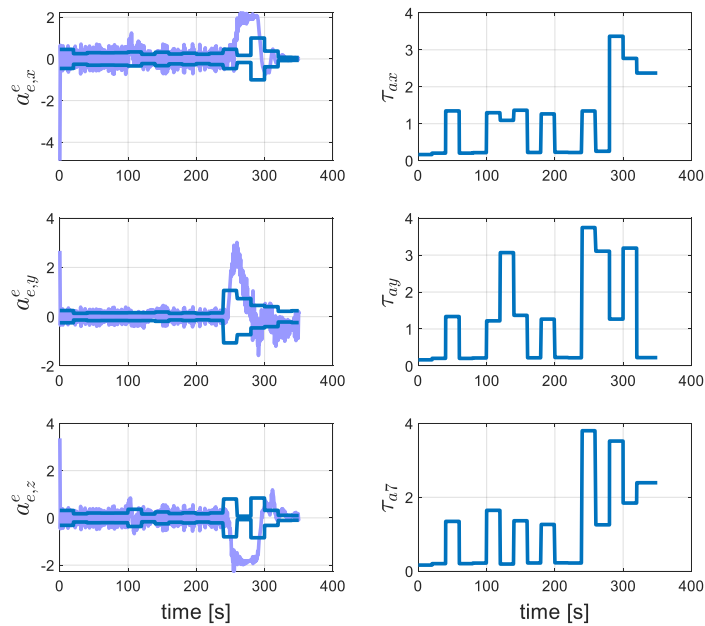
Especially in case of planned missions or even in case of auto landing, the flight trajectories are pre known and the needed variances and correlation time constants can be calculated in advance. The following figure shows the assumed variances and correlation time constants in case of the DA42 approach.

$$\begin{aligned} \sigma_{a_{x,LoS}}^2 &= \mathcal{E} \left\{ \mathbf{a}_{x,LoS}^2(t) - \mathcal{E}^2 \left\{ \mathbf{a}_{x,LoS}(t) \right\} \right\} \\ \sigma_{a_{y,LoS}}^2 &= \mathcal{E} \left\{ \mathbf{a}_{y,LoS}^2(t) - \mathcal{E}^2 \left\{ \mathbf{a}_{y,LoS}(t) \right\} \right\} \\ \sigma_{a_{z,LoS}}^2 &= \mathcal{E} \left\{ \mathbf{a}_{z,LoS}^2(t) - \mathcal{E}^2 \left\{ \mathbf{a}_{z,LoS}(t) \right\} \right\} \end{aligned} \quad (\text{III-148})$$

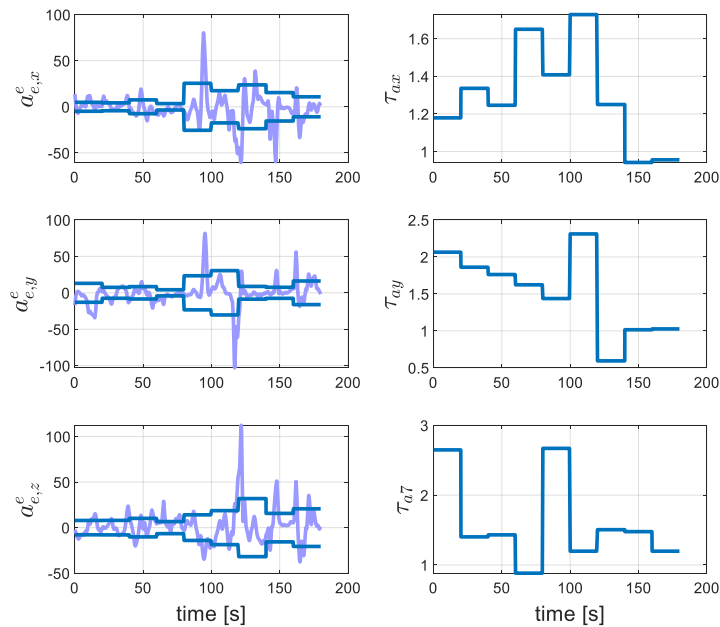
A vital parameter is the window time, used for variance calculation. The acceleration variances from equation (III-148) give constant values for the whole mission. For adaptive tuning, the following variance calculation approach is used.

$$\begin{aligned} \sigma_{a_{x,LoS}}^2(t) &= \left[\mathcal{E} \left\{ \mathbf{a}_{x,LoS}^2(t) - \left[\mathcal{E}^2 \left\{ \mathbf{a}_{x,LoS}(t) \right\} \right]_{t_{startMission}}^{t_{endMission}} \right\} \right]_t^{t+\Delta T} \\ \sigma_{a_{y,LoS}}^2(t) &= \left[\mathcal{E} \left\{ \mathbf{a}_{y,LoS}^2(t) - \left[\mathcal{E}^2 \left\{ \mathbf{a}_{y,LoS}(t) \right\} \right]_{t_{startMission}}^{t_{endMission}} \right\} \right]_t^{t+\Delta T} \\ \sigma_{a_{z,LoS}}^2(t) &= \left[\mathcal{E} \left\{ \mathbf{a}_{z,LoS}^2(t) - \left[\mathcal{E}^2 \left\{ \mathbf{a}_{z,LoS}(t) \right\} \right]_{t_{startMission}}^{t_{endMission}} \right\} \right]_t^{t+\Delta T} \end{aligned} \quad (\text{III-149})$$

The variance is calculated for a moving time window along the mission. The time window for adaptive tuning is selected to 20s. The following figures give the correlation time and variance of the Gauss Markov process, along the trajectory.



III-104 ECEF acceleration characteristics - DA42



III-105 ECEF acceleration characteristics - high dynamic

These correlation time constants, taken from the simulated mission dynamic, are used within matrix

\mathbf{A}_{33} .

The typical clock error can be described using a second order differential equation, as given in state space form in (VII-2). This clock error model is also used in the simulation to generate a corresponding clock error.

The driving noise for the clock error model is chosen according to appendix VII - 1.1 . Using this information, the model error covariance matrix can be written as follows.

$$\mathbf{Q} = \mathcal{E} \{ \mathbf{n} \cdot \mathbf{n}^T \} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{33}^{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & c^2 \cdot \sigma_{\delta t, clk}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 \cdot \sigma_{\delta f, clk}^2 \end{bmatrix}, \quad \mathbf{Q}_{33} = \begin{bmatrix} \frac{2 \cdot \sigma_{ax}^2}{\tau_{ax}} & 0 & 0 \\ \tau_{ax} & & \\ 0 & \frac{2 \cdot \sigma_{ay}^2}{\tau_{ay}} & 0 \\ & \tau_{ay} & \\ 0 & 0 & \frac{2 \cdot \sigma_{az}^2}{\tau_{az}} \\ & & \tau_{az} \end{bmatrix} \quad (\text{III-150})$$

Usage of extended Kalman Filter

Within the state correction

$$\mathbf{z}_{k+1, ECEF}^+ = \mathbf{z}_{k+1, ECEF}^- + \mathbf{K} \cdot (\tilde{\mathbf{p}} - \mathbf{h}(\mathbf{z}_{k+1, ECEF}^-)) \quad (\text{III-151})$$

the innovation $\tilde{\mathbf{p}} - \mathbf{h}(\mathbf{z}_{k+1, ECEF}^-)$ is calculated using a nonlinear relationship.

The nonlinear transformation from the filter states to the observation space is given by the following equation.

$$\delta \tilde{\mathbf{p}} = \tilde{\mathbf{p}} - \mathbf{h}(\mathbf{z}_{k+1, ECEF}^-) = \tilde{\mathbf{p}} - \hat{\mathbf{p}}$$

$$= \begin{bmatrix} \tilde{\rho}_{\#1} \\ \vdots \\ \tilde{\rho}_{\#n} \end{bmatrix} - \begin{bmatrix} \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \mathbf{x}_e(t_r)) + c \cdot \delta t_{clk} - c \cdot \delta t_{\#1}} \\ \vdots \\ \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \mathbf{x}_e(t_r)) + c \cdot \delta t_{clk} - c \cdot \delta t_{\#n}} \end{bmatrix} \quad (\text{III-152})$$

The matrix $\mathbf{C}_{e'}^e$ rotates the satellite position from e-frame at emission time into e-frame at reception time.

In order to set up the Kalman filter observation matrix \mathbf{H} , the nonlinear transformation $\mathbf{h}(\mathbf{z}_{k+1, ECEF}^-)$ is linearized at the every point along trajectory.

$$\delta \tilde{\mathbf{p}} = \mathbf{H} \cdot \delta \hat{\mathbf{z}}_{ECEF}$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \hat{\mathbf{p}}_{\#1}}{\partial \mathbf{z}_{ECEF}} \\ \vdots \\ \frac{\partial \hat{\mathbf{p}}_{\#n}}{\partial \mathbf{z}_{ECEF}} \end{bmatrix}_{\hat{\mathbf{x}}_e, \delta \hat{t}_{clk}} \cdot \delta \mathbf{z}_{ECEF} \quad (III-153)$$

$$= \begin{bmatrix} \frac{-2 \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)^T}{2 \cdot \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)}} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-2 \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)^T}{2 \cdot \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)}} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 1 & 0 \end{bmatrix} \cdot \delta \mathbf{z}_{ECEF}$$

The linearization is justified, because the deviation of $\hat{\mathbf{z}}_{ECEF}$ from the true trajectory is very small. This Kalman filter approach is named „extended Kalman filter“.

The real innovation is calculated using the nonlinear relation

$$\tilde{\mathbf{p}} - \mathbf{h}(\mathbf{z}_{k+1,ECEF}^-) \quad (III-154)$$

Benefits from combining optimal filter scalar loops and optimal filter positioning solution

The usage of optimal scalar tracking loops provides especially with regard to the optimal filter based navigation solution, an advantage. For setting up the measurement error variance matrix \mathbf{R} of the navigation filter, the estimated pseudorange error variances, contained in the state error covariance matrices of the optimal scalar tracking filters, can be used.

$$\sigma_{\rho,\#n}^2 = \rho_{11,\#n} \rightarrow r_{11,\#n} \quad (III-155)$$

Generally, also classical fixed gain tracking loops would provide the pseudorange error variance information, if a parallel error variance propagation according section III - 2.5 is used. But as it was already mentioned, this error variance information considers only errors due to noise. Errors caused by dynamic stress are not included. The pseudorange error variances within the matrix \mathbf{P} of the optimal scalar tracking filters consider both errors very well.

The Kalman filter is optimal only if the measurement errors are white, respective uncorrelated in time space. The pseudorange errors, which are at the same time the measurement errors, are correlated due to the low bandwidth of the scalar tracking loop. In order to take this fact into account, the measurement error can be modeled by additional states as Gauss Markov process. Or

on the other side, the measurement error variances can be simply increased in order to consider the nonwhite measurement errors.

The following figure shows the positioning solution, calculated by using the specified optimal filter based filtered solution.

At this point there will be no simulation because the positioning solutions are considered for every analyzed tracking architecture.

CONCLUSION

Major pros and cons of scalar optimal tracking

Filtered positioning solution	
+	The filtered positioning solution is supported by a dynamic model, which can be selected correspondingly to the actual mission. Due to this model, the solution at every time step benefits from solutions of previous time steps. This additional information makes the solution more robust
+	Scenario matched tuning can directly be applied to the tuning figures of the Kalman filter
+	Optimal filter based scalar tracking loops provide inherently the error covariance information of the pseudoranges. This information can directly be used as measurement error variance in the measurement error variance matrix of the Kalman filter.
+	The provided state error covariance information includes the errors caused by noise and also dynamic stress. (In contrast to the least squares approach for positioning calculation. There the separately calculated state error covariance information does not contain any information on dynamic stress)
-	Not applicable for an initial guess of position after acquisition. For that, the least squares approach is necessary anyway.

III - 7 Total state vector tracking - unaided

III - 7.1 Motivation

In previous sections, only scalar tracking was considered, with one separate tracking channel for each satellite.

This section is about vector tracking, where all satellites are tracked by one tracking filter. This architecture benefits from the fact, that for positioning only 4 satellites are necessary and every further satellite makes an over determined system.

The satellite respectively the line of sight vectors from the receiver to the satellites are not orthogonal. The line of sight dynamic of one satellite is correlated with the line of sight dynamic of another satellite, depending on the mutual elevation. Exactly this advantage, besides others, is exploited by a vector tracking approach. This joint tracking of all satellites and the mutual aiding, help reducing the closed loop tracking bandwidth of each single channel.

Moreover, if for one satellite, loss of lock occurs, the phase reference for the corresponding replique is kept. Immediately after the conditions for this satellite improve, its NCO is synchronized again.

Vector tracking can be realized in error state space or in the total state space. In this section, at first the total space realization will be analyzed.

Some sources for total state vector tracking in literature are given in [22], [54], [1], [55], [19] and [56]. A special total state vector tracking approach is given in [57], using a differential tracking approach. In [58], a combination of an extended Kalman filter-based vector tracking filter and scalar tracking loops is used. Both work in parallel and compared to each other. In case of failures in the scalar loops, the vector loop assists the scalar loops. There is also some literature, comparing some special realizations of vector tracking loops to some selected other tracking approaches as in [59], [60], [61], [62] and [63]. But there is no structured comparison. Moreover, no defined tunings are given in the literature examples.

Nevertheless, this section provides some contributions.

Contribution 1: Development of a centralized state space representation of the whole tracking architecture in equivalent base band, providing the possibility to calculate poles for each tracking channels and also the closed loop tracking bandwidth for each tracking channel.

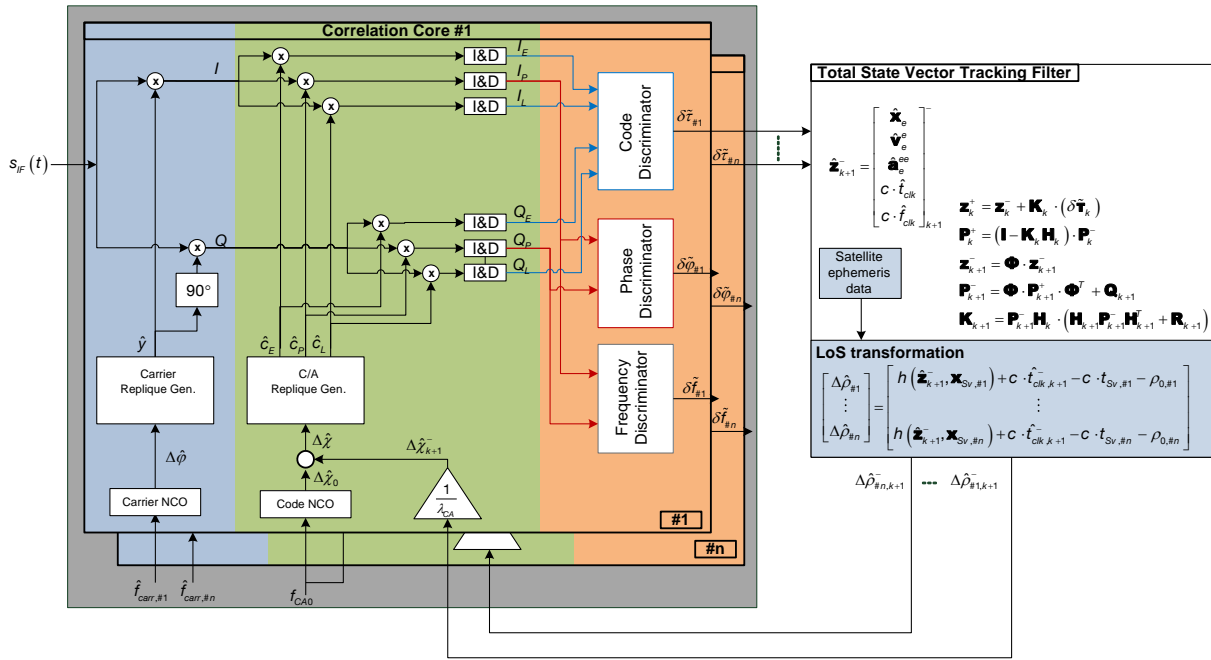
Contribution 2: Application of the already introduced scenario matched tuning to vector tracking filters.

Contribution 3: At the end of this section, in detail the differences between an unaided scalar tracking approach and the here developed unaided total state vector tracking approach will be worked out. The tracking behavior will be compared, given also jammed scenarios.

III - 7.2 Tracking architecture

The total state vector delay locked loop (VDLL) using position state formulation, is the most obvious realization of such a tracking architecture. But having a look into literature, if it is about vector tracking, typically error state tracking architectures are analyzed. Total state vector tracking architectures are quite rare. In [22] a total state vector tracking architecture is shown. But even here, no explicit realization is given. Also, no tuning approach is discussed, nor is any resilient simulation provided for comparing the performance of the total state vector tracking architecture with other tracking architectures. Therefore, at the end of this section, different total state and later, also different error state vector tracking architectures will be compared.

The following figure shows the standard correlation core together with an abstract model of the vector tracking filter. Similar to scalar tracking loops, in vector tracking, each satellite channel has its own correlation core.



III-106 Total state optimal filter-based vector tracking architecture

The vector tracking filter is realized in position state formulation. The platform dynamic is modeled using a second order dynamic, whereas the acceleration is modeled as a zero mean Gauss Markov process. The correlation time constants as well as the driving noise of this Gauss Markov process, describing the platform acceleration, are derived using the already introduced scenario matched tuning. The filter states are with reference to the ECEF coordinate system. For each coordinate direction, an own Gauss Markov process is used. These three processes are assumed to be uncorrelated.

The following equation describes the dynamic model, used in the vector delay locked loop.

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\mathbf{x}}_e \\ \dot{\mathbf{v}}_e \\ \dot{\mathbf{a}}_e^{ee} \\ c \cdot \delta \dot{t}_{clk} \\ c \cdot \delta \dot{f}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{I}_{12}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}_{23}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_a^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 1 \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_e \\ \mathbf{v}_e \\ \mathbf{a}_e^{ee} \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{f}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{n}_a \\ n_{\delta tclk} \\ n_{\delta fclk} \end{bmatrix} \quad (III-156)$$

$$\mathbf{A}_a^{3 \times 3} = \begin{bmatrix} -\frac{1}{\tau_{a,x}} & 0 & 0 \\ 0 & -\frac{1}{\tau_{a,y}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{a,z}} \end{bmatrix} \quad (III-157)$$

The clock error is modeled according to the derived clock error model in VII - 1.1 .

The process error covariance matrix is given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{Q}_{33}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & c^2 \cdot \sigma_{n, \delta t_{clk}}^2 & 0 \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & c^2 \cdot \sigma_{n, \delta f_{clk}}^2 \end{bmatrix}, \quad \mathbf{Q}_a^{3 \times 3} = \begin{bmatrix} \frac{2 \cdot \sigma_{ax}^2}{\tau_{ax}} & 0 & 0 \\ 0 & \frac{2 \cdot \sigma_{ay}^2}{\tau_{ay}} & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{az}^2}{\tau_{az}} \end{bmatrix} \quad (\text{III-158})$$

The discrete process error covariance matrix can be approximated as

$$\mathbf{Q}_k \approx \mathbf{Q} \cdot T_{corr} \quad (\text{III-159})$$

The following sequence describes the closed loop application of the extended Kalman filter, used for signal tracking.

1) State propagation:

$$\hat{\mathbf{z}}_{k+1}^- = (\mathbf{I}^{11 \times 11} + \mathbf{A} \cdot T_{corr}) \cdot \hat{\mathbf{z}}_k^+ \quad (\text{III-160})$$

For state propagation, no linearization is necessary. The dynamic model is already linear.

2) Nonlinear NCO control

The NCO control command is formed by using the estimated receiver position, together with the satellite position and the estimated clock error.

$$\hat{\mathbf{p}}_{k+1}^- = \mathbf{h}(\hat{\mathbf{z}}_{k+1}^-) = \begin{bmatrix} \sqrt{(\mathbf{C}_{e'}^e(\hat{\tau}_{k+1, \#1}^-) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \hat{\mathbf{x}}_{k+1, e}^-)^T \cdot (\mathbf{C}_{e'}^e(\hat{\tau}_{k+1, \#1}^-) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \hat{\mathbf{x}}_{k+1, e}^-) + c \cdot \delta \hat{t}_{k+1, clk}^- - c \cdot \delta \hat{t}_{\#1}^-} \\ \vdots \\ \sqrt{(\mathbf{C}_{e'}^e(\hat{\tau}_{k+1, \#n}^-) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\hat{\tau}_{k+1, \#n}^-) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \hat{\mathbf{x}}_{k+1, e}^-) + c \cdot \delta \hat{t}_{k+1, clk}^- - c \cdot \delta \hat{t}_{\#n}^-} \end{bmatrix} \quad (\text{III-161})$$

Because the signal travel time from the satellite to the receiver is not negligible and vector operations are only possible within the same coordinate system, the satellite position in the e-frame at transmission time must be transformed into the e-frame at reception time. For that the so called Sagnac correction is used.

The Sagnac correction $\mathbf{C}_{e'}^e(\tau_{k, \#1}^+)$ needs the knowledge of signal travel time τ . For calculating the pseudorange estimation and also the signal travel time, additionally the satellite position $\mathbf{x}_{\#n, e'}(t_{\#n, e})$ at signal transmission time $t_{\#n, e}$ is necessary. For a first guess, as transmission time the reception time is used, which is known in the receiver.

$$t_{e, \#n} = t_r \quad (\text{III-162})$$

The reception time is equal for all satellites. With this reception time, the satellite position can be calculated by using the ephemeris data, received within the navigation message.

Using these assumptions, the first guess for the pseudorange estimation is calculated as

$$\begin{aligned} \hat{\mathbf{p}}_{k+1}^- &= \mathbf{h}(\hat{\mathbf{z}}_{k+1}^-) \\ &= \begin{bmatrix} \sqrt{(\mathbf{x}_{\#1,e'}(t_r) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{x}_{\#1,e'}(t_r) - \hat{\mathbf{x}}_{k+1,e}^-) + c \cdot \delta \hat{t}_{k+1,clk}^- - c \cdot \delta t_{\#1}} \\ \vdots \\ \sqrt{(\mathbf{x}_{\#n,e'}(t_r) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{x}_{\#n,e'}(t_r) - \hat{\mathbf{x}}_{k+1,e}^-) + c \cdot \delta \hat{t}_{k+1,clk}^- - c \cdot \delta t_{\#n}} \end{bmatrix} \end{aligned} \quad (\text{III-163})$$

With this first guess, the signal travel time and thus a better approximation of the signal transmission time for the corresponding satellite can be calculated.

$$\hat{t}_{k+1,\#n}^- = \frac{\hat{\mathbf{p}}_{k+1}^-}{c} \quad \rightarrow \quad t_{e,\#n} = t_r - \hat{t}_{k+1,\#n}^-$$

$$\begin{aligned} \hat{\mathbf{p}}_{k+1}^- &= \mathbf{h}(\hat{\mathbf{z}}_{k+1}^-) \\ &= \begin{bmatrix} \sqrt{(\mathbf{c}_{e'}^e(\hat{t}_{k+1,\#1}^-) \cdot \mathbf{x}_{\#1,e'}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{c}_{e'}^e(\hat{t}_{k+1,\#1}^-) \cdot \mathbf{x}_{\#1,e'}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-) + c \cdot \delta \hat{t}_{k+1,clk}^- - c \cdot \delta t_{\#1}} \\ \vdots \\ \sqrt{(\mathbf{c}_{e'}^e(\hat{t}_{k+1,\#n}^-) \cdot \mathbf{x}_{\#n,e'}(t_{e,\#n}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{c}_{e'}^e(\hat{t}_{k+1,\#n}^-) \cdot \mathbf{x}_{\#n,e'}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-) + c \cdot \delta \hat{t}_{k+1,clk}^- - c \cdot \delta t_{\#n}} \end{bmatrix} \end{aligned} \quad (\text{III-164})$$

Now again, the estimated pseudorange $\hat{\mathbf{p}}_{k+1}^-$ is used, to calculate an even better approximation for the signal travel time. This improved signal travel time is again used for transmission time calculation, which in turn is used in a following iteration for pseudorange estimation and so on. Usually 3 to 5 iterations are enough to get a very good result for the pseudorange.

This estimated pseudorange can now be used in the feedback path to control the NCO in order to keep the replicated code phase aligned with the received code phase.

For direct phase control, only the relative pseudorange change $\Delta \hat{\mathbf{p}}_{k+1}^-$ since the start of tracking is needed.

$$\begin{aligned} \Delta \hat{\mathbf{p}}_{k+1}^- &= \hat{\mathbf{p}}_{k+1}^- - \hat{\mathbf{p}}_0 \\ \begin{bmatrix} \Delta \hat{\chi}_{k+1,\#1}^- \\ \vdots \\ \Delta \hat{\chi}_{k+1,\#n}^- \end{bmatrix} &= \frac{1}{\lambda_{C/A}} \cdot \Delta \hat{\mathbf{p}}_{k+1}^- \end{aligned} \quad (\text{III-165})$$

The initial pseudorange $\hat{\mathbf{p}}_0$ is calculated by the navigation processor. This first pseudorange is more or less a first guess, which is calculated based on the transmission time (being coded in the navigation message) and the reception time. There is no clock error correction.

3) Update step

The classical update equation is given below

$$\hat{\mathbf{z}}_{k+1}^+ = \hat{\mathbf{z}}_{k+1}^- + \mathbf{K}_{k+1} \cdot \left(\begin{bmatrix} \Delta \tilde{\tau}_{k+1,\#1} \\ \vdots \\ \Delta \tilde{\tau}_{k+1,\#n} \end{bmatrix} - \frac{1}{c} \cdot \mathbf{h}(\hat{\mathbf{z}}_{k+1}^-) \right) \quad (\text{III-166})$$

Because the Kalman filter is used in a feedback system, the innovation is calculated within the correlation core. The measurements $\delta \tilde{\mathbf{T}}_{k+1}$ are already the innovation.

$$\hat{\mathbf{z}}_{k+1}^+ = \hat{\mathbf{z}}_{k+1}^- + \mathbf{K}_{k+1} \cdot \begin{bmatrix} \delta \tilde{\tau}_{k+1,\#1} \\ \vdots \\ \delta \tilde{\tau}_{k+1,\#n} \end{bmatrix} \quad (\text{III-167})$$

The transformation of the Kalman filter state into measurement space is in this approach a nonlinear transformation. For calculation of the Kalman gains, the measurement model is linearized about the actual position and clock error estimation.

$$\mathbf{H}_{k+1} = \frac{1}{c} \cdot \left. \frac{\partial \mathbf{h}(\hat{\mathbf{z}}_{k+1}^-)}{\partial \hat{\mathbf{z}}_{k+1,e}^-} \right|_{\hat{\mathbf{z}}_{k+1,e}^-} \cdot \delta \mathbf{z} \quad (\text{III-168})$$

This approach assumes that the deviation $\delta \hat{\mathbf{x}}_{k+1,e}^-$ between the true position \mathbf{x}_e and the estimated position $\hat{\mathbf{x}}_{k+1,e}^-$ is very small. If this assumption does not apply, this could lead to divergent filter states.

$$\mathbf{H}_{k+1} = \frac{1}{c} \cdot \begin{bmatrix} \frac{-2 \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)^T}{2 \cdot \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e}(t_{e,\#1}) - \hat{\mathbf{x}}_{k+1,e}^-)}} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-2 \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)^T}{2 \cdot \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e}(t_{e,\#n}) - \hat{\mathbf{x}}_{k+1,e}^-)}} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 1 & 0 \end{bmatrix} \quad (\text{III-169})$$

As tracking error measurement in this approach, the discriminator outputs are used, which give the tracking error as time error in seconds, between the replicated C/A code and the received one.

$$\delta \tilde{\mathbf{T}} = \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#n} \end{bmatrix} = \begin{bmatrix} \delta \tau_{\#1} - \delta t_{clk} + \delta t_{Sv,\#1} + l + T + e + n_{\delta \tau,\#1} \\ \vdots \\ \delta \tau_{\#n} - \delta t_{clk} + \delta t_{Sv,\#n} + l + T + e + n_{\delta \tau,\#n} \end{bmatrix} \quad (\text{III-170})$$

Within the measurement error covariance matrix \mathbf{R} , only the noise like error parts $n_{\delta \tau}$ are considered. The measurement errors caused by ephemeris errors e , receiver clock error δt_{clk} , satellite clock error δt_{sv} , ionosphere and troposphere error, are considered by an own error model

within the state space model of the Kalman filter. This is necessary because these error components are nonwhite. The Kalman filter is an optimal filter only for white measurement errors. The corresponding nonwhite, respectively time correlated errors must be decorrelated by a corresponding error model. Therefore, a first approach for the measurement error covariance matrix is

$$\mathbf{R} = \begin{bmatrix} \sigma_{n,\delta\tau\#1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{n,\delta\tau\#n}^2 \end{bmatrix} \quad (\text{III-171})$$

The channel wise measurement error variances are derived pre mission by using link budget models and the discriminator error model given in IV - 2.7 . Even online tuning is possible, by measuring the actual signal to noise ratio and using it in the discriminator error variance formula.

III - 7.3 Error modeling and tuning

The already introduced scenario matched tuning can also be used for optimal tuning of vector tracking loops. This is a tremendous advantage in case of planned missions or known flight trajectories. Platform dynamic and signal to noise environment can be used to determine the tuning setup for the vector tracking loops.

As was already mentioned, the platform acceleration is modeled as a zero mean Gauss Markov process. It's variance and correlation time constants are determined by using scenario matched tuning, according to equation (III-149).

For calculation of the correlation time constants as well as the variances, a time window has to be defined, for which the derived constants are calculated. The length of this time window is also a vital figure. As the following two examples for the DA42 approach show, a short time window of 5s leads to shorter correlation times and also smaller variances. In contrast, a time window of 20s gives longer correlation times and higher variances. It's up to the filter designer, to select the appropriate time window.

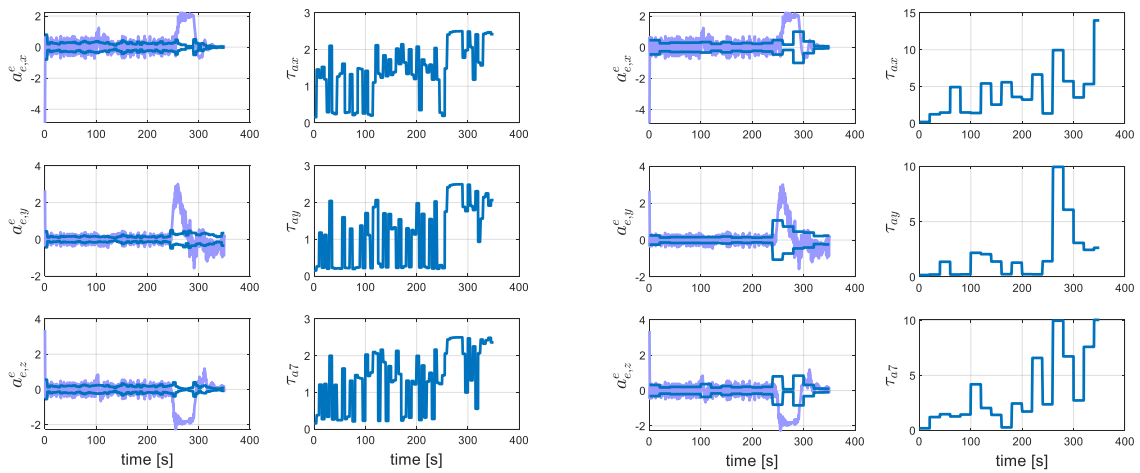
Calculating the Allan deviation might help to identify the correlation time constants and error variances, predominantly occur in the acceleration dynamic. The Allan deviation gives the process variance, depending on the considered correlation time window, [64].

The principle of the Allan deviation can also be used to calculate a kind of Allan correlation time.

$$\begin{aligned} \sigma_a^2(T_{window}) \\ \tau_a(T_{window}) \end{aligned} \quad (\text{III-172})$$

The measurement error variances are determined similar to the scalar code tracking loops, by using transmission path models and signal to noise power predictions.

The tuning parameters, describing the clock error in equation (III-158), are selected according to the clock error model in VII - 1.1 .



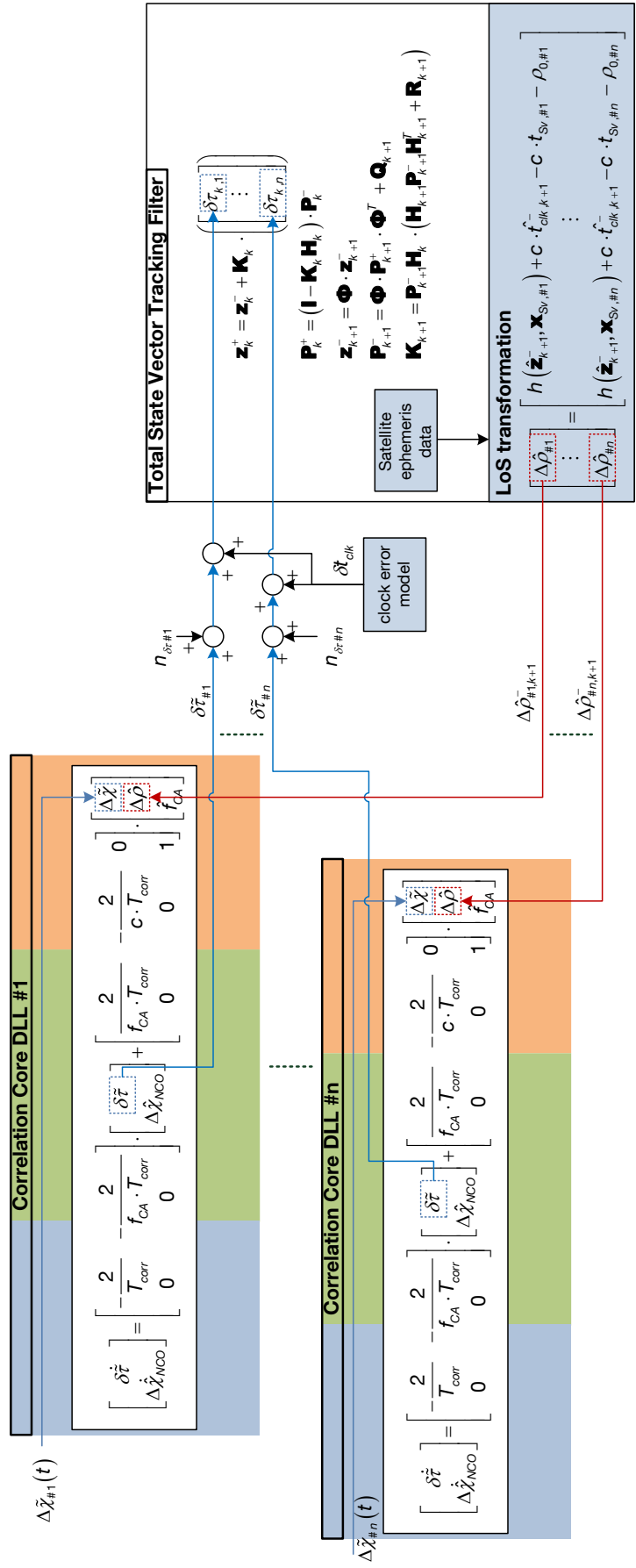
III-107 Acceleration characteristics DA42 (Twindow = 5s)

III-108 Acceleration characteristics DA42 (Twindow = 20s)

III - 7.4 Equivalent base band dynamic state space simulation

III - 7.4.1 Distributed state space realization

This section derives an equivalent base band solution for the given total state vector tracking architecture. The following figure shows the corresponding architecture.



Each satellite has its own correlation core. Whereby the NCO control commands, respectively direct phase controls, are derived by the vector tracking filter.

The difference between real tracking and equivalent base band simulation is only in the correlation core. The tracking filter itself, which is realized by the Kalman filter, is the same.

Even in equivalent base band simulation, the necessary Kalman filter equations are executed for online calculation of the Kalman gain and state error covariance matrix.

Besides the pure tracking, also the error behavior can be evaluated. For that the discriminator noise and the clock error is stimulated at the output of the discriminator.

III - 7.4.2 Centralized state space realization

The concept of a centralized state space representation was already introduced in the context of optimal scalar tracking loops. In this section, a centralized state space representation of the total state vector tracking loop in position state formulation will be derived.

The state space form is set up by combining the different components of the distributed state space formulation, together with all couplings, into one 1st order differential equation system.

The vector tracking loop poses a MIMO system. One main benefit of the centralized state space form is calculating the closed loop tracking bandwidth for a selected single path through the MIMO system. For that, the matrix differential equation system is transformed into Laplace space.

In [65] already an approach is described for deriving a transfer function of a vector delay locked loop and a vector frequency locked loop by writing the vector and frequency tracking loop as a parametric model. But the derived model does not consider the correlation core.

Due to the selected position state formulation, the transformation from the estimated position to the needed pseudorange for NCO, respectively direct phase control, is a nonlinear operation.

In the following, the nonlinear coupled differential equation system is given, describing the closed loop vector tracking in state space form.

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{F}(\mathbf{z}, \mathbf{u}(t), \mathbf{K}(t), \mathbf{x}(t)_{sv \#1}, \dots, \mathbf{x}(t)_{sv \#n}) + \mathbf{n} \\ \mathbf{u}^T &= [\Delta\tilde{\chi}_{\#1} \quad \dots \quad \Delta\tilde{\chi}_{\#N}] \end{aligned} \tag{III-173}$$

The nonlinear differential equation system is time dependent, due to the nonlinear transformation from position to pseudoranges and also due to time dependent Kalman gains. The Kalman gains, used within the centralized state space representation, are derived by calculating the distributed state space simulation. The settling Kalman gains are stored and used within the centralized state space formulation.

The time dependence of the Kalman gains is not explicitly written in the following equations.

The following equations define the 1st order differential equations of the centralized state space form.

$$\begin{aligned}
\delta \dot{\tilde{\tau}}_{\#1} &= -\frac{2}{T_{corr}} \cdot \delta \tilde{\tau}_{\#1} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta \hat{X}_{NCO,\#1} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta \tilde{\chi}_{\#1} - \frac{2}{c \cdot T_{corr}} \cdot (h(\mathbf{x}_{Sv\#1}(t), \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}(t), \hat{\mathbf{x}}_{e,0}) + c \cdot \hat{t}_{clk}) \\
\Delta \dot{\hat{X}}_{NCO,\#1} &= 0 \\
&\vdots \\
\delta \dot{\tilde{\tau}}_{\#N} &= -\frac{2}{T_{corr}} \cdot \delta \tilde{\tau}_{\#N} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta \hat{X}_{NCO,\#N} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta \tilde{\chi}_{\#N} - \frac{2}{c \cdot T_{corr}} \cdot (h(\mathbf{x}_{Sv\#N}(t), \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#N,0}(t), \hat{\mathbf{x}}_{e,0}) + c \cdot \hat{t}_{clk}) \\
\Delta \dot{\hat{X}}_{NCO,\#n} &= 0
\end{aligned} \tag{III-174}$$

$$\begin{aligned}
\dot{\hat{\mathbf{x}}}_e &= \begin{bmatrix} \frac{k_{11}}{T_{corr}} & 0 & \dots & \frac{k_{1N}}{T_{corr}} & 0 \\ \frac{k_{21}}{T_{corr}} & 0 & \dots & \frac{k_{2N}}{T_{corr}} & 0 \\ \frac{k_{31}}{T_{corr}} & 0 & \dots & \frac{k_{3N}}{T_{corr}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \end{bmatrix} + \mathbf{I}^{3 \times 3} \cdot \hat{\mathbf{v}}_e \\
\dot{\hat{\mathbf{v}}}_e &= \begin{bmatrix} \frac{k_{41}}{T_{corr}} & 0 & \dots & \frac{k_{4N}}{T_{corr}} & 0 \\ \frac{k_{51}}{T_{corr}} & 0 & \dots & \frac{k_{5N}}{T_{corr}} & 0 \\ \frac{k_{61}}{T_{corr}} & 0 & \dots & \frac{k_{6N}}{T_{corr}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \end{bmatrix} + \mathbf{I}^{3 \times 3} \cdot \hat{\mathbf{a}}_e^{ee} \\
\dot{\hat{\mathbf{a}}}_e^{ee} &= \begin{bmatrix} \frac{k_{71}}{T_{corr}} & 0 & \dots & \frac{k_{7N}}{T_{corr}} & 0 \\ \frac{k_{81}}{T_{corr}} & 0 & \dots & \frac{k_{8N}}{T_{corr}} & 0 \\ \frac{k_{91}}{T_{corr}} & 0 & \dots & \frac{k_{9N}}{T_{corr}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ \tau_{ax} & -1 & 0 \\ 0 & \tau_{ay} & -1 \\ 0 & 0 & \tau_{az} \end{bmatrix} \cdot \hat{\mathbf{a}}_e^{ee} + \begin{bmatrix} n_{ax} \\ n_{ay} \\ n_{az} \end{bmatrix} \\
\begin{bmatrix} c \cdot \dot{\hat{t}}_{clk} \\ c \cdot \dot{\hat{d}}_{clk} \end{bmatrix} &= \begin{bmatrix} \frac{k_{10,1}}{T_{corr}} & 0 & \dots & \frac{k_{10,N}}{T_{corr}} & 0 \\ \frac{k_{11,1}}{T_{corr}} & 0 & \dots & \frac{k_{11,N}}{T_{corr}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \end{bmatrix} + \begin{bmatrix} c \cdot \hat{d}_{clk} \\ 0 \end{bmatrix} + \begin{bmatrix} n_{tclk} \\ n_{dclk} \end{bmatrix}
\end{aligned} \tag{III-175}$$

The state space differential equation system is linearized about the actual point.

For that, equation (III-174) will be linearized.

$$\begin{aligned}
\dot{\mathbf{z}} &= \mathbf{F}(\hat{\mathbf{z}}, \mathbf{u}, \mathbf{K}, \mathbf{x}_{sv\#1}, \dots, \mathbf{x}_{sv\#n}) + \left. \frac{\partial \mathbf{F}}{\partial \mathbf{z}} \right|_{WP} \cdot \delta \mathbf{z} + \left. \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right|_{WP} \cdot \delta \mathbf{u} + \mathbf{n} \\
\dot{\mathbf{z}} - \mathbf{F}(\hat{\mathbf{z}}, \mathbf{u}, \mathbf{K}, \mathbf{x}_{sv\#1}, \dots, \mathbf{x}_{sv\#n}) &= \delta \dot{\mathbf{z}} = \mathbf{A} \cdot \delta \mathbf{z} + \mathbf{B} \cdot \delta \mathbf{u} + \mathbf{n}
\end{aligned} \tag{III-176}$$

At every time step, the linearization is done about the linearization point (WP) given as

$$\mathbf{WP} = \hat{\mathbf{z}}, \mathbf{u}, \mathbf{K}, \mathbf{x}_{sv\#1}, \dots, \mathbf{x}_{sv\#n} \tag{III-177}$$

The linearized closed loop state space representation is given in the next equation.

$$\begin{bmatrix} \delta \hat{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \hat{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \\ \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ C \cdot \hat{t}_{clk} \\ C \cdot \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\delta\tau\#1}^{2 \times 2} & \dots & \mathbf{0}^{2 \times 2} & \mathbf{A}_{h\#1}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}^{2 \times 2} & \dots & \mathbf{A}_{\tau\#n}^{2 \times 2} & \mathbf{A}_{h\#2}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} \\ \mathbf{A}_{kx\#1}^{3 \times 2} & \dots & \mathbf{A}_{kx\#n}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{kv\#1}^{3 \times 2} & \dots & \mathbf{A}_{kv\#n}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} \\ \mathbf{A}_{ka\#1}^{3 \times 2} & \dots & \mathbf{A}_{ka\#n}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_a^{3 \times 3} \\ \mathbf{A}_{kt\#1}^{1 \times 2} & \dots & \mathbf{A}_{kt\#n}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \\ \mathbf{A}_{kd\#1}^{1 \times 2} & \dots & \mathbf{A}_{kd\#n}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \end{bmatrix} \begin{bmatrix} -\frac{2}{c \cdot T_{corr}} \\ 0 \\ \vdots \\ -\frac{2}{c \cdot T_{corr}} \\ 0 \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} \delta \tilde{\tau}_{\#1} \\ \Delta \hat{X}_{NCO,\#1} \\ \vdots \\ \delta \tilde{\tau}_{\#N} \\ \Delta \hat{X}_{NCO,\#N} \\ \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{11}^{2N \times N} \\ \mathbf{0}^{11 \times N} \end{bmatrix} \cdot \begin{bmatrix} \Delta \tilde{X}_{\#1} \\ \vdots \\ \Delta \tilde{X}_{\#N} \end{bmatrix} \quad (\text{III-178})$$

$$\mathbf{A}_{\delta\tau\#1}^{2 \times 2} = \dots = \mathbf{A}_{\delta\tau\#n}^{2 \times 2} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \quad (\text{III-179})$$

$$\mathbf{A}_{h\#1}^{2 \times 3} = \begin{bmatrix} \frac{\partial (h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}))}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_e} = \dots = \begin{bmatrix} -\frac{2}{c \cdot T_{corr}} \cdot \frac{\partial \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)}}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_e} = \begin{bmatrix} -\frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T}{\|\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \quad (\text{III-180})$$

$$\mathbf{A}_{h\#n}^{2 \times 3} = \begin{bmatrix} -\frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#N} - \hat{\mathbf{x}}_e)^T}{\|\mathbf{x}_{Sv\#N} - \hat{\mathbf{x}}_e\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{kx \#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{11}}{T_{corr}} & 0 \\ \frac{k_{21}}{T_{corr}} & 0 \\ \frac{k_{31}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kx \#n}^{3 \times 2} = \begin{bmatrix} \frac{k_{1,N}}{T_{corr}} & 0 \\ \frac{k_{2,N}}{T_{corr}} & 0 \\ \frac{k_{3,N}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kv \#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{41}}{T_{corr}} & 0 \\ \frac{k_{51}}{T_{corr}} & 0 \\ \frac{k_{61}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kv \#n}^{3 \times 2} = \begin{bmatrix} \frac{k_{4,N}}{T_{corr}} & 0 \\ \frac{k_{5,N}}{T_{corr}} & 0 \\ \frac{k_{6,N}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-181})$$

$$\mathbf{A}_{ka \#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{71}}{T_{corr}} & 0 \\ \frac{k_{81}}{T_{corr}} & 0 \\ \frac{k_{91}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{ka \#n}^{3 \times 2} = \begin{bmatrix} \frac{k_{7,N}}{T_{corr}} & 0 \\ \frac{k_{8,N}}{T_{corr}} & 0 \\ \frac{k_{9,N}}{T_{corr}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{kt \#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,1}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kt \#n}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,N}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kd \#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,1}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{kd \#n}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,N}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-182})$$

$$\mathbf{A}_a^{3 \times 3} = \begin{bmatrix} -1 & 0 & 0 \\ \tau_{a,x} & & \\ 0 & -1 & 0 \\ & \tau_{a,y} & \\ 0 & 0 & -1 \\ & & \tau_{a,z} \end{bmatrix} \quad (\text{III-183})$$

$$\mathbf{B}_{11}^{2N \times N} = \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{2}{f_{CA} \cdot T_{corr}} & 0 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (\text{III-184})$$

The derived centralized state space representation is a MIMO system.

In the following, the SISO transfer functions for one channel are derived.

In case of position state formulation, the NCO command is derived from the states of the vector tracking filter, using a nonlinear transformation. Besides the already linearized centralized state space matrix, also the observation function must be linearized, in order to get a linear observation vector.

$$\begin{aligned}\Delta \hat{\chi}_{\#n} &= h_{Sv}(\hat{\mathbf{z}}, \mathbf{u}, \mathbf{K}, \mathbf{x}_{sv\#1}, \dots, \mathbf{x}_{sv\#n}) + \left. \frac{\partial h_{Sv}}{\partial \hat{\mathbf{z}}} \right|_{WP} \cdot \delta \mathbf{z} \\ \delta \Delta \hat{\chi}_{\#n} &= \Delta \hat{\chi}_{\#n} - h_{Sv}(WP) = \left. \frac{\partial h_{Sv}}{\partial \hat{\mathbf{z}}} \right|_{WP} \cdot \delta \mathbf{z} = \mathbf{c}^T \cdot \delta \mathbf{z} \\ \mathbf{c}_{Sv}^T &= \begin{bmatrix} \mathbf{0}^{1 \times 2N} & \mathbf{c}_{h\#Sv}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 0 \end{bmatrix} \\ \mathbf{c}_{h\#Sv}^{1 \times 3} &= \frac{-1}{\lambda_{C/A}} \cdot \frac{(\mathbf{x}_{\#Sv} - \hat{\mathbf{x}}_e)^T}{\|\mathbf{x}_{\#Sv} - \hat{\mathbf{x}}_e\|}\end{aligned}\tag{III-185}$$

The corresponding transfer function is again calculated, using

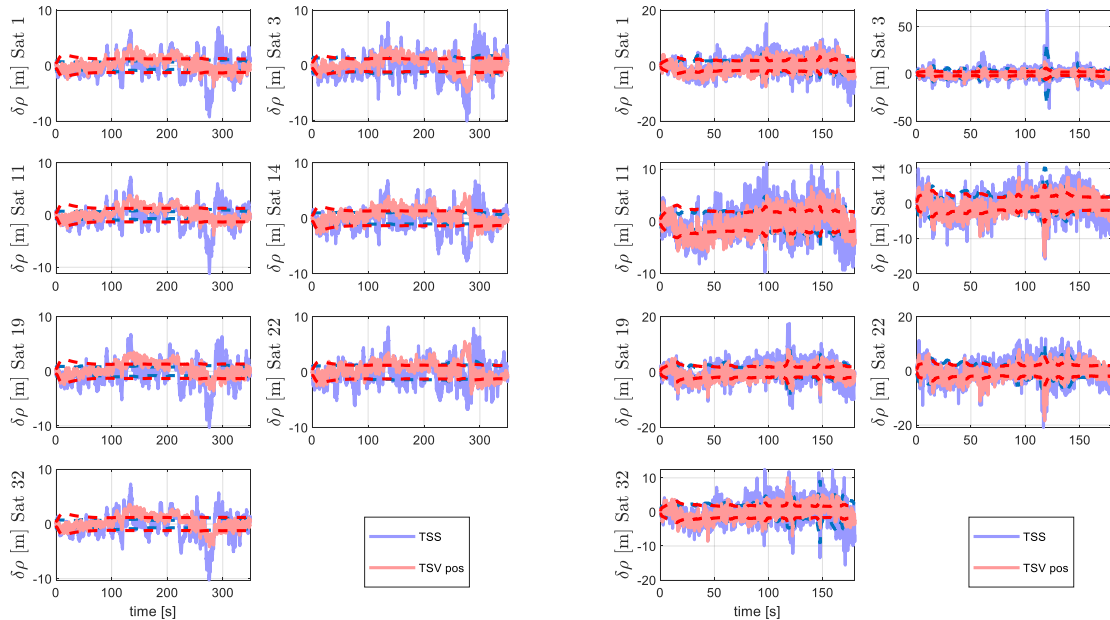
$$\mathbf{g}_{\#Sv}(s) = \mathbf{c}_{\#Sv}^T \cdot (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}_{\#Sv}\tag{III-186}$$

III - 7.5 Scalar total state tracking vs. Vector total state tracking

In this section, the previous architecture of unaided total state scalar tracking will be compared to total state unaided vector tracking.

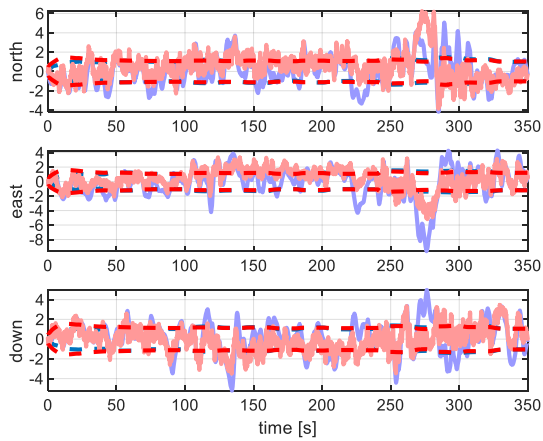
The following simulations show the DA42 scenario and the high dynamic scenario.

UNJAMMED

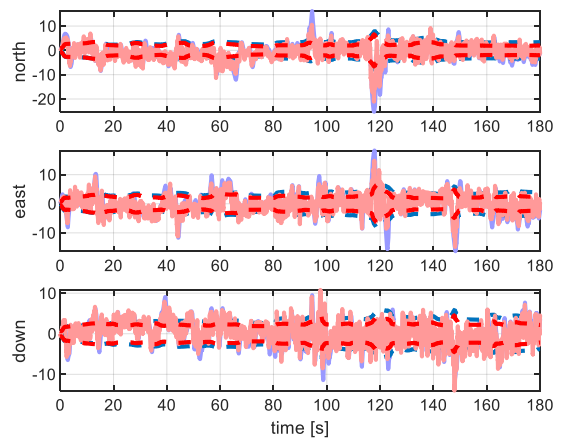


III-110 pseudorange error - DA42 - unjammed

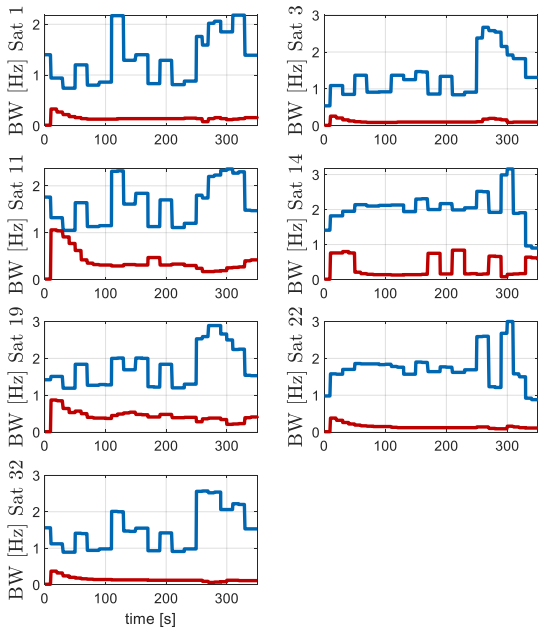
III-111 pseudorange error - high dynamic - unjammed



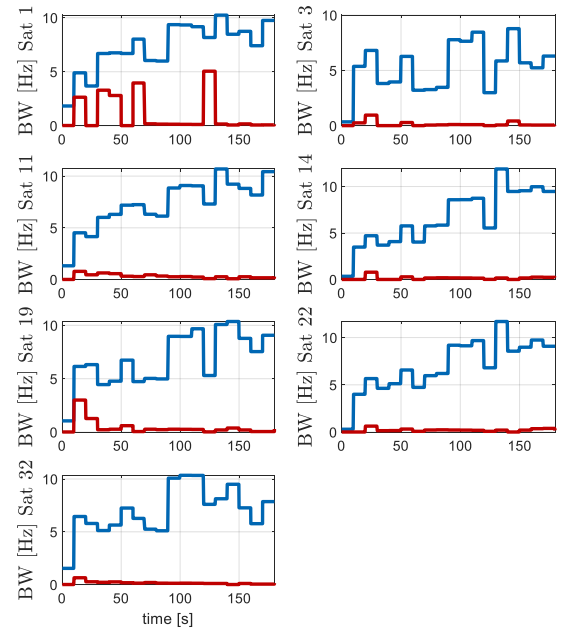
III-112 positioning error - DA42 - unjammed



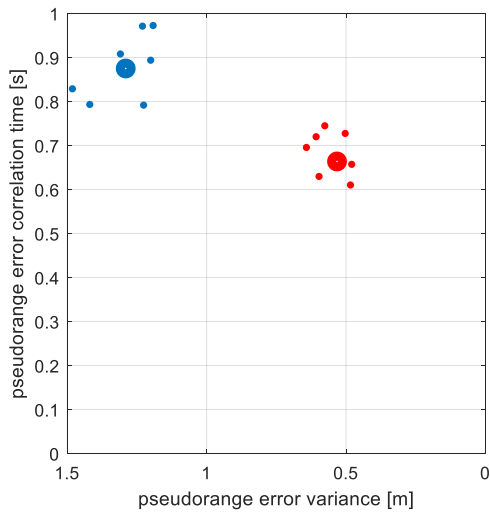
III-113 positioning error - high dynamic - unjammed



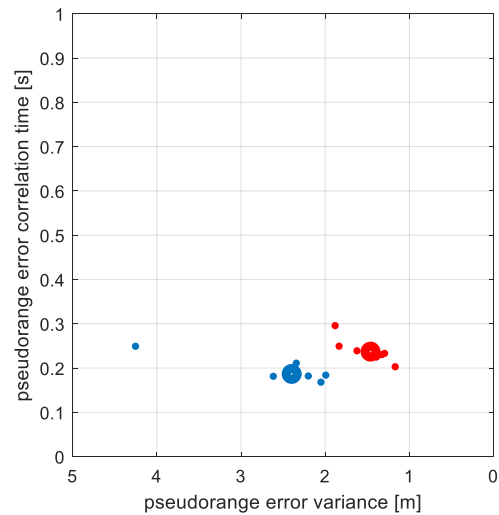
III-114 bandwidth - DA42 - unjammed



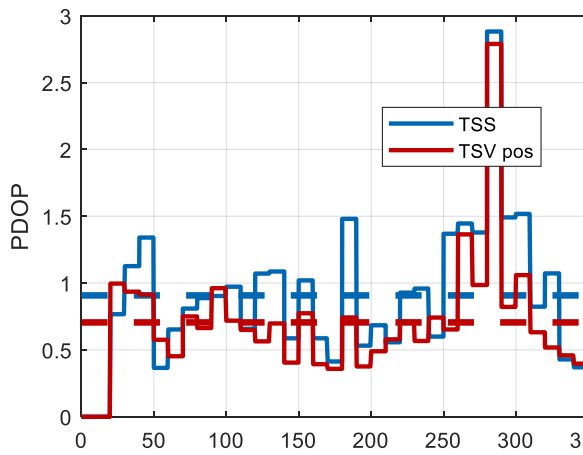
III-115 bandwidth - high dynamic - unjammed



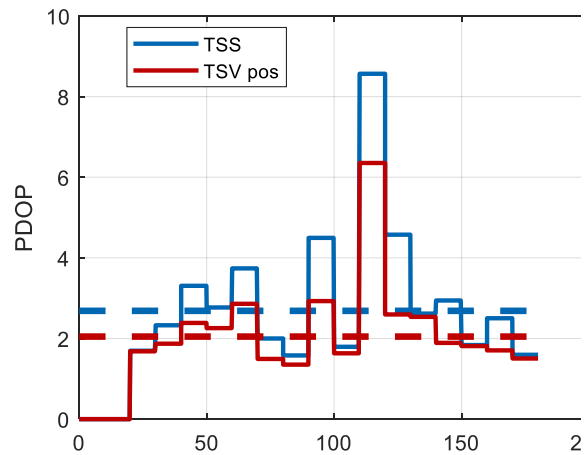
III-116 Comparing raw date - DA42 - unjammed



III-117 Comparing raw date - high dynamic - unjammed



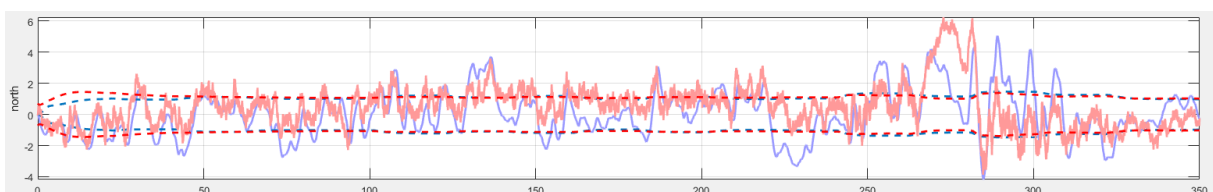
III-118 PDOP - DA42 - unjammed



III-119 PDOP - high dynamic - unjammed

In the following, a zoomed example of the positioning error in north direction for the DA42 approach is shown. The long-correlated errors of the vector tracking approach are similar to the long-correlated errors of the scalar case. Upon the long-correlated errors, the vector tracking approach shows very short correlated errors. This is due to the single filter stage.

The scalar tracking approach consists out of two filter stages. The first one, tracking the different pseudoranges and the second one for positioning solution. The vector tracking approach only has one tracking stage in contrast.



III-120 DA42 - total state vector position - zoomed

Out of these two architectures, the total state vector tracking approach shows the best tracking results.

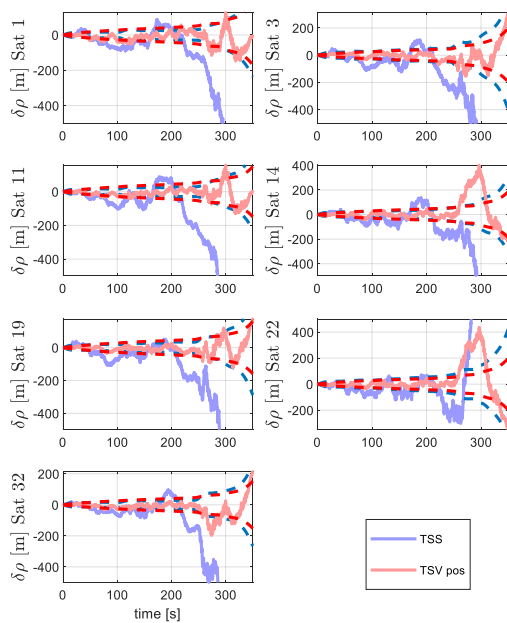
As the bandwidth analysis shows, the closed loop tracking bandwidth of the vector tracking approach is much smaller compared to the scalar tracking approach.

Nevertheless, the dynamic stress error is even smaller.

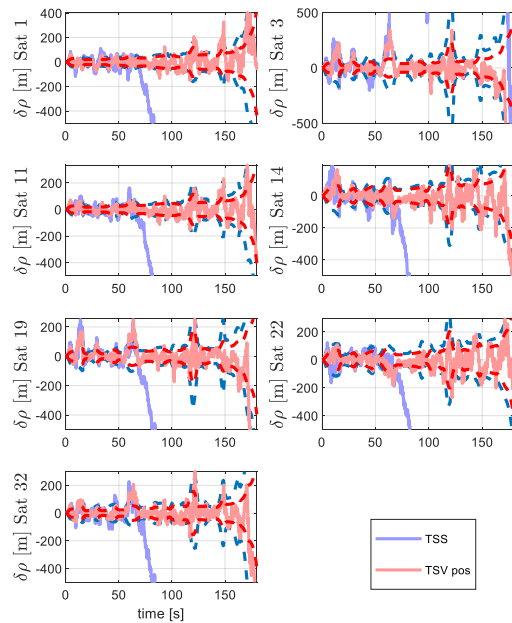
The huge advantage of the total state position state form is its low closed loop tracking bandwidth, making this architecture robust against jamming. Through the mutual aiding of the satellite channels, the remaining dynamic stress error becomes low, despite the low bandwidth.

The figures III-114 and III-115 on closed loop tracking bandwidth show quite well, which satellite channel benefits most from inter satellite aiding. Channel 1, 3, 22 and 32 show a very small closed loop tracking bandwidth, because the line of sight dynamic of these channels is correlated very well and therefore the line of sight dynamic of one satellite is used for aiding the other one and vice versa. Channel 11 and 19 for example show a higher closed loop tracking bandwidth. Their line of sight dynamic is less correlated with the others.

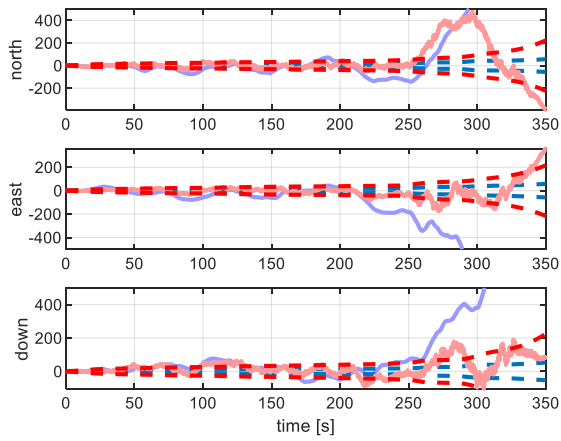
JAMMED



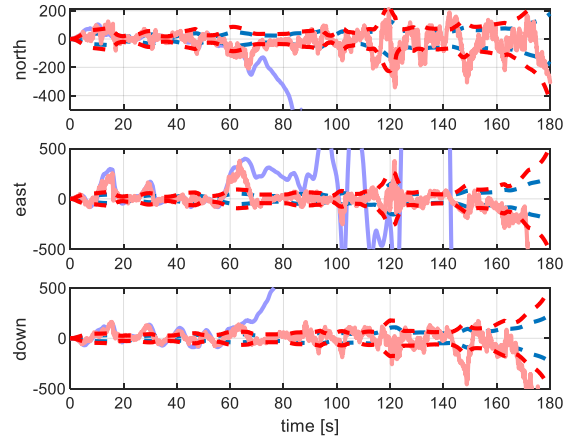
III-121 pseudorange error - DA42 - jammed



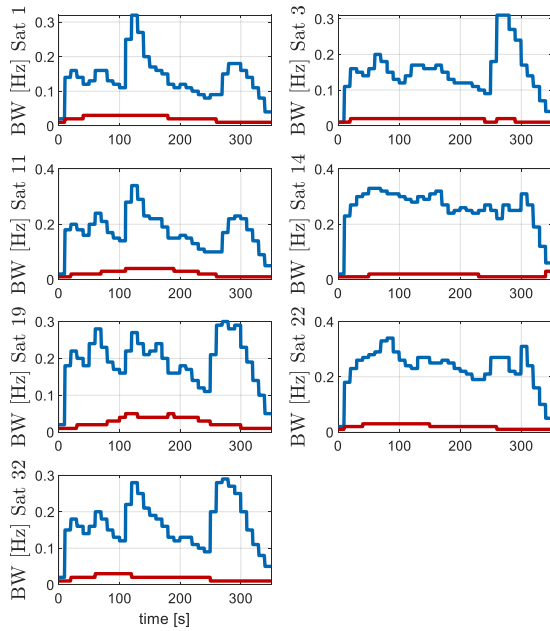
III-122 pseudorange error - high dynamic - jammed



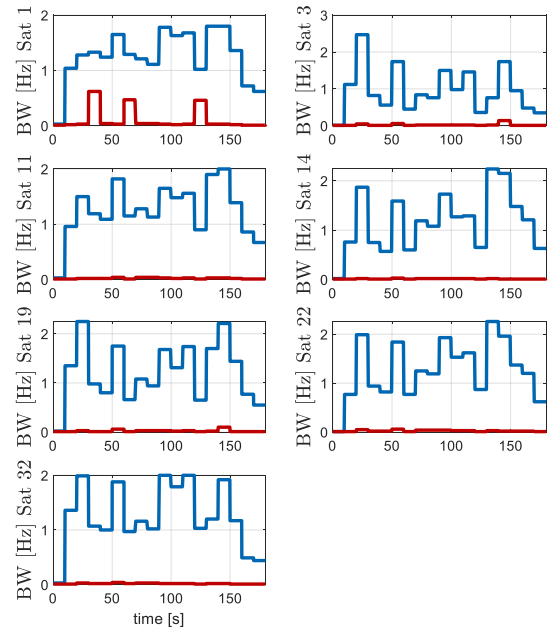
III-123 positioning error - DA42



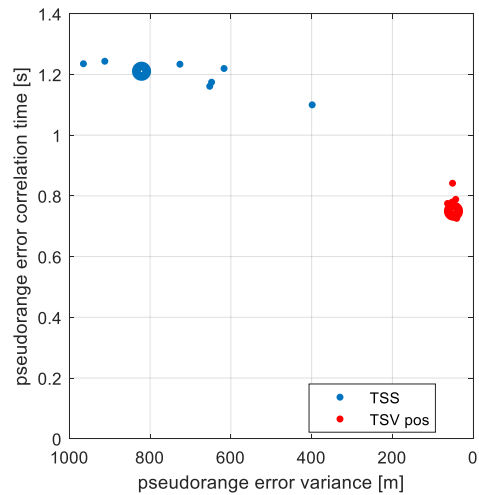
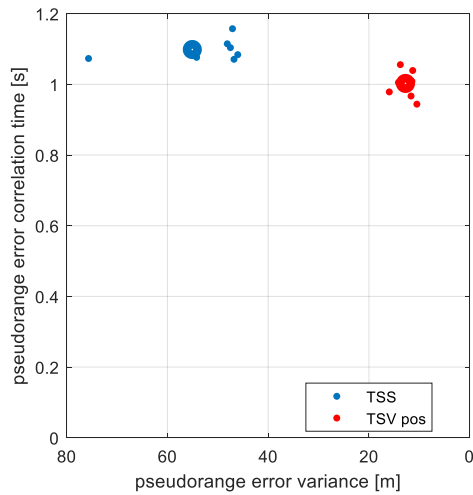
III-124 positioning error - high dynamic



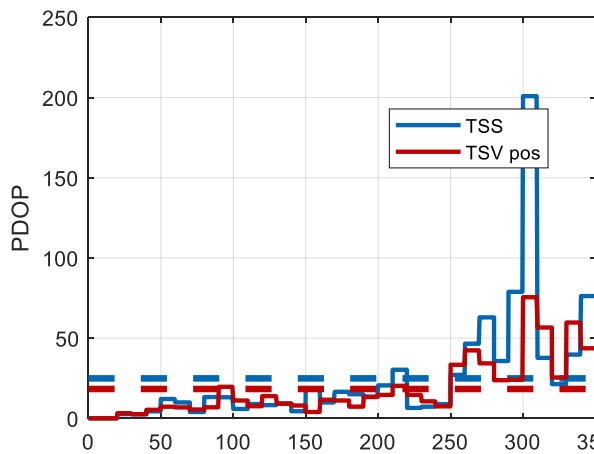
III-125 bandwidth - DA42



III-126 bandwidth - high dynamic

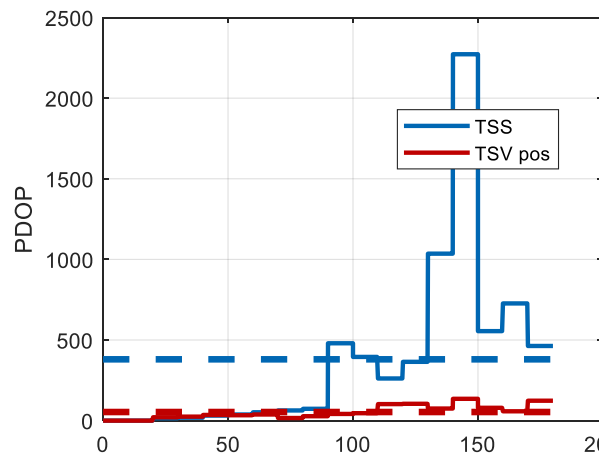


III-127 Comparing raw date - DA42



III-129 PDOP - DA42

III-128 Comparing raw date - high dynamic



III-130 PDOP - high dynamic

The simulations above show that vector tracking, even in the unaided case, provides better tracking results and interference benefits, which is in accordance with [66].

Major Pros and Cons of Total State Vector Tracking

Total State Vector Tracking - unaided

- + Small closed loop tracking bandwidth for most satellite channels due to inter satellite aiding.
- + Immediate reacquisition after tracking outage of one satellite
- + Due to low closed loop tracking bandwidth high robustness against jamming
- + The simulation of jammed scenarios shows the superiority of total state vector tracking. Although the vector tracking approach loses lock when jamming power becomes large, but can resist much longer than the total state scalar approach.
 Especially in case of jamming and high dynamic scenarios, the vector tracking approach can withstand much longer at an increasing jamming power.

III - 8 Error state vector tracking - unaided

III - 8.1 Motivation

In the previous section, vector tracking in a total state formulation was analyzed. The whole platform dynamic must be handled within the total state dynamic filter.

This section develops a tracking architecture, where the main tracking filter has an error state formulation. Some very general error state vector tracking architectures are given in [67], [56], [68] and [69] but there, only the error state dynamic model are given without the needed additional filter equations and tunings.

The main contributions of this section are:

Contribution 1: Development of an unaided error state vector tracking approach. The architecture in this section stands out from the given architectures by improved coupling between the error state tracking filter and the total state dynamic propagation. Additionally, detailed tuning measures are given.

Contribution 2: Evaluation if an error state vector tracking approach provides benefits, even if there is no aiding.

III - 8.2 Tracking architecture

In the previous section, the dynamic model used within the vector tracking has to consider the whole platform dynamic respectively whole line of sight dynamic in case of pseudorange state formulation. The acceleration was modeled as a Gauss Markov process with zero mean.

For the error state approach, there is an external model necessary, propagating the estimated platform dynamic. As total state line of sight dynamic, a constant acceleration is assumed.

$$\dot{\hat{\mathbf{z}}} = \begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e^e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ c \cdot \dot{\hat{t}}_{clk} \\ c \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \mathbf{A}_z \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 1 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} \quad (\text{III-187})$$

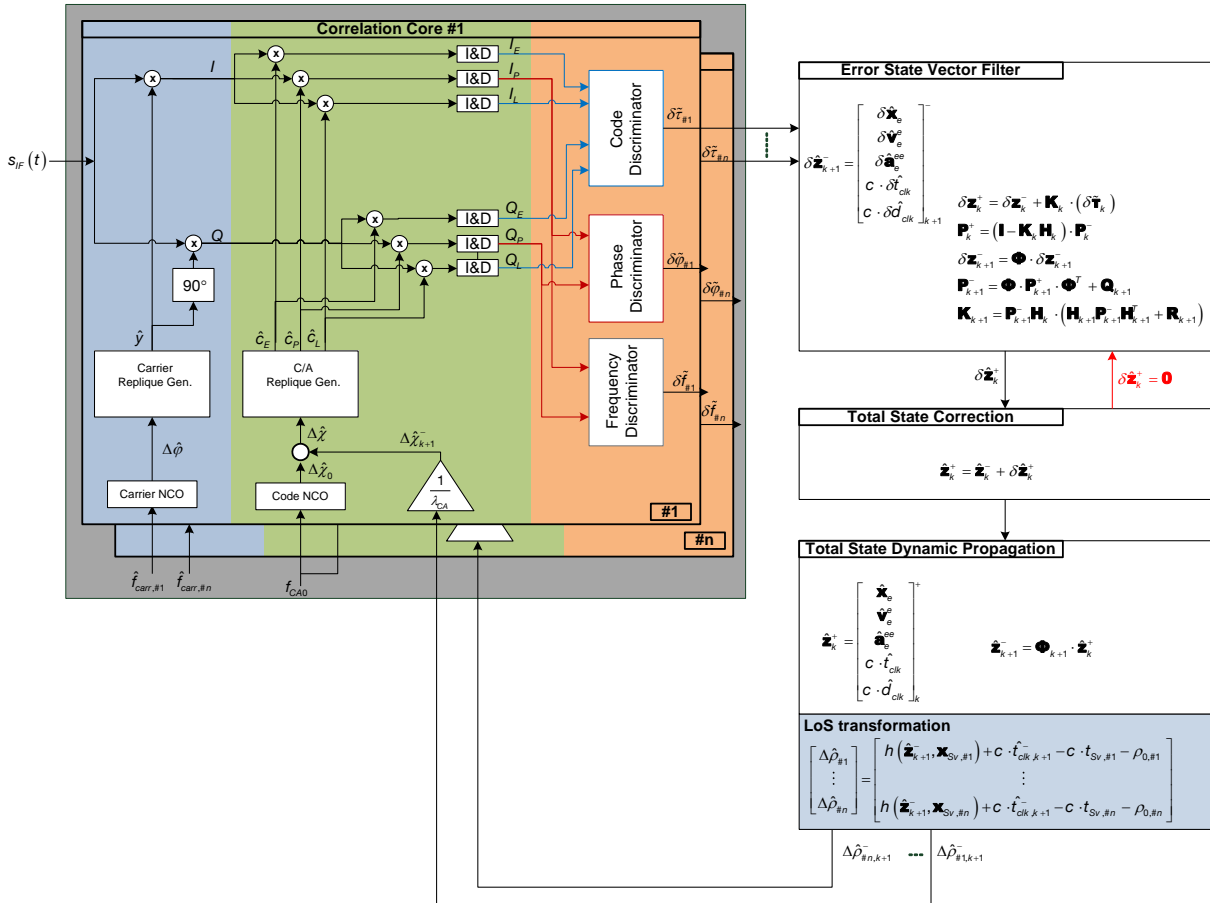
The dynamic model within the error state Kalman filter shall only consider the residual errors of the external platform dynamic. For that, the acceleration error is again modeled as a zero mean Gauss Markov process.

$$\delta \dot{\mathbf{z}} = \begin{bmatrix} \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_e^e \\ \delta \dot{\mathbf{a}}_e^{ee} \\ c \cdot \delta \dot{t}_{clk} \\ c \cdot \delta \dot{d}_{clk} \end{bmatrix} = \mathbf{A}_{\delta \mathbf{z}} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_e^e \\ \delta \mathbf{a}_e^{ee} \\ c \cdot \delta t_{clk} \\ c \cdot \delta d_{clk} \end{bmatrix} + \mathbf{n} = \begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_a & 0 & 0 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 1 \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_e^e \\ \delta \mathbf{a}_e^{ee} \\ c \cdot \delta t_{clk} \\ c \cdot \delta d_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{n}_{\delta a} \\ n_{\delta t, clk} \\ n_{\delta d, clk} \end{bmatrix}$$

(III-188)

$$\mathbf{A}_a = \begin{bmatrix} -1 & 0 & 0 \\ \tau_{ax} & -1 & 0 \\ 0 & \tau_{ay} & -1 \\ 0 & 0 & \tau_{az} \end{bmatrix}$$

The following figure shows the corresponding tracking architecture.



III-131 Error state vector tracking - unaided

The estimated tracking errors are used to correct the total states, which are propagated with their own dynamic model. After correction, the error states are reset.

For tuning, again scenario matched tuning is used. The acceleration dynamic is modeled as a Gauss Markov process with zero mean.

The propagated total states are used to calculate the NCO commands.

$$\hat{\rho}_{\#n,k+1}^- = h(\hat{\mathbf{z}}_{k+1}^-, \mathbf{x}_{Sv,\#n}) = \sqrt{(\mathbf{x}_{Sv,\#n} - \hat{\mathbf{x}}_{e,k+1}^-)^T \cdot (\mathbf{x}_{Sv,\#n} - \hat{\mathbf{x}}_{e,k+1}^-) + c \cdot \hat{t}_{clk,k+1}^- - c \cdot t_{Sv,\#n}} \quad (\text{III-189})$$

This absolute pseudorange $\hat{\rho}_{\#n,k+1}^-$ is converted into a relative pseudorange progress $\Delta\hat{\rho}_{\#n,k+1}^-$ since the beginning of tracking at time t_0 . This initial pseudorange is determined by the acquisition algorithm.

$$\Delta\hat{\rho}_{\#n,k+1}^- = \hat{\rho}_{\#n,k+1}^- - \hat{\rho}_{\#n}^-(t_0) \quad (\text{III-190})$$

The innovation is equally to previous architectures, calculated within the correlation core. At first, the discriminator outputs get

$$\delta\tilde{\mathbf{T}} = T_{C/A} \cdot \left(\Delta\tilde{\mathbf{x}}_{k+1} - \frac{1}{\lambda_{C/A}} \cdot \Delta\hat{\rho}_{k+1}^- \right) \quad (\text{III-191})$$

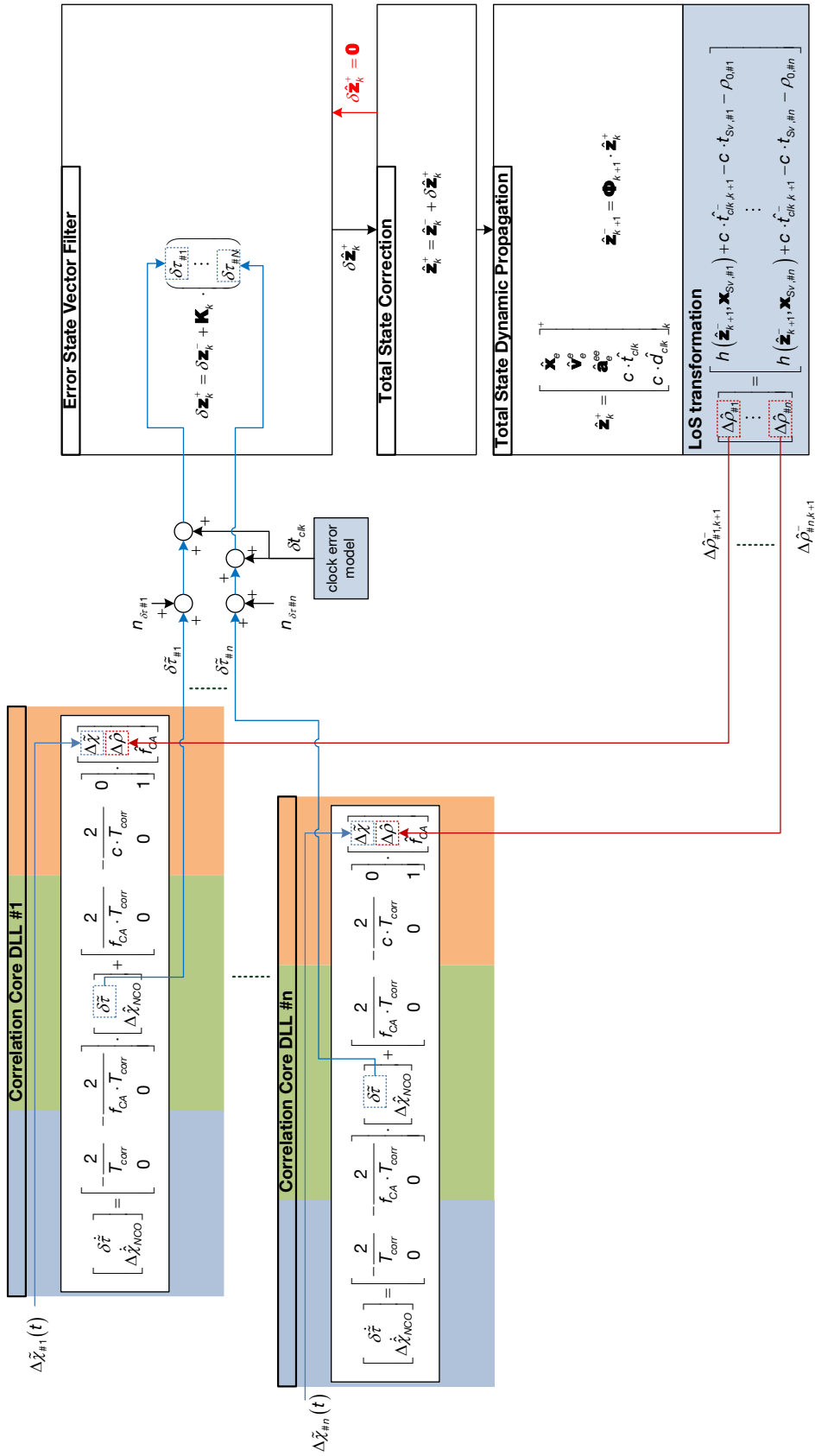
The model error covariance matrix \mathbf{Q}_k is given as

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{33} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & c^2 \cdot \sigma_{\delta t, clk}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 \cdot \sigma_{\delta f, clk}^2 \end{bmatrix}, \quad \mathbf{Q}_{33} = \begin{bmatrix} \frac{2 \cdot \sigma_{ax}^2}{\tau_{ax}} & 0 & 0 \\ 0 & \frac{2 \cdot \sigma_{ay}^2}{\tau_{ay}} & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{az}^2}{\tau_{az}} \end{bmatrix} \quad (\text{III-192})$$

III - 8.3 Equivalent base band dynamic state space simulation

III - 8.3.1 Distributed state space model

For equivalent base band simulation, also the error state vector tracking loop can be represented in a distributed state space realization. The following figure shows the corresponding architecture.



III-132 ESV VDLL unaided – distributed state space representation

III - 8.3.2 Centralized state space representation

For closed loop tracking bandwidth calculation, a centralized state space representation of the error state vector tracking loop will be developed in this section.

Within the centralized state space matrix, besides the error states, also the external total state propagation must be included.

In the following, the state vector of the centralized state space representation is given.

$$\hat{\mathbf{z}}^T = \left[\delta\tilde{\tau}_{\#1} \quad \Delta\hat{X}_{NCO,\#1} \quad \cdots \quad \delta\tilde{\tau}_{\#N} \quad \Delta\hat{X}_{NCO,\#N} \quad \delta\hat{\mathbf{x}}_e \quad \delta\hat{\mathbf{v}}_e^e \quad \delta\hat{\mathbf{a}}_e^{ee} \quad \mathbf{c} \cdot \delta\hat{\mathbf{t}}_{clk} \quad \mathbf{c} \cdot \delta\hat{\mathbf{d}}_{clk} \quad \hat{\mathbf{x}}_e \quad \hat{\mathbf{v}}_e^e \quad \hat{\mathbf{a}}_e^{ee} \quad \mathbf{c} \cdot \hat{\mathbf{t}}_{clk} \quad \mathbf{c} \cdot \hat{\mathbf{d}}_{clk} \right] \quad (\text{III-193})$$

The necessary NCO command, respectively feedback, is a nonlinear function of the estimated positioning solution.

Generally, the centralized state space form in nonlinear form gets

$$\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}, \mathbf{u}) \quad (\text{III-194})$$

Most equations are linear. Only the equations incorporating the feedback are nonlinear, as given below.

$$\begin{aligned} \delta\dot{\tilde{\tau}}_{\#1} &= -\frac{2}{T_{corr}} \cdot \delta\tilde{\tau}_{\#1} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\hat{X}_{NCO,\#1} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\tilde{\chi}_{\#1} - \frac{2}{\mathbf{c} \cdot T_{corr}} \cdot \left(h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}) + \mathbf{c} \cdot \hat{\mathbf{t}}_{clk} \right) \\ \Delta\dot{\hat{X}}_{NCO,\#1} &= 0 \\ &\vdots \\ \delta\dot{\tilde{\tau}}_{\#N} &= -\frac{2}{T_{corr}} \cdot \delta\tilde{\tau}_{\#N} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\hat{X}_{NCO,\#N} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\tilde{\chi}_{\#N} - \frac{2}{\mathbf{c} \cdot T_{corr}} \cdot \left(h(\mathbf{x}_{Sv\#N}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#N,0}, \hat{\mathbf{x}}_{e,0}) + \mathbf{c} \cdot \hat{\mathbf{t}}_{clk} \right) \\ \Delta\dot{\hat{X}}_{NCO,\#n} &= 0 \end{aligned} \quad (\text{III-195})$$

For bandwidth calculation and observability analysis, a linear state space system is necessary. For that, the differential equation system needs to be linearized at every sample point \mathbf{t}_k about the actual values at that time.

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{F}(\hat{\mathbf{z}}_k, \mathbf{u}_k) + \left. \frac{\partial \mathbf{F}(\mathbf{z}, \mathbf{u})}{\partial \mathbf{z}} \right|_{\hat{\mathbf{z}}_k, \mathbf{u}_k} \cdot \delta\mathbf{z} + \left. \frac{\partial \mathbf{F}(\mathbf{z}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\hat{\mathbf{z}}_k, \mathbf{u}_k} \cdot \delta\mathbf{u} \\ \dot{\mathbf{z}} - \mathbf{F}(\hat{\mathbf{z}}_k, \mathbf{u}_k) &= \delta\dot{\mathbf{z}} = \mathbf{A} \cdot \delta\mathbf{z} + \mathbf{B} \cdot \delta\mathbf{u} \end{aligned} \quad (\text{III-196})$$

In modeling the differential equations, two special features of the error state tracking architecture must be considered.

The first one is the correction of the total states

$$\hat{\mathbf{z}}_k^+ = \hat{\mathbf{z}}_k^- + \delta\hat{\mathbf{z}}_k^+ \quad (\text{III-197})$$

Writing equation (III-197) in continuous form, it gets

$$\dot{\hat{\mathbf{z}}} = \frac{1}{T_{corr}} \cdot \delta \hat{\mathbf{z}} \quad (\text{III-198})$$

The other feature is the state reset of the error states, after correction of the total states.

$$\delta \hat{\mathbf{z}}_k = \mathbf{0} \quad (\text{III-199})$$

In continuous form this gets

$$\delta \dot{\hat{\mathbf{z}}} = \frac{-1}{T_{corr}} \cdot \delta \hat{\mathbf{z}} \quad (\text{III-200})$$

The linearized steady state space differential equation system is given in the next equation. Due to the linearization, all states are error states. The states, which already have been error states like $\delta \hat{\mathbf{x}}_e$ would have to be written now with a second δ . For simplicity, this additional δ is neglected.

$$\mathbf{A}_{\delta\tau\#1}^{2 \times N} = \dots = \mathbf{A}_{\delta\tau\#N}^{2 \times N} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \quad (\text{III-202})$$

$$\mathbf{A}_{h\#1}^{2 \times 3} = \begin{bmatrix} \frac{\partial (h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}))}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \dots$$

$$= \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{\partial \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)}}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{h\#N}^{2 \times 3} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#N,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#N,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{k\delta x\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{11}}{T_{corr}} & 0 \\ \frac{k_{21}}{T_{corr}} & 0 \\ \frac{k_{31}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta x\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{1,N}}{T_{corr}} & 0 \\ \frac{k_{2,N}}{T_{corr}} & 0 \\ \frac{k_{3,N}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta v\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{41}}{T_{corr}} & 0 \\ \frac{k_{51}}{T_{corr}} & 0 \\ \frac{k_{61}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta v\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{4,N}}{T_{corr}} & 0 \\ \frac{k_{5,N}}{T_{corr}} & 0 \\ \frac{k_{6,N}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-204})$$

$$\mathbf{A}_{k\delta a\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{71}}{T_{corr}} & 0 \\ \frac{k_{81}}{T_{corr}} & 0 \\ \frac{k_{91}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta a\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{7,N}}{T_{corr}} & 0 \\ \frac{k_{8,N}}{T_{corr}} & 0 \\ \frac{k_{9,N}}{T_{corr}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{k\delta t\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,1}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta t\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,N}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta d\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,1}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta d\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,N}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-205})$$

$$\mathbf{A}_a^{3 \times 3} = \begin{bmatrix} \frac{-1}{\tau_{a,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{a,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{a,z}} \end{bmatrix} \quad (\text{III-206})$$

$$\mathbf{B}_{11}^{2N \times N} = \begin{bmatrix} \mathbf{B}_{sub}^{2 \times 2} & \dots & \mathbf{0}^{2 \times 2} \\ \vdots & \ddots & \vdots \\ \mathbf{0}^{2 \times 2} & \dots & \mathbf{B}_{sub}^{2 \times 2} \end{bmatrix}, \quad \mathbf{B}_{sub}^{2 \times 2} = \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{III-207})$$

The wanted SISO closed loop tracking bandwidth can be calculated according to next equation.

$$\Delta \tilde{\chi}_{\#1} \rightarrow \frac{1}{\lambda_{C/A}} \cdot \Delta \hat{\rho}_{\#1} \Leftrightarrow \Delta \tilde{\chi}_{\#1} \rightarrow \Delta \hat{\chi}_{\#1}: \quad g_{\#1,\#1}(s) = \mathbf{c}_{\#1}^T \cdot (s \cdot \mathbf{I} - \mathbf{A}_{CLSS})^{-1} \cdot \mathbf{b}_{\#1} \quad (\text{III-208})$$

The transformation $\mathbf{c}_{\#1}^T$ is a nonlinear function of the filter state. In order to calculate the transfer function, a linearization of this observation is necessary.

$$\begin{aligned} \Delta \hat{\chi}_{\#n} &= h(\hat{\mathbf{z}}) = h(\hat{\mathbf{z}}_k) + \left. \frac{\partial h(\hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} \right|_{\hat{\mathbf{z}}_k} \cdot \delta \mathbf{z} \\ \delta \Delta \hat{\chi}_{\#n} &= \Delta \hat{\chi}_{\#n} - h(\hat{\mathbf{z}}_k) = \left. \frac{\partial h(\hat{\mathbf{z}})}{\partial \hat{\mathbf{z}}} \right|_{\hat{\mathbf{z}}_k} \cdot \delta \mathbf{z} = \mathbf{c}_k^T \cdot \delta \mathbf{z} \\ \mathbf{c}_k^T &= [\mathbf{0}^{1 \times 2N} \quad \mathbf{0}^{1 \times 11} \quad \mathbf{c}_{N+1,k}^{1 \times 3} \quad \mathbf{0}^{1 \times 6} \quad 0 \quad 0] \\ \mathbf{c}_{N+1,k}^{1 \times 3} &= -\frac{1}{\lambda_{C/A}} \cdot \frac{(\mathbf{x}_{\#Sv,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{\#Sv,k} - \hat{\mathbf{x}}_{e,k}\|} \end{aligned} \quad (\text{III-209})$$

The input vector can be written as

$$\mathbf{b}_{\#1} = \begin{bmatrix} \mathbf{b}_{sub\#1}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{22 \times 1} \end{bmatrix}, \quad \mathbf{b}_{sub}^{2 \times 1} = \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 \end{bmatrix} \quad (\text{III-210})$$

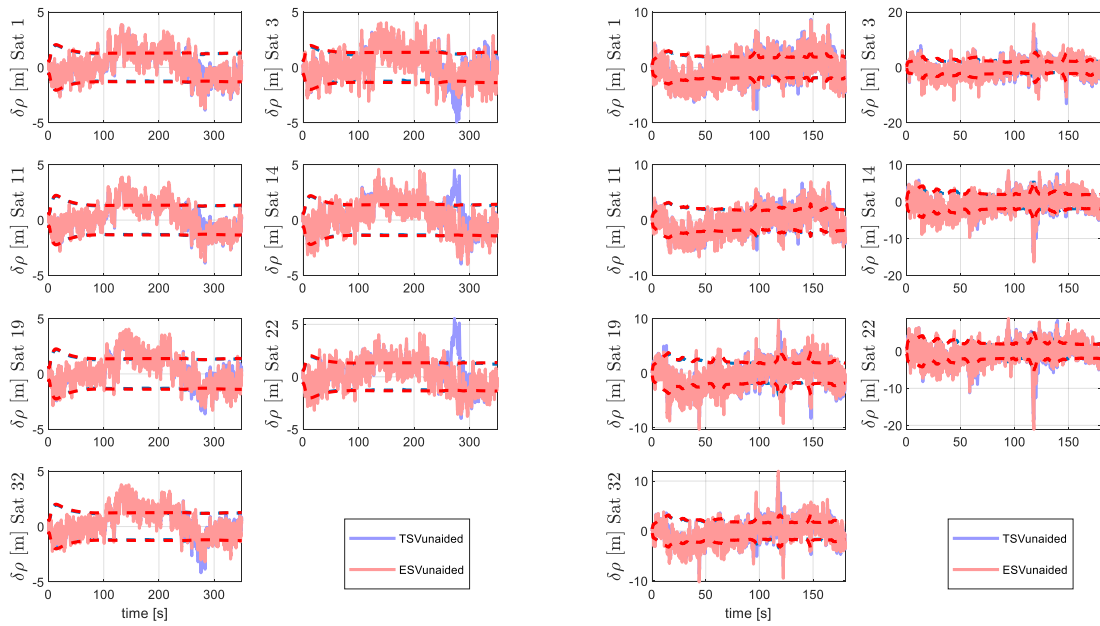
The input vector in equation (III-210) is for the SISO transfer function of channel 1, which corresponds in this example with satellite #1.

The Kalman gains used within equations (III-204) and (III-205) are time dependent. For calculating these Kalman gains, at first the distributed state space simulation done, where the settling Kalman gains are stored.

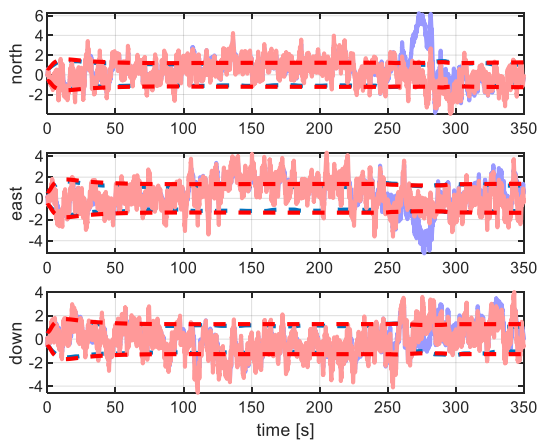
III - 8.4 Vector error state vs. vector total state (unaided)

In this section, the total state unaided vector tracking architecture will be compared to the error state unaided vector tracking architecture.

UNJAMMED

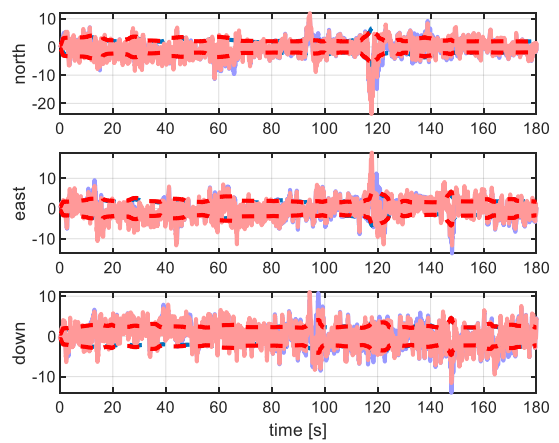


III-133 pseudorange error - DA42 - unjammed

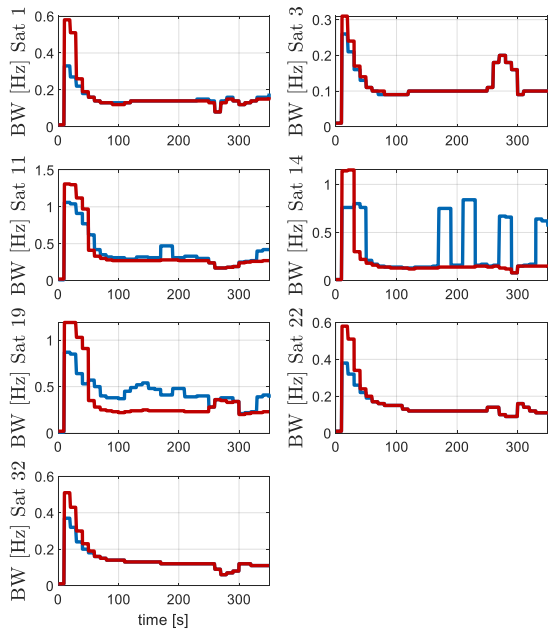


III-135 positioning error - DA42 - unjammed

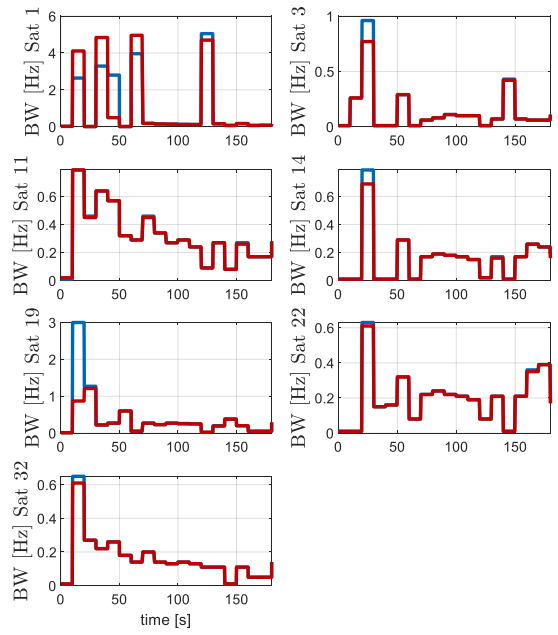
III-134 pseudorange error - high dynamic - unjammed



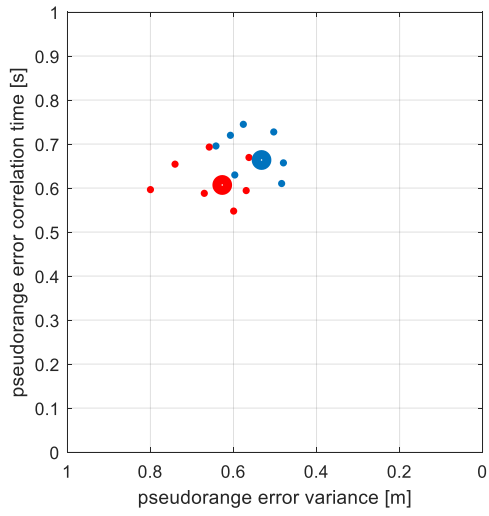
III-136 positioning error - high dynamic - unjammed



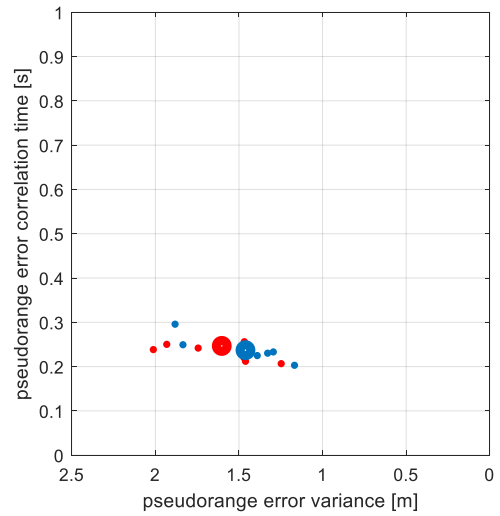
III-137 bandwidth - DA42 - unjammed



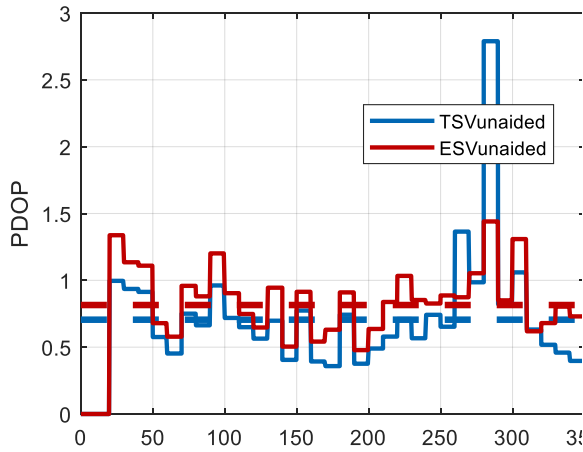
III-138 bandwidth - high dynamic - unjammed



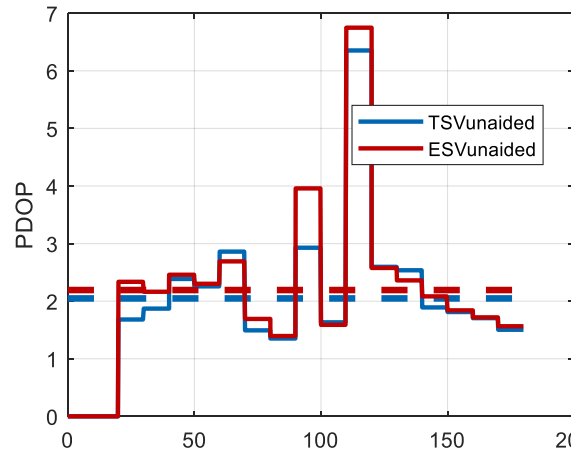
III-139 Comparing raw date - DA42 - unjammed



III-140 Comparing raw date - high dynamic - unjammed

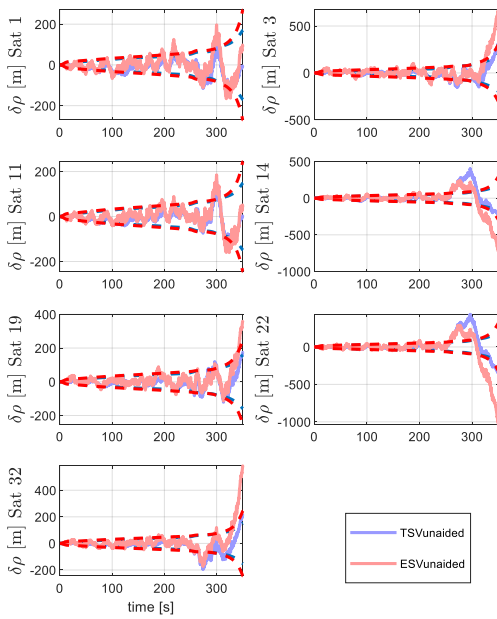


III-141 PDOP - DA42 - unjammed

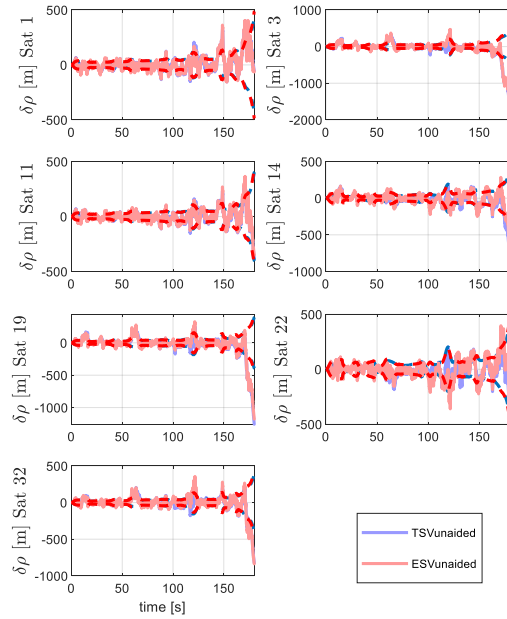


III-142 PDOP - high dynamic - unjammed

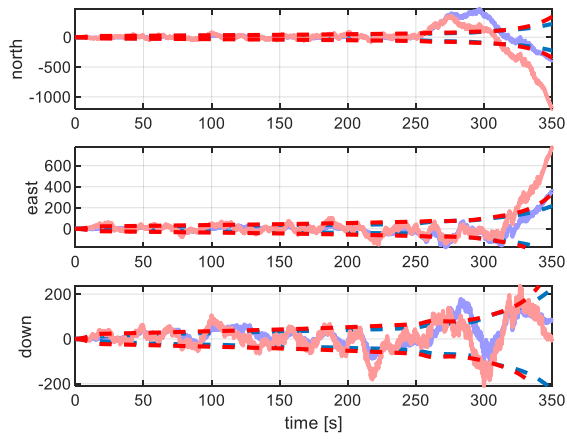
JAMMED



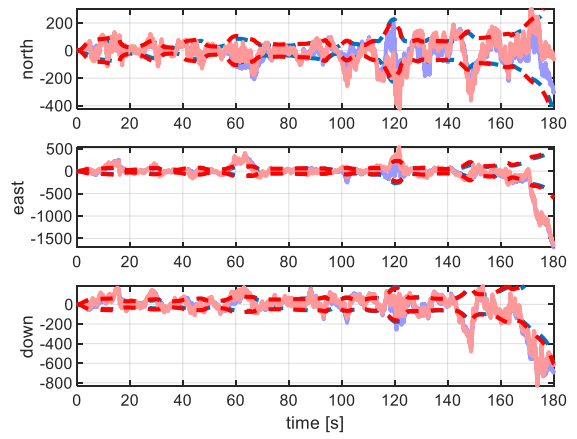
III-143 pseudorange error - DA42 - jammed



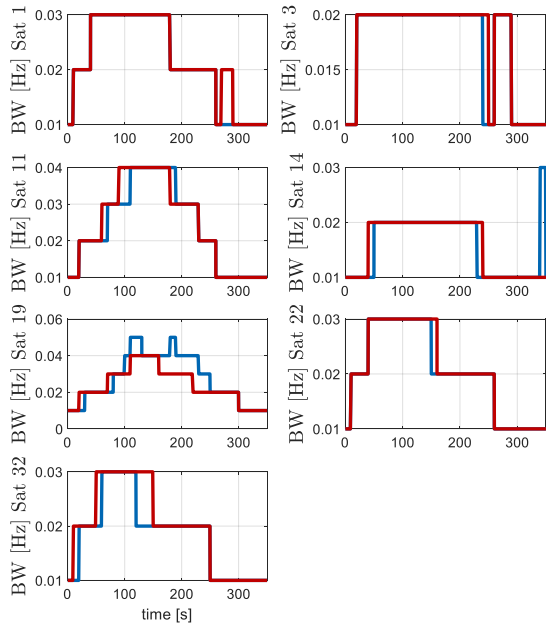
III-144 pseudorange error - high dynamic - jammed



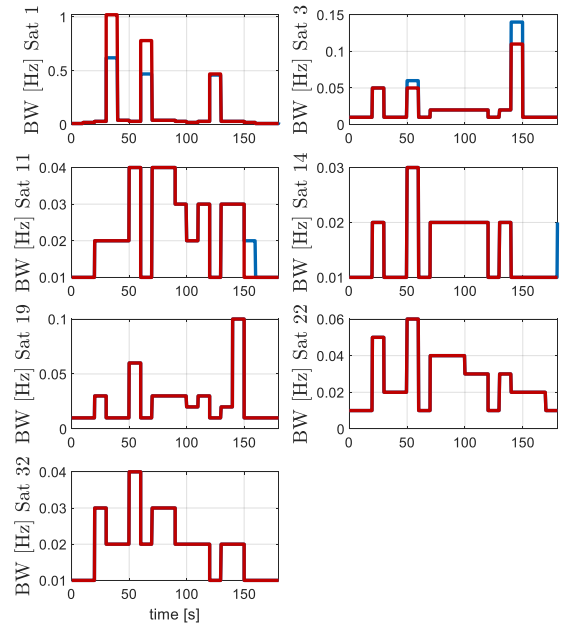
III-145 positioning error - DA42



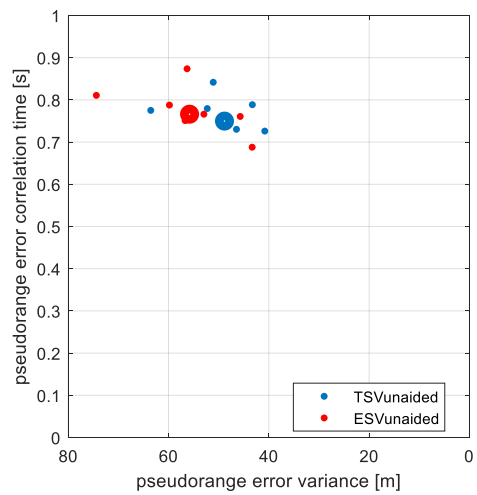
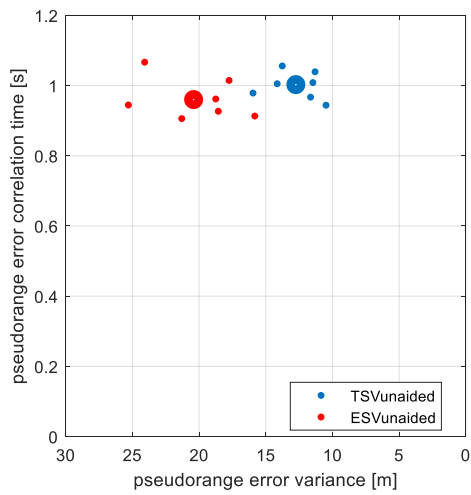
III-146 positioning error - high dynamic



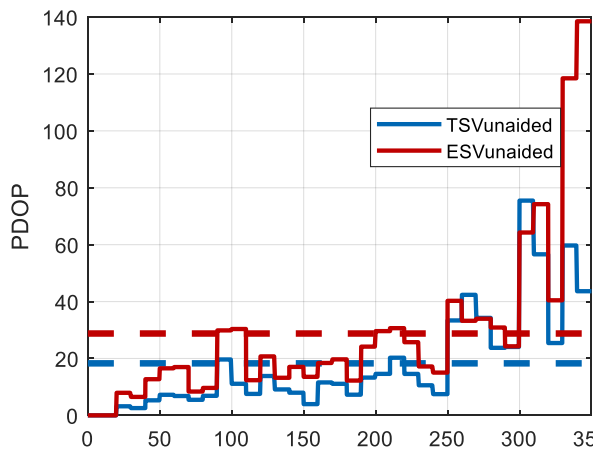
III-147 bandwidth - DA42



III-148 bandwidth - high dynamic

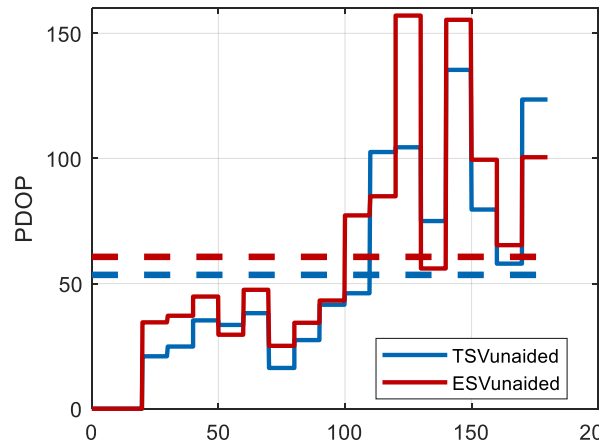


III-149 Comparing raw date - DA42



III-151 PDOP - DA42

III-150 Comparing raw date - high dynamic



III-152 PDOP - high dynamic

CONCLUSION: Error State Vector unaided vs. Total State Vector unaided

The comparison of the error state vector tracking architecture against the total state vector tracking architecture gives similar results as the comparison of the scalar error state tracking architecture with the scalar total state tracking architecture.

In case of no aiding, the error state approach does not provide any huge advantages. Without the pre knowledge from aiding, the error state tracking filter still must follow the full line of sight dynamic.

Pros		Cons	
+	Partly better usage of inter satellite aiding and less dynamic stress, mostly in low dynamic trajectories.	-	Higher complexity and higher processing load do not provide much performance improvement
+	If the raw data are used in a tightly coupled navigation system, the error correlation times are shorter compared to the total state approach. This fact makes it easier for error consideration within the integration filter.	-	Higher raw data error variance
+	Less dynamic stress errors in low dynamic scenarios.		

III - 9 Aided tracking – aiding error modeling

III - 9.1 Motivation

In the previous sections, only unaided tracking was considered. In the following sections, aided tracking will be considered.

The aiding information is usually taken out of an integration filter, like a tightly coupled filter or other external aiding sources. Aiding comes along with different kind of error which need to be modeled, estimated or considered within the tracking architecture.

In literature, there is surprisingly little discussion about aided GPS tracking architectures. But precisely for aiding, the selected tracking architecture, the way of aiding application respectively integration into the tracking architecture and the appropriate consideration of aiding errors, is important to get good results.

This and the following sections provide therefore the following vital contributions in order to design and setup a tracking architecture for optimal aiding application.

Contribution 1: A detailed derivation of aiding errors is given and simultaneously, aiding error models are developed with regard to scalar tracking architectures and vector tracking architectures.

Contribution 2: Novel tracking architectures are developed with different kinds of aiding integration and aiding error modeling.

Contribution 3: Aided tracking architectures need some special tuning, which is discussed in detail. Also some tuning pitfalls are discussed and evaluated.

Contribution 4: Depending on the type of aiding error and the type of used tracking architecture, it does sometimes make sense to estimate the aiding errors or just only consider the aiding errors. There will be an analysis and decision guideline about the optimal proceeding for different aiding error types and tracking architectures.

Contribution 5: Given some decentralized tracking filter architecture, where aiding errors are estimated in some local filters, the question will be answered, in which cases a feedback of aiding errors to superior filter improves the overall results.

Contribution 6: Aiding can be of good or bad quality. It is important to which aiding quality is at least necessary in order to improve tracking results. This thesis develops a decision matrix for different tracking architecture and dynamic ranges, which minimum aiding quality is necessary. The same decision matrix is also developed regarding the aiding rate.

III - 9.2 Aiding source

As external aiding source, velocity aiding in “NED” coordinates is used.

The aiding velocity is assumed to be the output of an inertial navigation system (INS). Typically, as aiding information for GPS receivers, is the velocity of an INS given in coordinates of the "north-east-down" frame or short n-frame \mathbf{V}_e^n . Internally, the receiver converts the velocity from

n-frame coordinates to e-frame coordinates in order to apply a line of sight transformation to get $V_{\#n,LoS}$.

$$\mathbf{v}_{e,aiding}^e = \mathbf{R}_{en}(\lambda, \phi) \cdot \mathbf{v}_{e,aiding}^n \quad (\text{III-211})$$

The transformation can be realized by using geodetic position information regarding longitude and latitude λ, ϕ from the INS through an additional aiding interface, or by using the receiver internal available position solution.

The transformation matrix \mathbf{R}_{en} is calculated by using the receiver internal position estimation $\hat{\mathbf{x}}_e$. The transformation from n-frame to e-frame is

$$\mathbf{R}_{en,k} = \begin{bmatrix} -\sin(\phi_k) \cos(\lambda_k) & -\sin(\lambda_k) & -\cos(\phi_k) \cos(\lambda_k) \\ -\sin(\phi_k) \sin(\lambda_k) & \cos(\lambda_k) & -\cos(\phi_k) \sin(\lambda_k) \\ \cos(\phi_k) & 0 & -\sin(\phi_k) \end{bmatrix} \quad (\text{III-212})$$

For calculation of Cartesian coordinates from geodetic coordinates, the following expression can be used. [34, p. 35]

$$\mathbf{x}_{e,k} = \begin{bmatrix} (R_N + h_k) \cos(\phi_k) \cos(\lambda_k) \\ (R_N + h_k) \cos(\phi_k) \sin(\lambda_k) \\ [R_N(1 - e^2) + h_k] \sin(\phi_k) \end{bmatrix} \quad (\text{III-213})$$

The reverse transformation is more difficult, because an iterative approach must be used.

In the first run, the values can be calculated according to next equations, which are given in [2, p. 42].

$$\begin{aligned} \sin(\phi_k) &= \frac{z_{e,k}}{(1 - e^2) \cdot R_N(\phi_0) + h_k} \\ \tan(\lambda_k) &= \frac{y_{e,k}}{x_{e,k}} \\ h_k &= \frac{\sqrt{x_{e,k}^2 + y_{e,k}^2}}{\cos(\phi_k)} - R_N(\phi_0) \end{aligned} \quad (\text{III-214})$$

Because the ellipsoid normal radius $R_N(\phi_k)$ being a function of the latitude, the geodetic coordinates must be calculated iteratively. This iteration must be realized within the GPS receiver, if for aiding transformation the own positioning solution is used.

The aiding velocity $\mathbf{v}_{e,aiding}^n$ contains errors of different characteristics. Additionally, the transformation $\mathbf{v}_{e,aiding}^e = \mathbf{R}_{en}(\lambda, \phi) \cdot \mathbf{v}_{e,aiding}^n$ couples actual receiver positioning errors into the

aiding velocity, if λ, ϕ are derived out of the receiver internal available position.

The following equation specifies the different error parts within the aiding velocity.

$$\tilde{\mathbf{v}}_{e,aiding}^e = \mathbf{v}_{e,aiding}^e + \delta\mathbf{v}_{e,\Delta t}^e + \delta\mathbf{v}_{e,\delta\psi}^e + \delta\mathbf{v}_{e,N}^e \quad (\text{III-215})$$

$\delta\mathbf{v}_{e,\Delta T}^e$:= velocity error caused by time delay (out of sequence aiding)

$\delta\mathbf{v}_{e,\delta\psi}^e$:= velocity error caused by tilt errors

$\delta\mathbf{v}_{e,N}^e$:= noise like errors

$\delta\mathbf{v}_{e,R}^e$:= errors caused by low aiding rate

(III-216)

In the following subsections, these errors will be considered in detail.

III - 9.3 Noise like aiding error

The aiding velocity is given in n-frame coordinates. Its components show zero mean noise like errors being correlated in time. These error components are typically approximated by a Gauss Markov process of 1st order. The Gauss Markov process is characterized by its correlation time constant and its error variance.

In this thesis the following characteristics are used.

$$\begin{aligned} \sigma_{\delta v_{e,n,N}^n} &= 0.1 \frac{m}{s}, \quad \sigma_{\delta v_{e,e,N}^n} = 0.1 \frac{m}{s}, \quad \sigma_{\delta v_{e,d,N}^n} = 0.1 \frac{m}{s} \\ \tau_{\delta v_{e,n,N}^n} &= 50s, \quad \tau_{\delta v_{e,e,N}^n} = 50s, \quad \tau_{\delta v_{e,d,N}^n} = 50s \end{aligned} \quad (\text{III-217})$$

These parameters depend on the used INS and also on the IMU, which is used for the INS. The selected parameters, being used in this work, are based on an INS system analyzed in [33, p. 253].

$$\begin{aligned} \delta\dot{\mathbf{v}}_{e,N}^n &= \mathbf{A}_{\delta v,N}^n \cdot \delta\mathbf{v}_{e,N}^n + \mathbf{n}_{\delta v,N}^n \\ \begin{bmatrix} \delta\dot{v}_{e,n,N}^n \\ \delta\dot{v}_{e,e,N}^n \\ \delta\dot{v}_{e,d,N}^n \end{bmatrix} &= \begin{bmatrix} \frac{-1}{\tau_{\delta v_{e,n}^n}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v_{e,e}^n}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v_{e,d}^n}} \end{bmatrix} \cdot \begin{bmatrix} \delta v_{e,n,N}^n \\ \delta v_{e,e,N}^n \\ \delta v_{e,d,N}^n \end{bmatrix} + \begin{bmatrix} n_1 \cdot \sqrt{\frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}}} \\ n_2 \cdot \sqrt{\frac{\sigma_{\delta v_{e,e,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,e,N}^n}}} \\ n_3 \cdot \sqrt{\frac{\sigma_{\delta v_{e,d,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,d,N}^n}}} \end{bmatrix} \end{aligned} \quad (\text{III-218})$$

It is assumed, that in n-frame, the noise like error components in north, east and down direction are uncorrelated.

$$\mathcal{E}\left\{\mathbf{n}_{\delta\mathbf{v}_{e,N}^n} \cdot \mathbf{n}_{\delta\mathbf{v}_{e,N}^n}^T\right\} = \begin{bmatrix} \sigma_{\delta v_{e,n,N}^n}^2 & 0 & 0 \\ 0 & \sigma_{\delta v_{e,e,N}^n}^2 & 0 \\ 0 & 0 & \sigma_{\delta v_{e,d,N}^n}^2 \end{bmatrix} \quad (\text{III-219})$$

Moreover, it is assumed that n_1, n_2, n_3 are uncorrelated and are white Gaussian noise processes, having a variance of 1.

The tracking is realized in e-frame. Therefore, the derived n-frame Gauss Markov process must be transformed into e-frame according to the following equations.

$$\delta\mathbf{v}_{e,N}^e = \mathbf{R}_{en}(\lambda(t), \phi(t)) \cdot \delta\mathbf{v}_{e,N}^n \quad (\text{III-220})$$

The original uncorrelated errors in n-frame become correlated in e-frame. In order to consider this correlation in the right way within the corresponding tracking filters, the differential equation system, describing the Gauss Markov process, will be transformed.

$$\delta\dot{\mathbf{v}}_{e,N}^e = \dot{\mathbf{R}}_{en} \cdot \delta\mathbf{v}_{e,N}^n + \mathbf{R}_{en} \cdot \delta\dot{\mathbf{v}}_{e,N}^n \quad (\text{III-221})$$

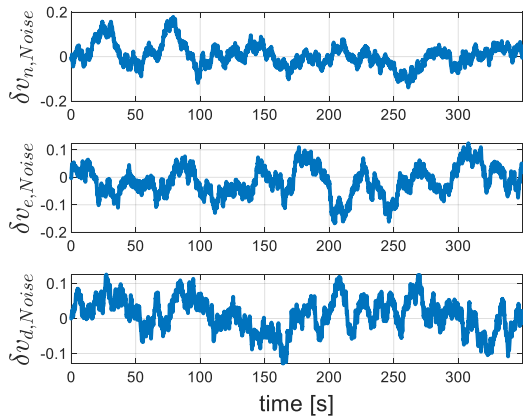
By using

$$\dot{\mathbf{R}}_{en} = \mathbf{R}_{en} \cdot \boldsymbol{\Omega}_{en}^n \quad (\text{III-222})$$

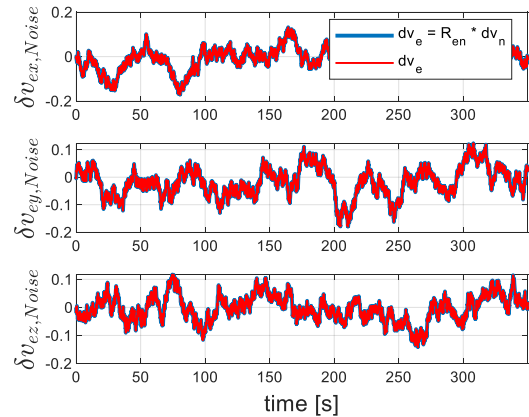
together with (III-220) and (III-221), the Gauss Markov process in e-frame coordinates gets

$$\delta\dot{\mathbf{v}}_{e,N}^e = \left(\mathbf{R}_{en} \cdot \boldsymbol{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N}^n \cdot \mathbf{R}_{en}^T\right) \cdot \delta\mathbf{v}_{e,N}^e + \mathbf{R}_{en} \cdot \mathbf{n}_{\delta v,N}^n \quad (\text{III-223})$$

Important to notice is, that the transformation matrix \mathbf{R}_{en} is time dependent.



III-153 Gauss Markov ode in n-frame



III-154 Gauss Markov in e-frame

The left figure above shows the uncorrelated Gauss Markov processes in n-frame. The right figure compares the Gauss Markov process, generated according to equation (III-223) with the Gauss

Markov process, being transformed from n-frame to e-frame by using

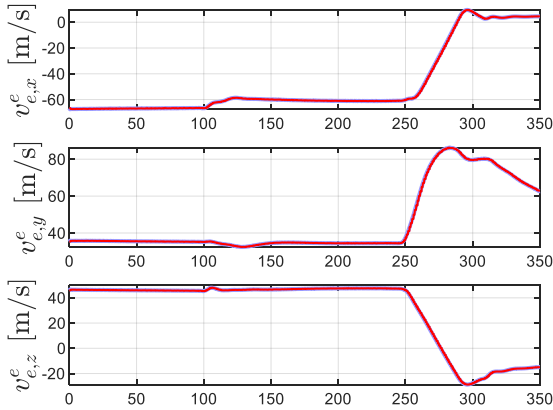
$$\delta \mathbf{v}_{e,N}^e = \mathbf{R}_{en} \cdot \delta \mathbf{v}_{e,N}^n \quad (\text{III-224})$$

This representation of the Gauss Markov process, using n-frame characteristics and e-frame coordinates, is necessary for aided vector tracking approach. There, this model can directly be integrated in the Kalman filter state space system.

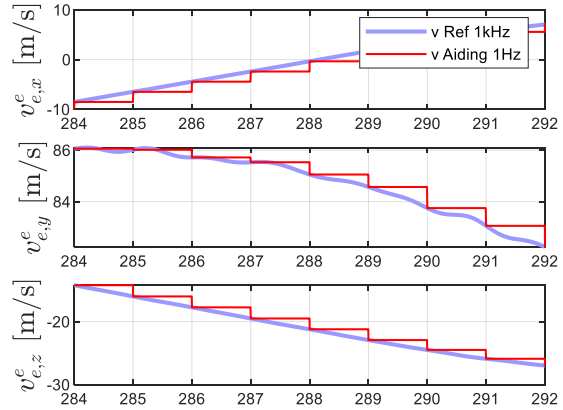
III - 9.4 Low aiding rate

The dynamic model within an optimal filter-based tracking architecture is propagated at a defined rate. In this thesis, the propagation rate equals the used correlation time of 1 kHz. In the optimal case, the aiding should be available and applied with the same rate.

But the typical aiding rate is lower, mostly 1 Hz. Between two subsequent aiding updates, the aiding input $\tilde{\mathbf{v}}_{e,Aid,k}^n$ is kept constant. The following figures show the resulting aiding error if the aiding rate is 1Hz, compared to an optimal aiding rate of 1kHz.



III-155 Comparing aiding 1kHz vs. 1Hz



III-156 Comparing aiding 1kHz vs. 1Hz - ZOOMED

The resulting error between the high rate velocity propagation and the low aiding rate must be considered within the tracking filters. An estimation of this error does not make sense, because the error becomes zero at every new aiding update.

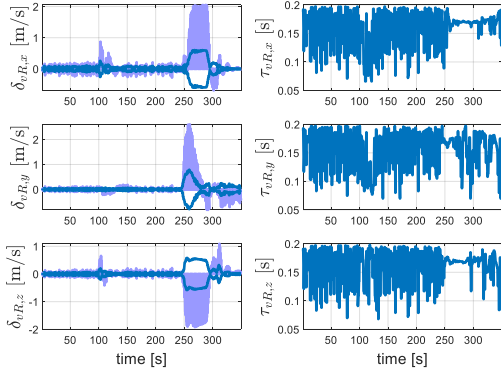
The error due to low aiding rate will be modeled as a Gauss Markov process.

The rate aiding error is written as

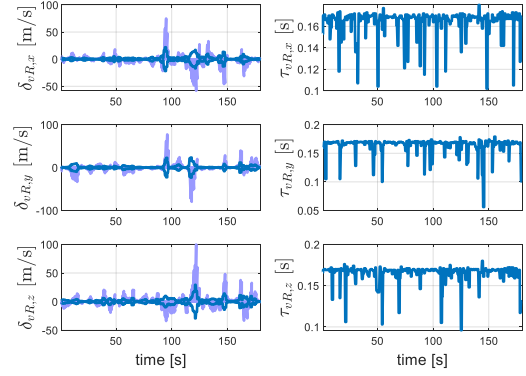
$$\delta \mathbf{v}_{e,R}^e = \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,Aid,1kHz}^n - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,Aid,1Hz}^n \quad (\text{III-225})$$

For tuning the corresponding Gauss Markov process, the characteristic figures like correlation time constant and error variance of this error must be determined.

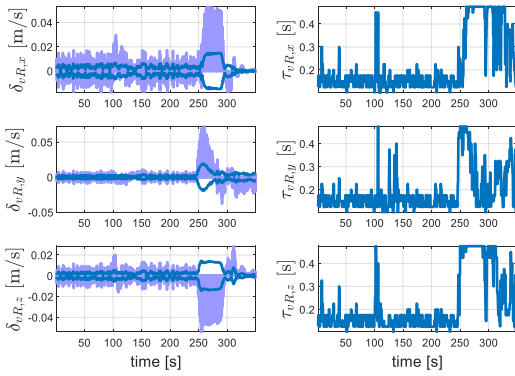
The following two figures show the estimated correlation time constants and the variance of the error processes for the DA42 scenario and for the high dynamic trajectory.



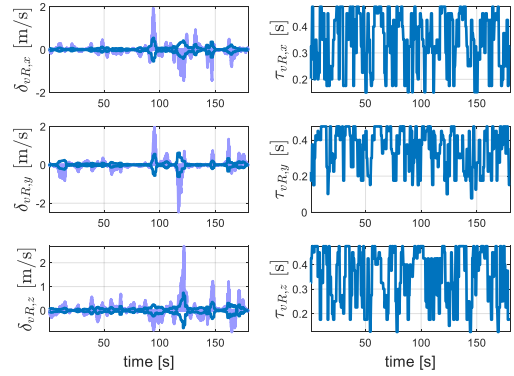
III-157 Aiding error characteristics (DA42, 1Hz)



III-158 Aiding error characteristics (High Dynamic, 1Hz)



III-159 Aiding error characteristics (DA42, 40Hz)



III-160 Aiding error characteristics (High Dynamic, 40Hz)

The aiding error caused by a low aiding rate becomes high, every time the platform dynamic changes very fast.

The remaining dynamic between the aiding updates is modeled as a Gauss Markov process using the following system model.

$$\delta \dot{\mathbf{v}}_{e,R}^e = \mathbf{A}_{\delta v,R}^e \cdot \delta \mathbf{v}_{e,R}^e + \mathbf{n}_{e,\delta v R}^e$$

$$\begin{bmatrix} \delta v_{e,x,R}^e \\ \delta v_{e,y,R}^e \\ \delta v_{e,z,R}^e \end{bmatrix} = \begin{bmatrix} \frac{-1}{\tau_{\delta v_{e,x,R}^e}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v_{e,y,R}^e}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v_{e,z,R}^e}} \end{bmatrix} \cdot \begin{bmatrix} \delta v_{e,x,R}^e \\ \delta v_{e,y,R}^e \\ \delta v_{e,z,R}^e \end{bmatrix} + \begin{bmatrix} n_1 \cdot \sqrt{\frac{\sigma_{\delta v_{e,x,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,x,R}^e}}} \\ n_2 \cdot \sqrt{\frac{\sigma_{\delta v_{e,y,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,y,R}^e}}} \\ n_3 \cdot \sqrt{\frac{\sigma_{\delta v_{e,z,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,z,R}^e}}} \end{bmatrix} \quad (\text{III-226})$$

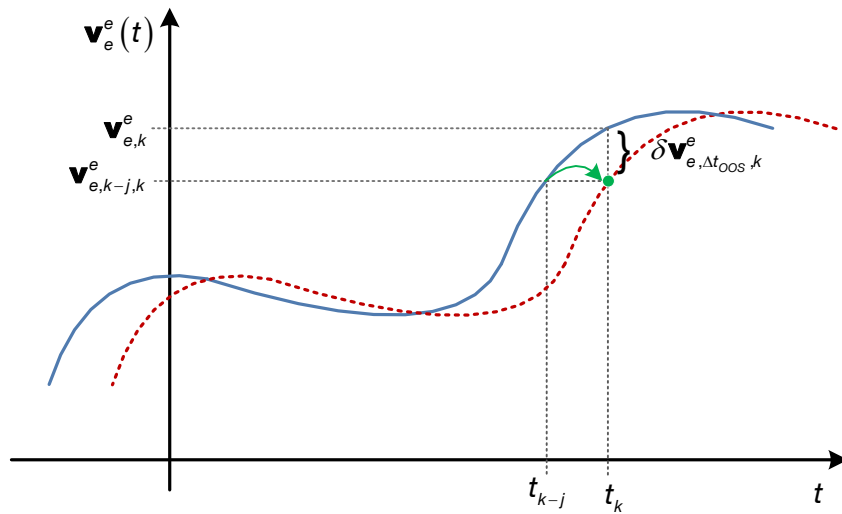
The characteristic figures for the Gauss Markov process are derived out of the derived velocity $\delta \mathbf{v}_{e,R}^e$ error in e-frame. Given some approximate trajectory, the characteristics can be derived pre mission.

III - 9.5 Aiding delay

The velocity aiding information available at the receiver, typically has a time delay Δt_{oos} . Depending on the sampling rate, this time delay equals j samples. The available aiding information at time k is valid for time $k - j$, due to transmission delay or internal processing delay of the INS.

The graphic below visualizes the effect of delayed aiding information. The true velocity at time k would be $\mathbf{v}_{e,k}^e$, but available is only $\mathbf{v}_{e,k-j}^e$, which will be written as $\mathbf{v}_{e,k-j,k}^e$.

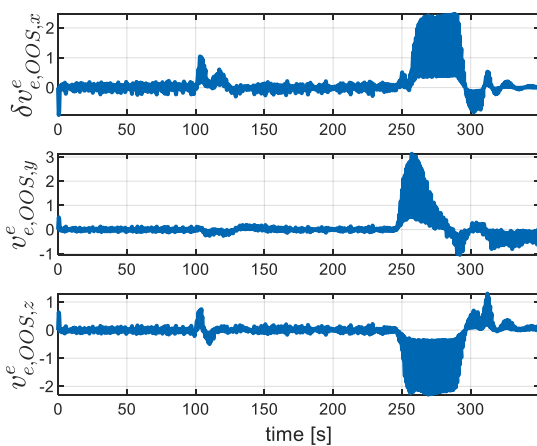
The velocity error at time k , caused by the delayed measurement, is $\delta \mathbf{v}_{\text{OOS},k}^e$.



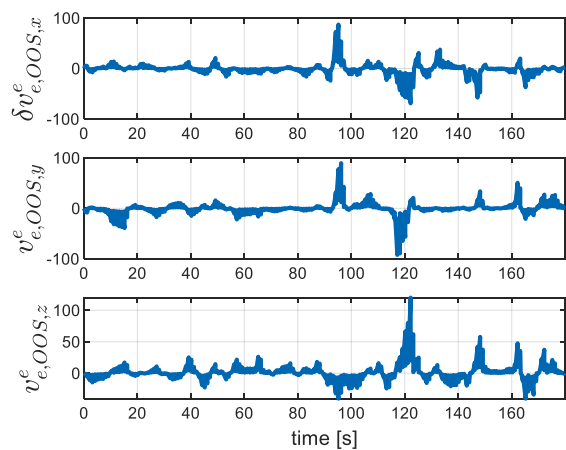
III-161 Delayed aiding

The following two figures show the aiding velocity error, caused by the delayed aiding.

$$\delta \mathbf{v}_{e,\text{OOS}}^e(t) = \mathbf{v}_e^e(t) - \mathbf{v}_e^e(t - \Delta t_{\text{OOS}}) \quad (\text{III-227})$$



III-162 OOS velocity aiding error - DA42



III-163 OOS velocity aiding error - Fighter dynamic

III - 9.6 Tilt error

The aiding velocity is related to the ECEF coordinates, but represented in n-frame coordinates, because typically the IMU mechanization is realized in n-frame coordinates. In case of any misalignment of IMU axis with regard to the vehicle body frame axis, there will be some tilt errors in the n-frame velocity. For IMU misalignment only pitch and roll angle errors are considered.

Starting from a theoretical correct n-frame velocity, the tilted aiding velocity gets

$$\mathbf{v}_e^{\tilde{n}} = \mathbf{R}_{\tilde{n}n} \cdot \mathbf{v}_e^n \quad (\text{III-228})$$

$$\mathbf{R}_{\tilde{n}n} = \begin{pmatrix} \cos \Delta\Psi \cdot \cos \Delta\Theta & \sin \Delta\Psi \cdot \cos \Delta\Theta & -\sin \Delta\Theta \\ \cos \Delta\Psi \cdot \sin \Delta\Theta \cdot \sin \Delta\Phi - \sin \Delta\Psi \cdot \cos \Delta\Phi & \sin \Delta\Psi \cdot \sin \Delta\Theta \cdot \sin \Delta\Phi + \cos \Delta\Psi \cdot \cos \Delta\Phi & \cos \Delta\Theta \cdot \sin \Delta\Phi \\ \cos \Delta\Psi \cdot \sin \Delta\Theta \cdot \cos \Delta\Phi - \sin \Delta\Psi \cdot \sin \Delta\Phi & \sin \Delta\Psi \cdot \sin \Delta\Theta \cdot \cos \Delta\Phi + \cos \Delta\Psi \cdot \sin \Delta\Phi & \cos \Delta\Theta \cdot \cos \Delta\Phi \end{pmatrix} \quad (\text{III-229})$$

The angles are defined as

$$\begin{aligned} \Delta\Psi &:= \text{azimuth angle tilt} \rightarrow \text{is assumed to be zero} \\ \Delta\Theta &:= \text{pitch angle tilt} \\ \Delta\Phi &:= \text{bank angle tilt} \\ \text{Positive angles are defined for } n &\rightarrow \tilde{n} \end{aligned} \quad (\text{III-230})$$

With the azimuth angle being zero, equation (III-229) becomes

$$\mathbf{R}_{\tilde{n}n} = \begin{pmatrix} \cos \Delta\Theta & 0 & -\sin \Delta\Theta \\ \sin \Delta\Theta \cdot \sin \Delta\Phi & \cos \Delta\Phi & \cos \Delta\Theta \cdot \sin \Delta\Phi \\ \sin \Delta\Theta \cdot \cos \Delta\Phi & \sin \Delta\Phi & \cos \Delta\Theta \cdot \cos \Delta\Phi \end{pmatrix} \quad (\text{III-231})$$

The estimated tilt angle errors are assumed to be $\Delta\hat{\Phi}$ and $\Delta\hat{\Theta}$. In order to get the residual velocity error, caused by the difference between the real tilt angle errors and the estimated one, the following equation is introduced.

$$\delta\mathbf{v}_e^{\tilde{n}} = \mathbf{v}_e^{\tilde{n}} - \mathbf{R}_{\tilde{n}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \mathbf{R}_{ne} \cdot \mathbf{v}_e^e \quad (\text{III-232})$$

This error, in e-frame coordinates gets

$$\delta\mathbf{v}_{e,T}^e = \mathbf{R}_{en} \cdot \mathbf{R}_{n\tilde{n}}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \delta\mathbf{v}_e^{\tilde{n}} = \mathbf{R}_{en} \cdot \mathbf{R}_{n\tilde{n}}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \mathbf{v}_e^{\tilde{n}} - \mathbf{v}_e^e \quad (\text{III-233})$$

The following figures show the resulting velocity error in ECEF coordinates, given the following tilt errors

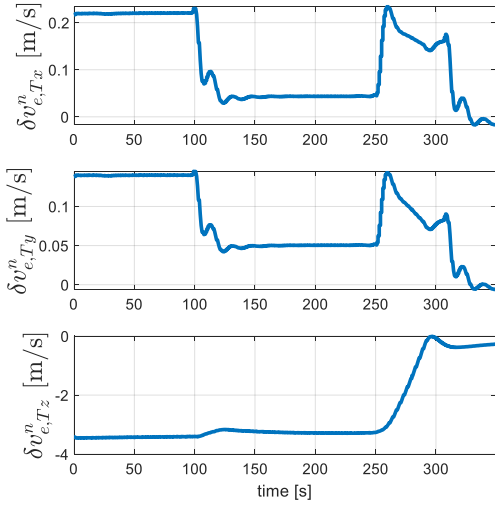
$$\begin{aligned} \Delta\Phi &= -1^\circ \\ \Delta\Theta &= 2^\circ \end{aligned} \quad (\text{III-234})$$

The estimated tilt errors are assumed as

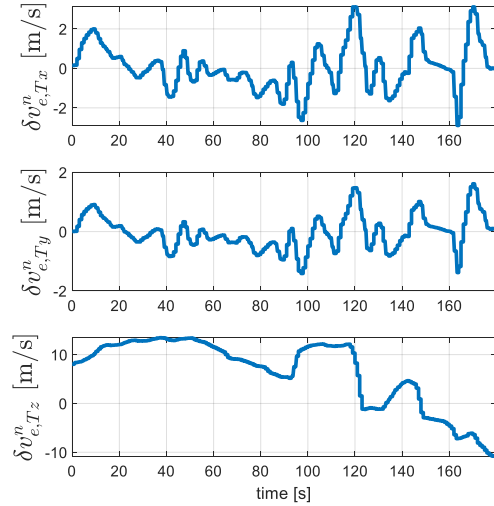
$$\begin{aligned}\Delta\hat{\Phi} &= 0^\circ \\ \Delta\hat{\Theta} &= 0^\circ\end{aligned}\tag{III-235}$$

The resulting error in NED coordinates gets

$$\delta\mathbf{v}_{e,T}^n = \mathbf{R}_{n\tilde{n}}(\Delta\hat{\Phi} = 0^\circ, \Delta\hat{\Theta} = 0^\circ) \cdot \mathbf{v}_e^{\tilde{n}} - \mathbf{v}_e^n = \mathbf{v}_e^{\tilde{n}} - \mathbf{v}_e^n\tag{III-236}$$



III-164 Tilt aiding error (DA42)



III-165 Tilt aiding error (High dynamic)

III - 10 Aided error state vector tracking

III - 10.1 Motivation

The part on aided tracking architectures starts with the tracking architecture, providing the best results, being the aided error state vector tracking approach.

In literature some publications can be found on error state vector tracking, as for example in [70] [19], [71] and [72], but all these publications do not consider in detail the different aiding errors.

In this section, for all introduced aiding errors, an appropriate modeling, consideration or estimation concept within the error state tracking architecture, will be developed and analyzed.

Moreover, two different approaches, of how aiding can be applied to the tracking filter, are developed and compared.

Also special tuning measures are evaluated, improving the aided tracking performance.

There will be an evaluation, what's the minimum necessary aiding quality to outperform the best unaided tracking architecture.

In subsequent sections, also other aided tracking architectures are developed, like a total state aided vector approach or even some scalar aided architectures. For all those different aided tracking architectures, the error state vector tracking approach is used as reference, in order to show the disadvantages of all others.

III - 10.2 Tracking architecture – Aiding as “control input”

III - 10.2.1 Noise like aiding error + Rate aiding error

In this section, an aided error state architecture will be developed, consisting out of an error state vector tracking filter and an external total state dynamic model.

The aiding is applied to the total state dynamic model as a kind of “control input”.

An essential difference of the aided error state approach compared to the unaided error state approach is, that within the error state Kalman filter, no platform dynamic is present. The error state Kalman filter above only has to care about the aiding error dynamic, the receiver and satellite clock errors, as well as the dynamic of ionosphere and troposphere errors and ephemeris errors.

An optimal tuning regarding the aiding errors to be estimated or considered is possible and the error state filter can differ between the different error components.

In the following, step by step, the filter equations will be developed for different aiding error types.

In the first step, only noise like aiding errors and rate aiding errors are assumed.

The following equation gives the external total state dynamic model, with applied aiding as control input.

$$\dot{\mathbf{z}} = \mathbf{A}_z \cdot \mathbf{z} + \mathbf{B}_z \cdot \mathbf{u} \quad (III-237)$$

$$\begin{bmatrix} \dot{\mathbf{x}}_e \\ c \cdot \dot{t}_{clk} \\ c \cdot \dot{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_e \\ c \cdot t_{clk} \\ c \cdot d_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{R}_{en} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \cdot \tilde{\mathbf{v}}_{e,Aid}^n$$

The error state tracking filter is given below.

$$\delta \dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta \mathbf{z} + \mathbf{n}_{\delta z} \quad (III-238)$$

$$\begin{bmatrix} \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_{e,R}^e \\ \delta \dot{\mathbf{v}}_{e,N}^e \\ c \cdot \delta \dot{t}_{clk} \\ c \cdot \delta \dot{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta ve,R} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R}^e \\ \delta \mathbf{v}_{e,N}^e \\ c \cdot \delta t \\ c \cdot \delta d \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{n}_{\delta vR}^e \\ \mathbf{R}_{en} \cdot \mathbf{n}_{\delta vN}^n \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix}$$

The only errors being considered within this first error state tracking approach are

$$\begin{aligned}
&\text{clock error: } \Delta t_{clk} \\
&\text{clock drift: } \Delta d_{clk} \\
&\text{noise like aiding error: } \delta \mathbf{v}_{e,N}^e \\
&\text{rate aiding error: } \delta \mathbf{v}_{e,R}^e
\end{aligned} \tag{III-239}$$

Modeling of noise like aiding error

The noise like aiding error cannot be estimated, it is only realized as a consider state. The noise like aiding error is brought to the tracking system via the total state dynamic model and therefore leads to a noisy NCO command. Finally, the noise like aiding error shows up as an error component in the discriminator measurement, besides the noise component caused by the received noise at the GPS antenna.

As already introduced in III - 9.3 , the noise like aiding error is modeled as a Gauss Markov process.

$$\delta \dot{\mathbf{v}}_{e,N}^e = \left(\mathbf{R}_{en} \cdot \boldsymbol{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N}^n \cdot \mathbf{R}_{en}^T \right) \cdot \delta \mathbf{v}_{e,N}^e + \mathbf{R}_{en} \cdot \mathbf{n}_{\delta v,N}^n \tag{III-240}$$

In order to consider this error within the measurement, the modeled error velocity is coupled with the positioning error within the error dynamic model.

Despite this coupling, the velocity error is only a consider state, because there is no propagation of error dynamics. After each innovation update, only some error states are used to update the total states. But all error states are set to zero after update. Because the velocity error $\delta \mathbf{v}_{e,N}^e$ is not used for update, the error itself and its coupling within the error state dynamic model, is only considered in the error covariance propagation and Kalman gain calculation.

The reason why it makes sense to consider the noise aiding error within the error dynamic model and not within the observation matrix $\mathbf{H}_{\delta z}^T$ becomes clear by the following argumentation. If the error is coupled within the dynamic model, given a high noise like aiding error variance, the modeled positioning error state $\delta \mathbf{x}_e$ becomes more uncertain and the Kalman filter relies more on the discriminator measurement. Due to the architecture, where aiding is applied to the NCO command by the total state aiding integration dynamic model, the NCO command itself contains the noise like aiding error and therefore also the discriminator measurement. Due to the modeled dynamic characteristics of the aiding error within the error state dynamic model, the Kalman filter finds the optimal balance between discriminator noise caused by normal antenna noise and discriminator noise caused by aiding. In case of low noise like aiding error variance and for example high antenna noise, the Kalman filter trusts its own error dynamic model and less the discriminator measurement. If the aiding error $\delta \mathbf{v}_{e,N}^e$ would be coupled within the observation matrix, the Kalman filter cannot differ between measurement noise caused by noise GPS signals and a noisy aiding error.

Modeling of rate aiding error

The rate aiding error is also modeled as Gauss Markov process according to following equation.

$$\delta \dot{\mathbf{v}}_{e,R}^e = \mathbf{A}_{\delta v,R}^e \cdot \delta \mathbf{v}_{e,R}^e + \mathbf{n}_{e,\delta v R}^e \quad (\text{III-241})$$

One might ask, why the Gauss Markov characteristics are written in ECEF frame, in contrast to the characteristics of the noise like aiding error. The reason is, that for the noise like aiding error, the Gauss Markov characteristics are given in NED frame. The characteristics for the rate aiding error are derived directly in ECEF frame.

The rate aiding error will be also implemented as a consider state.

For considering the effect of the rate aiding error in the estimated position error, $\delta \mathbf{v}_{e,R}^e$ is also coupled within the error dynamic model with the estimated position error. Due to this coupling, a low aiding rate and respectively high rate error forces the Kalman filter to rely less on the error dynamic model but much more on the measurement. As a result, the Kalman filter compensates the residual dynamic caused by low aiding rate, with the drawback of a higher closed loop tracking bandwidth and therefore less improvement regarding antijam.

After each innovation update, all error states are set to zero, therefore the modeled rate aiding error behaves as a consider state, because it is not used to update the external total states.

The other option would be considering the rate aiding error within the observation matrix $\mathbf{H}_{\delta \mathbf{z}}^T$. But in this case, a high rate aiding error and therefore a high rate aiding error variance would force the Kalman filter, rely less on the discriminator measurement. The total state would be not compensated in a proper way to counteract the rate aiding errors. The tracking loop and therefore the discriminator measurement show the resulting code phase deviation, which is not compensated by aiding. Therefore, it is important that the Kalman filter takes the discriminator measurements to update the states in order to compensate the rate aiding error.

Error components within discriminator measurement

The discriminator measurement contains in this scenario the following error components

$$\delta \tilde{\tau}_{\#Sv} = \frac{\delta \rho_{\#Sv}}{c} + \delta t_{clk} + \delta t_{vR} + \delta t_{vN} + n_{\delta \tau} \quad (\text{III-242})$$

In this scenario, only the clock error, the noise like aiding error and the rate aiding error are stimulated. All other error components, like satellite clock error, ionosphere error, troposphere error or ephemeris errors, are not considered.

The observation equation for the error state tracking filter gets

$$\mathbf{H}_{\delta \mathbf{z}}^T = \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \end{bmatrix} \quad (\text{III-243})$$

The model error covariance matrix is given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v N} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v R} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & c^2 \sigma_{\delta t clk}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 d_{\delta t clk}^2 \end{bmatrix} \quad (\text{III-244})$$

$$\mathbf{Q}_{\delta v N} = \mathbf{R}_{en} \cdot \begin{bmatrix} \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,d,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,d,N}^n}} \end{bmatrix} \cdot \mathbf{R}_{en}^T, \quad (\text{III-245})$$

$$\mathbf{Q}_{\delta v R} = \begin{bmatrix} \frac{\sigma_{\delta v_{e,x,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,x,R}^e}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,y,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,y,R}^e}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,z,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,z,R}^e}} \end{bmatrix}$$

For NCO control, a nonlinear transformation from the position \mathbf{x}_e in ECEF coordinates to the pseudoranges is necessary.

$$\begin{bmatrix} \hat{\rho}_{k+1,\#1} \\ \vdots \\ \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{\#1}(\hat{\mathbf{x}}_{e,k+1}, \hat{t}_{clk,k+1}) \\ \vdots \\ \mathbf{h}_{\#Nsat}(\hat{\mathbf{x}}_{e,k+1}, \hat{t}_{clk,k+1}) \end{bmatrix} = \begin{bmatrix} \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_{e,k+1})^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_{e,k+1})} + c \cdot \hat{t}_{clk,k+1} \\ \vdots \\ \sqrt{(\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_{e,k+1})^T \cdot (\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_{e,k+1})} + c \cdot \hat{t}_{clk,k+1} \end{bmatrix} \quad (\text{III-246})$$

For NCO control only the pseudorange change since the beginning of tracking is relevant. Therefore, the applied NCO control gets

$$\begin{bmatrix} \Delta \hat{\rho}_{k+1,\#1} \\ \vdots \\ \Delta \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{k+1,\#1} \\ \vdots \\ \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} - \begin{bmatrix} \hat{\rho}_{0,\#1} \\ \vdots \\ \hat{\rho}_{0,\#Nsat} \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{k+1,\#1} \\ \vdots \\ \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} - \begin{bmatrix} \sqrt{(\mathbf{x}_{\#1,0} - \hat{\mathbf{x}}_{e,0})^T \cdot (\mathbf{x}_{\#1,0} - \hat{\mathbf{x}}_{e,0})} \\ \vdots \\ \sqrt{(\mathbf{x}_{\#Nsat,0} - \hat{\mathbf{x}}_{e,0})^T \cdot (\mathbf{x}_{\#Nsat,0} - \hat{\mathbf{x}}_{e,0})} \end{bmatrix} \quad (\text{III-247})$$

The following sequence shows the explicit realization of the different steps while filter execution.

#1 total state propagation: $\mathbf{z}_{k+1}^- = (\mathbf{I}^{8 \times 8} + \Delta T \cdot \mathbf{A}_z) \cdot \mathbf{z}_k^+ + \Delta T \cdot \mathbf{B}_z \cdot \mathbf{u}_k$

#2 NCO command application:
$$\begin{bmatrix} \Delta \hat{\rho}_{k+1, \#1} \\ \vdots \\ \Delta \hat{\rho}_{k+1, \#N_{\text{sat}}} \end{bmatrix}$$

#3 error state propagation: $\delta \mathbf{z}_{k+1}^- = \Phi_{\delta \mathbf{z}} \cdot \delta \mathbf{z}_k^+$

#4 error state covariance propagation: $\mathbf{P}_{\delta \mathbf{z}} = \Phi_{\delta \mathbf{z}} \cdot \mathbf{P}_{\delta \mathbf{z}} \cdot \Phi_{\delta \mathbf{z}}^T + \mathbf{Q}_{\delta \mathbf{z}}$

#5 calculation of Kalman gain: $\mathbf{K}_{\delta \mathbf{z}} = (\mathbf{P}_{\delta \mathbf{z}} \cdot \mathbf{H}_{\delta \mathbf{z}}^T) \cdot (\mathbf{H}_{\delta \mathbf{z}} \cdot \mathbf{P}_{\delta \mathbf{z}} \cdot \mathbf{H}_{\delta \mathbf{z}}^T + \mathbf{R}_{\delta \mathbf{z}})^{-1}$

#6 error state update: $\delta \mathbf{z}_{k+1}^+ = \delta \mathbf{z}_{k+1}^- + \mathbf{K}_{11 \times N_{\text{sat}}} \cdot (\delta \mathbf{r}) = \mathbf{K}_{11 \times N_{\text{sat}}} \cdot (\delta \mathbf{r})$

#7 total state correction:
$$\begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{\text{clk}} \\ \mathbf{c} \cdot d_{\text{clk}} \end{bmatrix}_k^+ = \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{\text{clk}} \\ \mathbf{c} \cdot d_{\text{clk}} \end{bmatrix}_k^- + \begin{bmatrix} \delta \mathbf{x}_e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix}_k^+$$

#8 error state reset:
$$\begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R}^e \\ \delta \mathbf{v}_{e,N}^e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix}_k^+ = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \text{ only when new aiding update is applied} \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix}$$

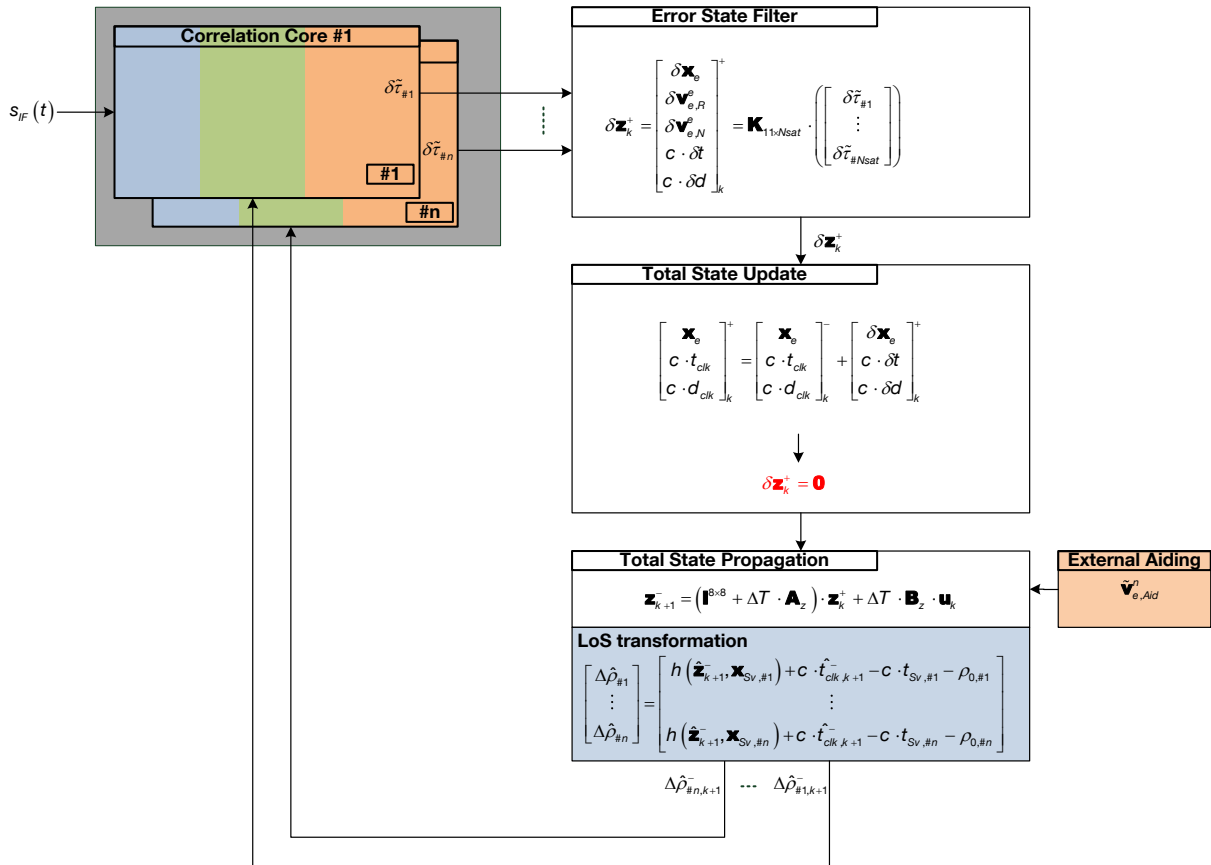
The estimated position error, estimated clock error and estimated clock error drift, are used after each Kalman filter update, to correct the total states.

The estimated position error $\delta \mathbf{x}_e$ is set to zero after each update, like the estimated clock error and estimated clock drift.

The estimated noise aiding error $\delta \mathbf{v}_{e,N}^e$ is also set to zero after each Kalman filter update. But the noise aiding error is not stored in some external total state. Due to this, the noise aiding error is only a consider state.

The estimated rate aiding error $\delta \mathbf{v}_{e,R}^e$ is not set to zero within the aiding interval. The aiding error is estimated. After a new aiding update is applied, the rate aiding error is zero and the estimated error is also set to zero.

The following figure shows the introduced aided error state tracking architecture, where the aiding is applied as a control input.



Problems of this architecture

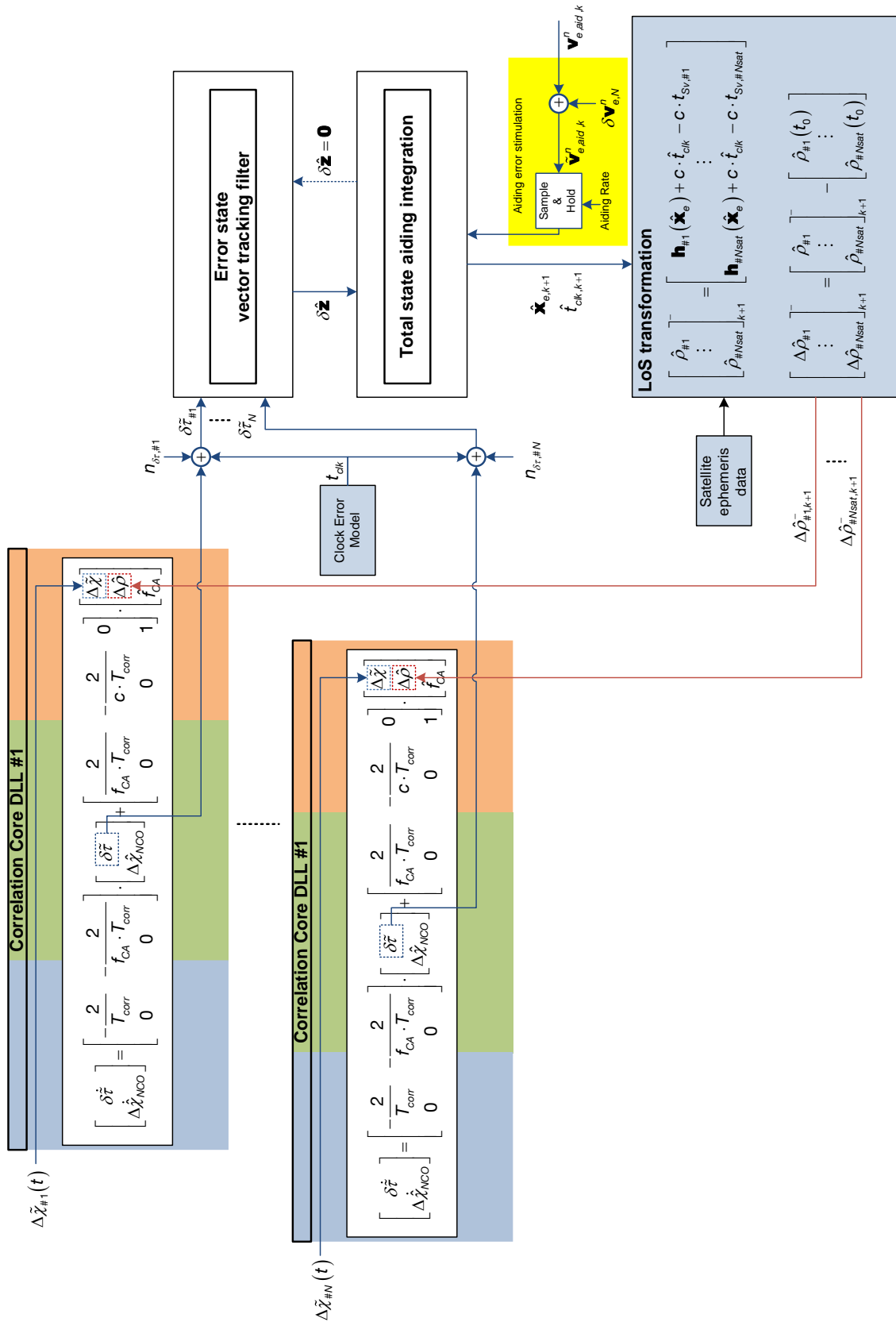
This first error state architecture suffers two major drawbacks. At first, because the aiding enters the tracking loop as a kind of control input, it is not possible to prevent aiding of bad quality from entering the tracking loop and deteriorating the tracking performance. The aiding errors are included in the discriminator measurement, because the integrated velocity aiding is applied as NCO command. In case of bad aiding quality, the only option for the error state Kalman filter is to separate the aiding errors within the discriminator measurements from all the other error components like receiver clock error and UERE's (ionosphere error, troposphere error, ephemeris error, satellite clock error). Additionally, using this architecture, it is not possible to estimate the actual platform acceleration, which is a mandatory information in order to make tilt aiding errors and delay aiding errors observable, as will be shown in the next sections.

III - 10.3 Equivalent base band dynamic state space simulation – aiding as control input

III - 10.3.1 Distributed state space simulation

In the following the already introduced distributed state space simulation approach in equivalent base band is used, to evaluate the behavior of this tracking architecture pre mission, in different noise and dynamic scenarios.

The following figure shows the non-steady state space simulation set up.



III-167 Distributed state space representation - including noise and low aiding rate error

The only difference between the real tracking implementation and the distributed state space simulation, is the equivalent base band representation of the correlation core. The error state filter

and external total state dynamic implementations are equal.

Within the distributed state space simulation, also the aiding errors are stimulated, as given in the yellow marked box.

III - 10.3.2 Centralized state space representation

In order to get a centralized state space representation or parametric model of the aided error state vector tracking loop, at first all differential equations will be written down, which are necessary to describe the complete behavior of the tracking architecture.

The differential equations describing the correlation core are given below.

$$\begin{aligned}
\delta\dot{\tilde{\tau}}_{\#1} &= -\frac{2}{T_{corr}} \cdot \delta\tilde{\tau}_{\#1} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\hat{X}_{NCO,\#1} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\tilde{\chi}_{\#1} - \frac{2}{c \cdot T_{corr}} \cdot \left(h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}) + c \cdot \hat{t}_{clk} \right) \\
\Delta\dot{\hat{X}}_{NCO,\#1} &= 0 \\
&\vdots \\
\delta\dot{\tilde{\tau}}_{\#N} &= -\frac{2}{T_{corr}} \cdot \delta\tilde{\tau}_{\#N} - \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\hat{X}_{NCO,\#N} + \frac{2}{f_{CA} \cdot T_{corr}} \cdot \Delta\tilde{\chi}_{\#N} - \frac{2}{c \cdot T_{corr}} \cdot \left(h(\mathbf{x}_{Sv\#N}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#N,0}, \hat{\mathbf{x}}_{e,0}) + c \cdot \hat{t}_{clk} \right) \\
\Delta\dot{\hat{X}}_{NCO,\#n} &= 0
\end{aligned} \tag{III-248}$$

The line of sight transformation from the positioning solution to the estimated pseudorange in equation (III-248) is nonlinear.

The differential equations describing the total state aiding integration are given in equation (III-237)

$$\dot{\mathbf{z}} = \mathbf{A}_z \cdot \mathbf{z} + \mathbf{B}_z \cdot \mathbf{u} \tag{III-249}$$

The continuous Kalman filter differential equation gets

$$\delta\dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta\mathbf{z} + \frac{1}{\Delta T} \cdot \mathbf{K} \cdot (\delta\tilde{\mathbf{r}}) \tag{III-250}$$

The Kalman gain matrix \mathbf{K} is the one of the discrete Kalman filter. For continuous Kalman filters, the Kalman gain must be normed by the sampling time of discrete Kalman filter implementation.

All these differential equations are now combined in one matrix differential equation system of 1st order, whereby also the correction of the total states with the error states and the subsequent zero reset of the error states need to be considered.

The correction of the total states with the actual error states is given below.

$$\begin{bmatrix} \mathbf{x}_e \\ c \cdot t_{clk} \\ c \cdot d_{clk} \end{bmatrix}_k^+ = \begin{bmatrix} \mathbf{x}_e \\ c \cdot t_{clk} \\ c \cdot d_{clk} \end{bmatrix}_k^- + \begin{bmatrix} \delta\mathbf{x}_e \\ c \cdot \delta t \\ c \cdot \delta d \end{bmatrix}_k^+ \tag{III-251}$$

In the continuous differential equations, the correction of the total states by the error states and the total state propagation can be combined.

$$\begin{bmatrix} \dot{\mathbf{x}}_e \\ \mathbf{c} \cdot \dot{t}_{clk} \\ \mathbf{c} \cdot \dot{d}_{clk} \end{bmatrix} = \mathbf{A}_z \cdot \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix} + \frac{1}{T_{corr}} \cdot \begin{bmatrix} \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 1 & 0 \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R}^e \\ \delta \mathbf{v}_{e,N}^e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix} + \mathbf{B}_z \cdot \tilde{\mathbf{v}}_{e,Aid}^n \quad (\text{III-252})$$

In discrete form of equation (III-252) gets

$$\begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix}_{k+1} = \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix}_k + T_{corr} \cdot \mathbf{A}_z \cdot \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix}_k + \frac{T_{corr}}{T_{corr}} \cdot \begin{bmatrix} \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 1 & 0 \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,N}^e \\ \delta \mathbf{v}_{e,R}^e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix}_k + T_{corr} \cdot \mathbf{B}_z \cdot \tilde{\mathbf{v}}_{e,Aid}^n \quad (\text{III-253})$$

The zero reset of the error states, after updating the external total states, can also be written as a continuous differential equation.

$$\delta \dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta \mathbf{z} - \frac{1}{T_{corr}} \cdot \begin{bmatrix} \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & 1 & 0 \\ \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,N}^e \\ \delta \mathbf{v}_{e,R}^e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix} + \frac{1}{T_{corr}} \cdot \mathbf{K} \cdot (\delta \tilde{\mathbf{r}}) \quad (\text{III-254})$$

Due to equations (III-248), the centralized state space differential equation is nonlinear.

$$\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}, \mathbf{u}) \quad (\text{III-255})$$

For tracking bandwidth calculation, the nonlinear differential equation system needs to be linearized for every point on trajectory.

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{F}(\hat{\mathbf{z}}_k, \mathbf{u}_k) + \left. \frac{\partial \mathbf{F}(\mathbf{z}, \mathbf{u})}{\partial \mathbf{z}} \right|_{\hat{\mathbf{z}}_k, \mathbf{u}_k} \cdot \delta \mathbf{z} + \left. \frac{\partial \mathbf{F}(\mathbf{z}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\hat{\mathbf{z}}_k, \mathbf{u}_k} \cdot \delta \mathbf{u} \\ \dot{\mathbf{z}} - \mathbf{F}(\hat{\mathbf{z}}_k, \mathbf{u}_k) &= \delta \dot{\mathbf{z}} = \mathbf{A} \cdot \delta \mathbf{z} + \mathbf{B} \cdot \delta \mathbf{u} \end{aligned} \quad (\text{III-256})$$

The detailed structure of the linearized state space differential equation system is given in the following equation.

$$\begin{bmatrix} \delta \ddot{x}_{\#1} \\ \Delta \dot{\ddot{x}}_{NCO,\#1} \\ \vdots \\ \delta \ddot{x}_{\#7} \\ \Delta \dot{\ddot{x}}_{NCO,\#7} \\ \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_{e,R}^p \\ \delta \dot{\mathbf{v}}_{e,N}^p \\ \mathbf{C} \cdot \delta \dot{\mathbf{d}}_{clk} \\ \mathbf{C} \cdot \delta \dot{\mathbf{d}}_{clk} \\ \dot{\mathbf{x}}_e \\ \mathbf{C} \cdot \dot{\mathbf{t}}_{clk} \\ \mathbf{C} \cdot \dot{\mathbf{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\delta t,\#1}^{2 \times 2} & \dots & \mathbf{0}^{2 \times 2} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 1} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 1} & \mathbf{0}^{2 \times 1} & \mathbf{A}_{h,\#1}^{2 \times 3} & \mathbf{0}^{2 \times 1} & \mathbf{0}^{2 \times 1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}^{2 \times 2} & \dots & \mathbf{A}_{\delta t,\#N}^{2 \times 2} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{A}_{h,\#N}^{2 \times 3} & \mathbf{0}^{2 \times 1} & \mathbf{0}^{2 \times 1} \\ \mathbf{A}_{k,\delta x,\#1}^{3 \times 2} & \dots & \mathbf{A}_{k,\delta x,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{A}_{k,\delta R,\#1}^{3 \times 2} & \dots & \mathbf{A}_{k,\delta R,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{A}_{k,\delta N,\#1}^{3 \times 2} & \dots & \mathbf{A}_{k,\delta N,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{A}_{k,\delta t,\#1}^{1 \times 2} & \dots & \mathbf{A}_{k,\delta t,\#N}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{A}_{k,\delta d,\#1}^{1 \times 2} & \dots & \mathbf{A}_{k,\delta d,\#N}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 2} & \dots & \mathbf{0}^{3 \times 2} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{1 \times 2} & \dots & \mathbf{0}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0} & \mathbf{1} \\ \mathbf{0}^{1 \times 2} & \dots & \mathbf{0}^{1 \times 2} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{11}^{14 \times 7} & \mathbf{0}^{14 \times 3} \\ \mathbf{0}^{11 \times 7} & \mathbf{0}^{11 \times 3} \\ \mathbf{0}^{3 \times 7} & \mathbf{R}_{en} \\ \mathbf{0}^{2 \times 7} & \mathbf{0}^{2 \times 3} \end{bmatrix} \cdot \begin{bmatrix} \Delta \dot{\ddot{x}}_{\#1} \\ \vdots \\ \Delta \dot{\ddot{x}}_{\#7} \\ \dot{\mathbf{v}}_{e,Aid}^p \end{bmatrix}$$

(III-257)

$$\mathbf{A}_{\delta\tau\#1}^{2 \times 2} = \dots = \mathbf{A}_{\delta\tau\#1}^{2 \times 2} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \quad (\text{III-258})$$

$$\mathbf{A}_{h\#1}^{2 \times 3} = \begin{bmatrix} \frac{\partial(h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}))}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \dots$$

$$\begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{\partial \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)}}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{h\#N}^{2 \times 3} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#7,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#7,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{k\delta x\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{11}}{T_{corr}} & 0 \\ \frac{k_{21}}{T_{corr}} & 0 \\ \frac{k_{31}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta x\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{1,7}}{T_{corr}} & 0 \\ \frac{k_{2,7}}{T_{corr}} & 0 \\ \frac{k_{3,7}}{T_{corr}} & 0 \end{bmatrix} \quad \mathbf{A}_{k\delta R\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{41}}{T_{corr}} & 0 \\ \frac{k_{51}}{T_{corr}} & 0 \\ \frac{k_{61}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta R\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{47}}{T_{corr}} & 0 \\ \frac{k_{57}}{T_{corr}} & 0 \\ \frac{k_{67}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-260})$$

$$\mathbf{A}_{k\delta N\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{71}}{T_{corr}} & 0 \\ \frac{k_{81}}{T_{corr}} & 0 \\ \frac{k_{91}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta N\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{77}}{T_{corr}} & 0 \\ \frac{k_{87}}{T_{corr}} & 0 \\ \frac{k_{97}}{T_{corr}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{k\delta t\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,1}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta t\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,7}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta d\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,1}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta d\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,7}}{T_{corr}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{\partial v R} = \begin{bmatrix} \frac{-1}{\tau_{\partial v_{e,R,x}}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\partial v_{e,R,y}}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\partial v_{e,R,z}}} \end{bmatrix}, \quad \mathbf{A}_{\partial v N} = (\mathbf{R}_{en} \cdot \mathbf{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\partial v N} \cdot \mathbf{R}_{en}^T) \quad (\text{III-261})$$

$$\mathbf{B}_{11}^{14 \times 7} = \begin{bmatrix} \mathbf{B}_{sub}^{2 \times 1} & \dots & \mathbf{0}^{2 \times 1} \\ \vdots & \ddots & \vdots \\ \mathbf{0}^{2 \times 1} & \dots & \mathbf{B}_{sub}^{2 \times 1} \end{bmatrix}, \quad \mathbf{B}_{sub}^{2 \times 1} = \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 \end{bmatrix} \quad (\text{III-262})$$

The grey colored term in equation (III-257) $\frac{-1}{T_{corr}} \cdot \mathbf{I}^{3 \times 3}$ is because of the reset mechanism of the velocity error due to a low aiding rate. The reset is carried out only at the aiding rate and not at the filter execution rate.

III - 10.4 Tracking architecture – Aiding as measurement

In the previous section, an aided error state architecture was developed, where the aiding was applied as a control input to an external total state dynamic model. One big disadvantage of this approach is, that there is no estimated platform acceleration available as filter state. The platform acceleration is necessary for estimating aiding delay and tilt aiding errors. Therefore, in this section, a second aided error state vector tracking approach will be derived, where aiding is applied as a measurement to an external total state dynamic filter.

III - 10.4.1 Noise like aiding error and rate aiding error

Modeling and filter integration of these two error types were already analyzed in the previous aided error state vector tracking architecture, where the aiding was applied as a control input.

In this section, a new tracking architecture will be developed, which applies aiding as a measurement. The modeling and tuning of error types in this architecture is different. Therefore, this section derives at first a tracking architecture, considering only these two error types again.

This architecture contains an error state Kalman filter, but additionally a total state Kalman filter, where the velocity aiding is applied as a measurement.

In the following, the dynamic model of the total state Kalman filter is given.

$$\dot{\mathbf{z}}_z = \mathbf{A}_z \cdot \mathbf{z}_z + \mathbf{n}_z$$

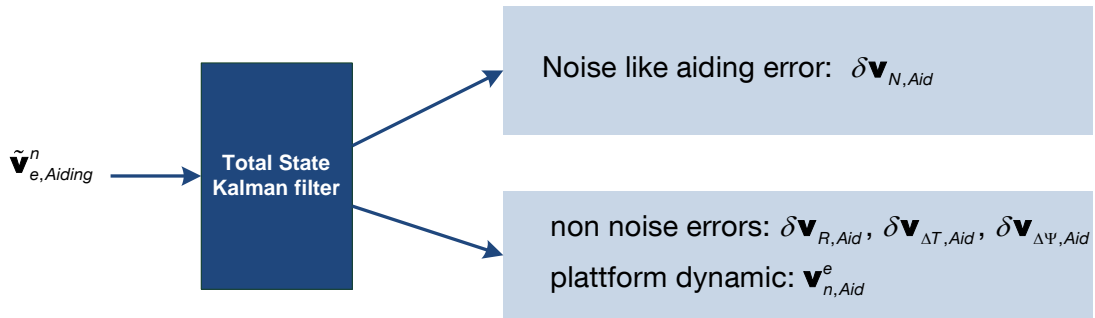
$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e \\ \dot{\hat{\mathbf{a}}}_e \\ \delta \dot{\hat{\mathbf{v}}}_{N,Aid} \\ c \cdot \dot{\hat{t}}_{clk} \\ c \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e \\ \hat{\mathbf{a}}_e \\ \delta \hat{\mathbf{v}}_{N,Aid} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v,N} \\ 0 \\ 0 \end{bmatrix} \quad (III-263)$$

$$\mathbf{A}_a = \begin{bmatrix} \frac{-1}{\tau_{a,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{a,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{a,z}} \end{bmatrix} \quad \mathbf{A}_{\delta v,N}^n = \begin{bmatrix} \frac{-1}{\tau_{\delta vN,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta vN,x}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta vN,x}} \end{bmatrix} \quad (III-264)$$

$$\mathbf{A}_{\delta v,N}^e = (\mathbf{R}_{en} \cdot \mathbf{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N}^n \cdot \mathbf{R}_{en}^T) \quad (III-265)$$

As dynamic model, a 2nd order singer filter is used, whereby the acceleration is modeled as a Gauss Markov process as in previous chapters. The characteristic figures of the Gauss Markov process, which are the correlation time constant and the error variance, are derived using scenario matched tuning.

The velocity aiding contains besides the platform dynamic $\mathbf{v}_{n,Aid}^e$, different types of errors. Basically, the errors can be clustered into two categories, which are noise like aiding errors $\delta \mathbf{v}_{N,Aid}$ and non-noise errors, like aiding delay errors, tilt errors and errors due to a low aiding rate. The task of the total state Kalman filter is, to separate the noise like error from the platform dynamic together with the remaining non noise like errors.



III-168 Total state Kalman filter for error separation

As the following observation matrix of the total state Kalman filter shows, contrary to equation (III-237), the velocity aiding is handled as a measurement.

$$\mathbf{H}_z = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (III-266)$$

Because the noise like aiding error shall only be handled as a consider state, the coupling of this noise like error is only within the observation matrix used for Kalman gain calculation. For calculating the real innovation, the estimated noise like aiding error is not mapped to the measurement space.

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K}_z \cdot \left(\tilde{\mathbf{v}}_{e,Aid}^n - \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \cdot \mathbf{z}_{k+1}^- \right) \quad (\text{III-267})$$

The Kalman filter needs the information, of how to separate the platform dynamic and non-noise like errors from the noise like errors. This information is provided to the Kalman filter by the dynamic model, which models the acceleration as a Gauss Markov process.

Because the noise like aiding error is not white, its variance cannot just be considered in the measurement error variance matrix \mathbf{R}_z of the total state Kalman filter. This noise like measurement error must be modeled as an own Gauss Markov process within the total state dynamic model and coupled into the observation within the observation matrix. Therefore, the total state measurement error covariance matrix gets

$$\mathbf{R}_z = \mathbf{0}_{3 \times 3} \quad (\text{III-268})$$

If the error variance of the noise like aiding error becomes very large, the aiding is used less by the Kalman filter, due to the coupling of the modeled noise like aiding error into the measurement.

Applying aiding according to this architecture prevents bad aiding information from entering the tracking architecture. This is a huge advantage compared to the previous approach, where aiding was applied as a control input, with no mechanism, preventing aiding of bad quality from entering the tracking filter.

In this application, after settling is completed, the total state Kalman filter works as a Wiener filter. Similar Kalman filter architectures can be found in [73]. The platform dynamic is separated from the noise like errors, based on the different spectral characteristics. The noise like errors are modeled as a Gauss Markov process with a short correlation time, which means that the main energy of this process is located at higher frequencies. The platform dynamic itself and the non-noise like errors, are of a far lower dynamic than the noise process. Their main energy is located at lower frequencies. The total state Kalman filter therefore can separate the two components, by realizing a high pass channel from the aiding measurement to the state, representing the noise like error and a low pass channel from the velocity aiding measurement to the estimated total state velocity.

The model error covariance matrix \mathbf{Q}_z is given in the following equation.

$$\mathbf{Q}_z = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v N} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \quad (\text{III-269})$$

$$\mathbf{Q}_a = \begin{bmatrix} \frac{2 \cdot \sigma_{ax}^2}{\tau_{ax}} & 0 & 0 \\ 0 & \frac{2 \cdot \sigma_{ay}^2}{\tau_{ay}} & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{az}^2}{\tau_{az}} \end{bmatrix} \quad (\text{III-270})$$

$$\mathbf{Q}_{\delta vN} = \mathbf{R}_{en} \cdot \begin{bmatrix} \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,d,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,d,N}^n}} \end{bmatrix} \cdot \mathbf{R}_{en}^T \quad (\text{III-271})$$

The estimated position of the total state Kalman filter is used for calculating the NCO command, by using the following nonlinear line of sight transformation.

$$\begin{bmatrix} \Delta \hat{\rho}_{k+1,\#1} \\ \vdots \\ \Delta \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{k+1,\#1} \\ \vdots \\ \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} - \begin{bmatrix} \hat{\rho}_{0,\#1} \\ \vdots \\ \hat{\rho}_{0,\#Nsat} \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{k+1,\#1} \\ \vdots \\ \hat{\rho}_{k+1,\#Nsat} \end{bmatrix} - \begin{bmatrix} \sqrt{(\mathbf{x}_{\#1,0} - \hat{\mathbf{x}}_{e,0})^T \cdot (\mathbf{x}_{\#1,0} - \hat{\mathbf{x}}_{e,0})} \\ \vdots \\ \sqrt{(\mathbf{x}_{\#Nsat,0} - \hat{\mathbf{x}}_{e,0})^T \cdot (\mathbf{x}_{\#Nsat,0} - \hat{\mathbf{x}}_{e,0} - \hat{\mathbf{x}}_{e,k+1})} \end{bmatrix} \quad (\text{III-272})$$

Besides the total state Kalman filter, an error state Kalman filter is used to process the discriminator measurements and to estimate GPS errors and aiding errors.

The discriminator measurements contain the following error components

$$\begin{array}{l} \text{GPS system errors:} \\ \text{receiver clock error: } \mathbf{t}_{clk} \\ \text{satellite clock error: } \mathbf{t}_{clk, \#Sv} \\ \text{Ionosphere-, Troposphere- and Ephemeris Error} \\ \text{Discriminator noise: } n_{\delta r} \end{array} \quad (\text{III-273})$$

$$\begin{array}{l} \text{Remaining aiding errors:} \\ \text{Low aiding rate error: } \delta \mathbf{v}_{R,Aid} \\ \text{Out of sequence error: } \delta \mathbf{v}_{\Delta T, Aid} \\ \text{Tilt error: } \delta \mathbf{v}_{\Delta \Psi, Aid} \\ \text{Long correlated remaining total state error: } \delta \mathbf{x}_{TS} \end{array} \quad (\text{III-274})$$

The long-correlated error $\delta \mathbf{x}_{TS}$ is an effect of the total state Kalman filter. Because the total state Kalman filter separates the noise like aiding error from the real platform dynamic based on a singer filter like dynamic model and a model for the noise characteristics, the separation is not perfect. The dynamic, which is mapped from the total state Kalman filter on the positioning state, is not

the full dynamic and has still some small parts of noise like aiding errors. These errors are highly correlated. Because the positioning state is used to generate the NCO command, also the long-correlated errors are included in the NCO command and therefore also included in the discriminator measurements. It is necessary to consider these long-correlated aiding errors within the error state dynamic model.

The following equation shows the error state Kalman filter dynamic model.

$$\delta \dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta \hat{\mathbf{z}} + \mathbf{n}_{\delta z}$$

$$\begin{bmatrix} \delta \hat{\mathbf{x}}_e \\ \delta \hat{\mathbf{v}}_{e,R,Aid}^e \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R,Aid} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \hat{\mathbf{x}}_e \\ \delta \hat{\mathbf{v}}_{e,R,Aid}^e \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_x \\ \mathbf{n}_{\delta v,R,Aid} \\ \mathbf{n}_{\delta tclk} \\ \mathbf{n}_{\delta dclk} \end{bmatrix} \quad (\text{III-275})$$

The long-correlated errors are modeled as a Gauss Markov process by the matrix \mathbf{A}_{xTS} .

$$\mathbf{A}_x = \begin{bmatrix} \frac{-1}{\tau_{xTS}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{yTS}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{zTS}} \end{bmatrix} \quad (\text{III-276})$$

The variances $\sigma_{x,TS}^2$, $\sigma_{y,TS}^2$ and $\sigma_{z,TS}^2$ are taken out of the total state error covariance matrix \mathbf{P}_{TS} . The correlation time constants must be selected based on measurements, which can also be done in real time.

The error caused by the low aiding rate $\delta \hat{\mathbf{v}}_{e,R,Aid}^e$, is only observable within the error state Kalman filter, where the code phase errors between the received code phase and commanded code phase are available within the discriminator measurement. A high rate aiding error increases the uncertainty of the estimated positioning error $\delta \hat{\mathbf{x}}_e$. As a result, the Kalman filter puts more weight on the discriminator measurement, which is the right behavior in order to compensate rate aiding errors.

Within the total state Kalman filter, the singer filter dynamic model has not the needed constraint to extract this error type from the aiding velocity.

The rate aiding error is again modeled as a Gauss Markov process, similar to the previous section.

The error $\delta \hat{\mathbf{v}}_{e,R,Aid}^e$ is coupled with the positioning error. This rate aiding error is nevertheless handled as a consider state, because the real state propagation is not realized. Every time after the error state update equation is carried out, the error states are stored within the total states by the following equation and afterwards the error states are set to zero. (The estimated rate aiding error is not used to update the total state velocity).

$$\begin{bmatrix} \hat{\mathbf{x}}_e \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_e \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \delta \hat{\mathbf{x}}_e \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \end{bmatrix} \quad (\text{III-277})$$

The error state Kalman filter update equation gets:

$$\delta \mathbf{z}_{k+1}^+ = \delta \mathbf{z}_{k+1}^- + \mathbf{K}_{\delta z} \cdot \begin{bmatrix} \delta \tau_{\#1} \\ \vdots \\ \delta \tau_{\#Nsat} \end{bmatrix} \quad (\text{III-278})$$

As in previous architecture, also here the innovation is calculated within the correlation core.

The corresponding linearized observation matrix gets

$$\mathbf{H}_{\delta z} = \begin{bmatrix} \left(\frac{\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e}{c \cdot \|\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e}{c \cdot \|\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \end{bmatrix} \quad (\text{III-279})$$

This "storing" or correction of the total states, is mathematical equivalent with a perfect measurement within the total state Kalman filter update equation as written below.

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K}_z \cdot \begin{pmatrix} \delta \hat{\mathbf{x}}_e \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \\ \tilde{\mathbf{v}}_{e,Aid}^n - [\mathbf{0}_{3 \times 3} \quad \mathbf{R}_{ne} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 1} \quad \mathbf{0}_{3 \times 1}] \cdot \mathbf{z}_{TS,k+1}^- \end{pmatrix} \quad (\text{III-280})$$

The observation matrix belonging to the update equation (III-280) gets

$$\mathbf{H}_z = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 1 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (\text{III-281})$$

This perfect measurement can also be interpreted as a constrained Kalman filter implementation as given in [39, p. 212].

The problem of this implementation are the zero entries within the measurement error covariance matrix \mathbf{R}_z , which can lead to instabilities. Therefore, the better approach is a direct correction of the total states, outside the total state Kalman filter.

In case of updating the total states by the estimated error states according to equation (III-277), outside the Kalman filter equations, the total state update equation and observation matrix is given by the following equation.

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K}_z \cdot \left(\tilde{\mathbf{v}}_{e,Aid}^n - \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \cdot \mathbf{z}_{k+1}^- \right) \quad (\text{III-282})$$

$$\mathbf{H}_z = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (\text{III-283})$$

Compared to the previous section, where aiding was applied to the total state dynamic as a control input, the architecture developed in this chapter has the essential advantage, that despite only a velocity aiding is available, also the acceleration is estimated.

As will be seen in the next section, where additional aiding errors like delay errors or tilt errors are included, the actual acceleration is a vital figure for estimating these error components.

The combination of a total state Kalman filter, separating noise like errors from the vehicle dynamic and other non-noise like errors, together with an error state Kalman filter has additionally the advantage, that estimation of the acceleration is independent of the discriminator noise.

Regarding the acceleration, the quality of the estimated acceleration deteriorates in case of low aiding rates.

The total state positioning error variances σ_x^2 , σ_y^2 and σ_z^2 are diverging, because only the aiding velocity is as measurement available and the position is not stabilized. The stabilization takes place by the state correction outside the total state Kalman filter, which is not considered in the total state error covariance propagation. Therefore, for the error covariance prediction of the positioning solution, respectively the GDOP value, the entries of the state error covariance matrix $\mathbf{P}_{\delta\mathbf{z}}$ of the error state Kalman filter are used. (If the storing of the error states within the total states would be implemented according to equation (III-280), the total state positioning error variances would also be stable.).

The model error covariance matrix of the error state filter is given below.

$$\mathbf{Q}_{\delta\mathbf{z}} = \begin{bmatrix} \mathbf{Q}_x & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta vR} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \sigma_{\delta clk}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & \sigma_{\delta dclk}^2 \end{bmatrix} \quad (\text{III-284})$$

$$\mathbf{Q}_{\delta vR} = \begin{bmatrix} \frac{\sigma_{\delta v_{e,x,R}}^2 \cdot 2}{\tau_{\delta v_{e,x,R}}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,y,R}}^2 \cdot 2}{\tau_{\delta v_{e,y,R}}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,z,R}}^2 \cdot 2}{\tau_{\delta v_{e,z,R}}} \end{bmatrix} \quad (\text{III-285})$$

$$\mathbf{Q}_x = \begin{bmatrix} \frac{\sigma_{x,ex}^2 \cdot 2}{\tau_{x,ex}} & 0 & 0 \\ \tau_{x,ex} & & \\ 0 & \frac{\sigma_{x,ey}^2 \cdot 2}{\tau_{x,ey}} & 0 \\ 0 & 0 & \frac{\sigma_{x,eZ}^2 \cdot 2}{\tau_{x,eZ}} \end{bmatrix}$$

The following equations show the explicit realization of the discrete filter execution.

#1 total state propagation: $\mathbf{z}_{k+1}^- = (\mathbf{I}^{14 \times 14} + \Delta T \cdot \mathbf{A}_z) \cdot \mathbf{z}_k^+$
 $\mathbf{P}_{z,k+1}^- = \Phi_z \cdot \mathbf{P}_{z,k}^+ \cdot \Phi_z^T + \mathbf{Q}_z$

#2 total state Kalman gain calculation: $\mathbf{K}_z = (\mathbf{P}_z \cdot \mathbf{H}_z^T) \cdot (\mathbf{H}_z \cdot \mathbf{P}_z \cdot \mathbf{H}_z^T + \mathbf{R}_z)^{-1}$

#3 update of total states:

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K}_z \cdot (\tilde{\mathbf{v}}_{e,Aid}^n - [\mathbf{0}_{3 \times 3} \quad \mathbf{R}_{ne} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 1} \quad \mathbf{0}_{3 \times 1}] \cdot \mathbf{z}_{k+1}^-)$$

#4 update total state error covariance: $\mathbf{P}_{z,k+1}^+ = (\mathbf{I}^{14 \times 14} - \mathbf{K}_z \cdot \mathbf{H}_z) \cdot \mathbf{P}_{z,k+1}^-$

#5 NCO command application:
$$\begin{bmatrix} \Delta \hat{\rho}_{k+1, \#1} \\ \vdots \\ \Delta \hat{\rho}_{k+1, \#N_{sat}} \end{bmatrix}$$

#6 error state propagation: $\delta \mathbf{z}_{k+1}^- = \Phi_{\delta z} \cdot \delta \mathbf{z}_k^+$

#7 error state covariance propagation: $\mathbf{P}_{\delta z} = \Phi_{\delta z} \cdot \mathbf{P}_{\delta z} \cdot \Phi_{\delta z}^T + \mathbf{Q}_{\delta z}$

#8 error state Kalman gain calculation: $\mathbf{K}_{\delta z} = (\mathbf{P}_{\delta z} \cdot \mathbf{H}_{\delta z}^T) \cdot (\mathbf{H}_{\delta z} \cdot \mathbf{P}_{\delta z} \cdot \mathbf{H}_{\delta z}^T + \mathbf{R}_{\delta z})^{-1}$

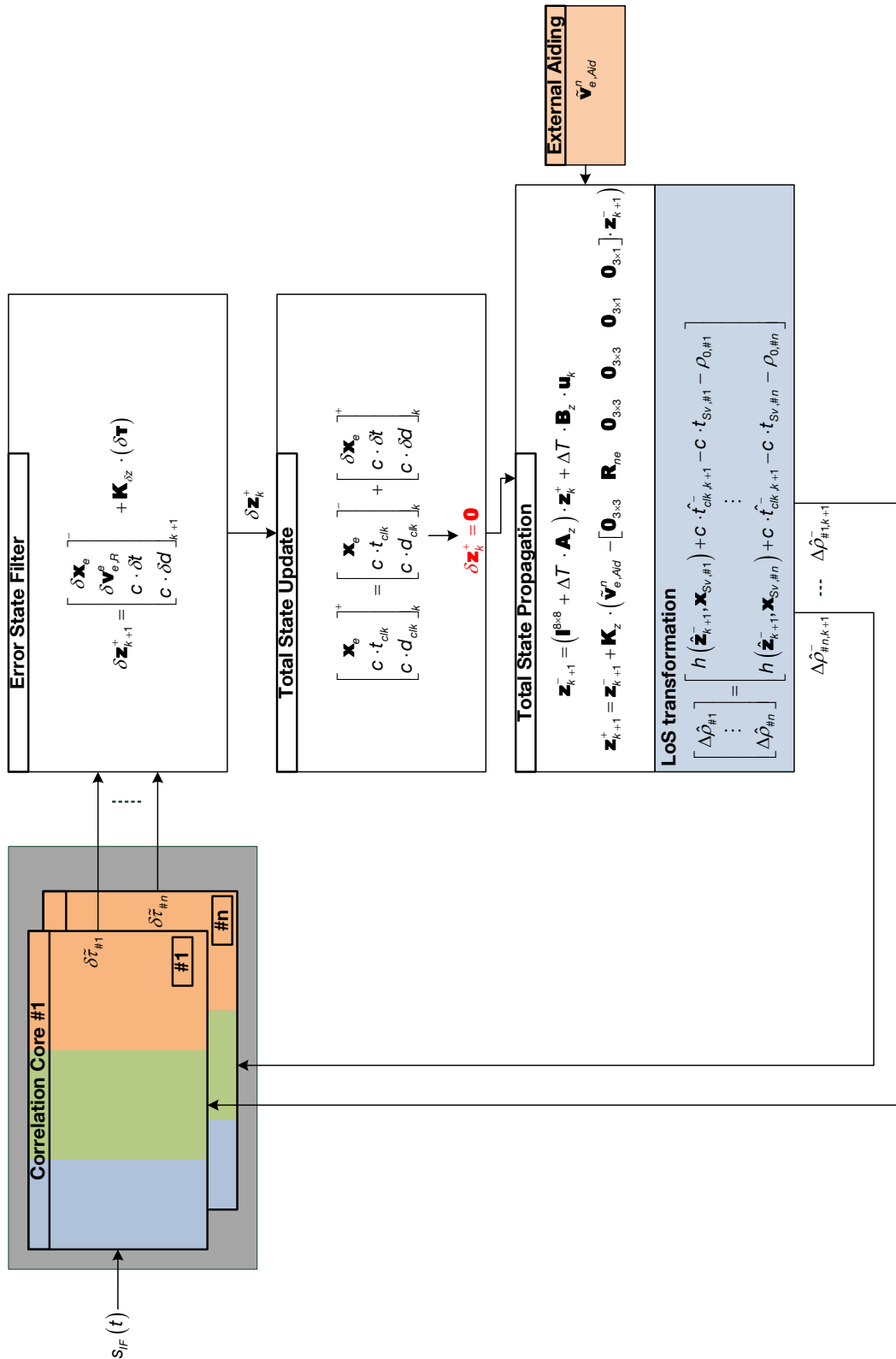
#9 update error state error covariance: $\mathbf{P}_{\delta z,k+1}^+ = (\mathbf{I}^{14 \times 14} - \mathbf{K}_{\delta z} \cdot \mathbf{H}_{\delta z}) \cdot \mathbf{P}_{\delta z,k+1}^-$

#9 error state update: $\delta \mathbf{z}_{k+1}^+ = \delta \mathbf{z}_{k+1}^- + \mathbf{K}_{\delta z} \cdot (\delta \mathbf{T})$

#10 total state correction:
$$\begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix}_{k+1}^+ = \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot t_{clk} \\ \mathbf{c} \cdot d_{clk} \end{bmatrix}_{k+1}^+ + \begin{bmatrix} \delta \mathbf{x}_e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix}_{k+1}^+$$

#11 error state reset:
$$\begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R}^e \\ \mathbf{c} \cdot \delta t \\ \mathbf{c} \cdot \delta d \end{bmatrix}_k^+ = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix}$$

The following figure shows the introduced tracking architecture.



III-169 Aided error state vector tracking - aiding as measurement

Problems of this architecture

In contrast to the previous architecture, where aiding was used as a control input, here it is possible to prevent aiding with high noise like aiding errors from entering the tracking loop, because in case of high aiding noise, the total state Kalman filter puts more weight on its own dynamic model.

The challenge of this architecture is the coupling of an error state and total state Kalman filter. Some of the total states are measurements of the error state filter, via the feedback path through the NCO command and discriminator measurement. The errors of the total states are correlated, which need to be considered in the design of the error state filter. Moreover, the aiding error estimation needs to be separated about the two filters.

It is for example necessary, to consider the rate aiding error within the error state Kalman filter and not in the total state Kalman filter, because given low aiding rate and therefore high rate aiding errors, it is important to put more weight on the discriminator measurement to compensate the low aiding rate. Exactly this happens if the rate aiding error is considered in the error states.

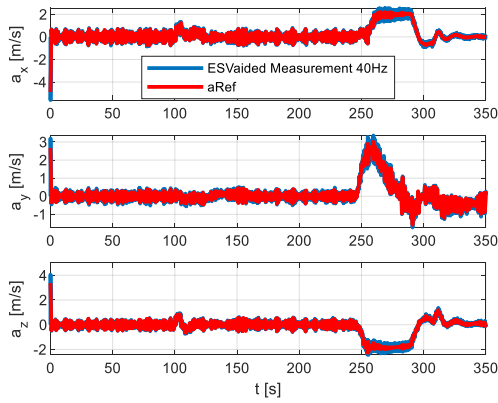
If the rate aiding error would be considered in the total state filter, the filter puts more weight on its own dynamic model in case of low aiding rates. This would be indeed the correct behavior, but this does not force the error state Kalman filter to put more weight on the discriminator measurement in order to compensate the error caused by low aiding rates.

Due to the two Kalman filter, there are two error state covariance matrices, $\mathbf{P}_{\delta\mathbf{z}}$ and $\mathbf{P}_{\mathbf{z}}$. Valid for the estimated position error variance is $\mathbf{P}_{\delta\mathbf{z}}$, because the total state errors are considered in the error state Kalman filter through $\mathbf{A}_{\mathbf{x}}$ and $\mathbf{Q}_{\mathbf{x}}$.

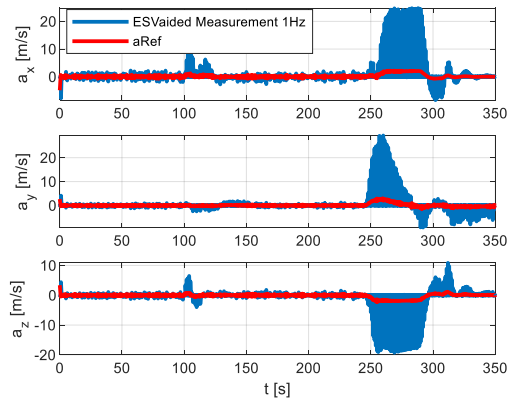
Estimation of acceleration

One advantage of this aided error state approach is the capability to estimate the platform acceleration. The acceleration is vital for estimating some specific aiding errors, as will be shown in the following sections.

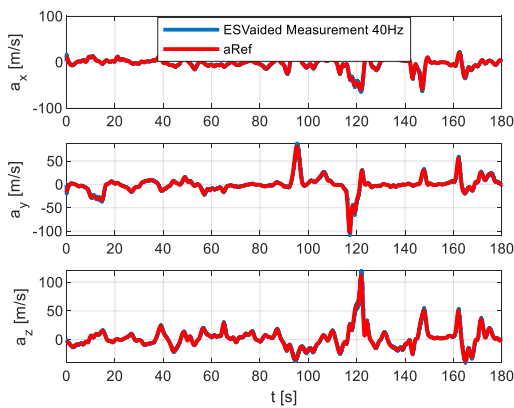
The following figures compare the real acceleration with the estimated acceleration for low and high aiding rates.



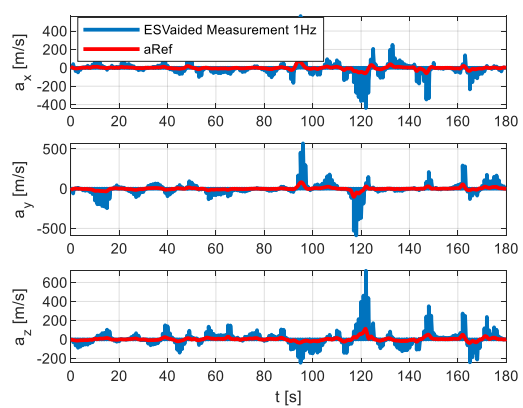
III-170 estimation of acceleration (DA42 40Hz aiding)



III-171 estimation of acceleration (DA42 1Hz aiding)



III-172 estimation of acceleration (High dynamic 40Hz aiding)



III-173 estimation of acceleration (High dynamic 1Hz aiding)

The simulations above show, that the platform acceleration can be estimated, given an appropriate aiding rate. An aiding rate of only 1Hz is not enough, estimating the platform acceleration.

The simulations above are unjammed scenarios. But even in case of jammed scenarios, the estimated acceleration is equal. Jamming has no influence on the quality of the estimated acceleration, because discriminator error variance is not coupled in some way with the estimated total state acceleration.

III - 10.5 Equivalent space band dynamic state space simulation – aiding as measurement

III - 10.5.1 Distributed state space realization

The distributed state space representation is like the architecture given in figure III-169. The Kalman filter realization is the same. The only difference is the correlation core formulation in state space form, as already given in figure III-169.

III - 10.5.2 Centralized state space realization

For calculating the closed loop tracking bandwidth of selected tracking channels, in the following, the centralized state space realization for the error state vector tracking form is given, whereby the aiding information is applied as measurement.

$$\mathbf{A}_{\delta\tau\#1}^{2 \times 2} = \dots = \mathbf{A}_{\delta\tau\#1}^{2 \times 2} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \quad (\text{III-288})$$

$$\mathbf{A}_{h\#1}^{2 \times 3} = \begin{bmatrix} \frac{\partial(h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e) - h(\mathbf{x}_{Sv\#1,0}, \hat{\mathbf{x}}_{e,0}))}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \dots$$

$$\begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{\partial \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)}}{\partial \hat{\mathbf{x}}_e} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}_{\hat{\mathbf{x}}_{e,k}} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{h\#N}^{2 \times 3} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#7,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#7,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{k\delta x\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{11}}{T_{corr}} & 0 \\ \frac{k_{21}}{T_{corr}} & 0 \\ \frac{k_{31}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta x\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{1,7}}{T_{corr}} & 0 \\ \frac{k_{2,7}}{T_{corr}} & 0 \\ \frac{k_{3,7}}{T_{corr}} & 0 \end{bmatrix} \quad \mathbf{A}_{k\delta vR\#1}^{3 \times 2} = \begin{bmatrix} \frac{k_{41}}{T_{corr}} & 0 \\ \frac{k_{51}}{T_{corr}} & 0 \\ \frac{k_{61}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta vR\#N}^{3 \times 2} = \begin{bmatrix} \frac{k_{47}}{T_{corr}} & 0 \\ \frac{k_{57}}{T_{corr}} & 0 \\ \frac{k_{67}}{T_{corr}} & 0 \end{bmatrix} \quad (\text{III-290})$$

$$\mathbf{A}_{k\delta t\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,1}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta t\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{10,7}}{T_{corr}} & 0 \end{bmatrix}, \quad \mathbf{A}_{k\delta d\#1}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,1}}{T_{corr}} & 0 \end{bmatrix} \dots \mathbf{A}_{k\delta d\#N}^{1 \times 2} = \begin{bmatrix} \frac{k_{11,7}}{T_{corr}} & 0 \end{bmatrix}$$

$$\mathbf{A}_{\delta vR} = \begin{bmatrix} \frac{-1}{\tau_{\delta v_{e,R,x}^e}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v_{e,R,y}^e}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v_{e,R,z}^e}} \end{bmatrix}, \quad \mathbf{A}_{\delta vN} = (\mathbf{R}_{en} \cdot \mathbf{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N} \cdot \mathbf{R}_{en}^T) \quad (\text{III-291})$$

$$\mathbf{A}_x = \begin{bmatrix} \frac{-1}{\tau_{xTS}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{yTS}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{zTS}} \end{bmatrix}, \quad \mathbf{A}_a = \begin{bmatrix} \frac{-1}{\tau_{ax}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{ay}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{az}} \end{bmatrix} \quad (\text{III-292})$$

$$\mathbf{A}_{Kx}^{3 \times 3} = \begin{bmatrix} \frac{K_{11}}{T_{corr}} & \frac{K_{12}}{T_{corr}} & \frac{K_{13}}{T_{corr}} \\ \frac{K_{14}}{T_{corr}} & \frac{K_{15}}{T_{corr}} & \frac{K_{16}}{T_{corr}} \\ \frac{K_{17}}{T_{corr}} & \frac{K_{18}}{T_{corr}} & \frac{K_{19}}{T_{corr}} \end{bmatrix}, \quad \mathbf{A}_{Kv}^{3 \times 3} \dots \text{ have a similar structure} \quad (\text{III-293})$$

All Kalman gains within the total state matrices are taken from \mathbf{K}_z and all Kalman gains in the error state matrices are taken from $\mathbf{K}_{\delta z}$.

The orange colored entries in the centralized state space matrix, are for realizing the error state reset after update. The green entries are realizing the total state correction by selected error states.

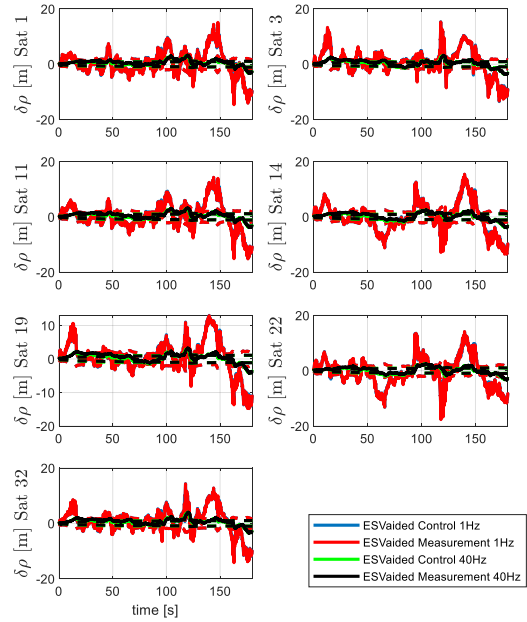
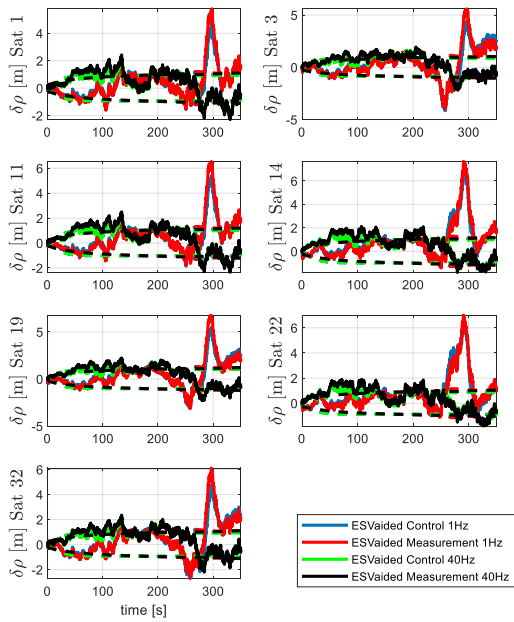
III - 10.6 Aiding as control input vs. aiding as measurement

In the previous two sections, two different approaches for an aided error state vector tracking architecture were developed. The first one applies aiding as a control input to a total state dynamic propagation. The second approach applies aiding as a measurement to a separate total state Kalman filter.

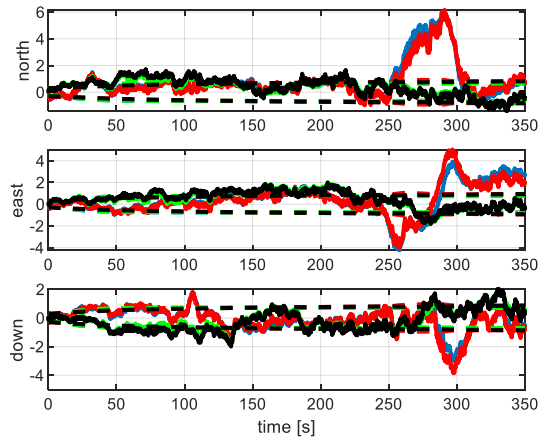
In this section, these two different types of aided error state vector tracking architectures will be compared.

The aiding rate is a vital parameter, defining the tracking performance. Therefore, in the following, both aided error state architectures will be compared for an aiding rate of 1Hz and an aiding rate of 40 Hz, also in unjammed and jammed scenarios.

UNJAMMED

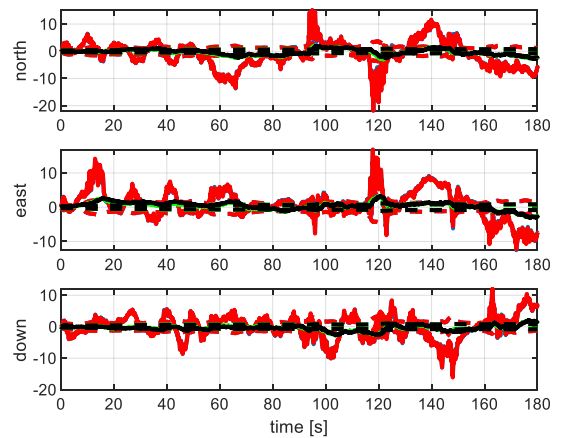


III-174 pseudorange error - DA42 - unjammed

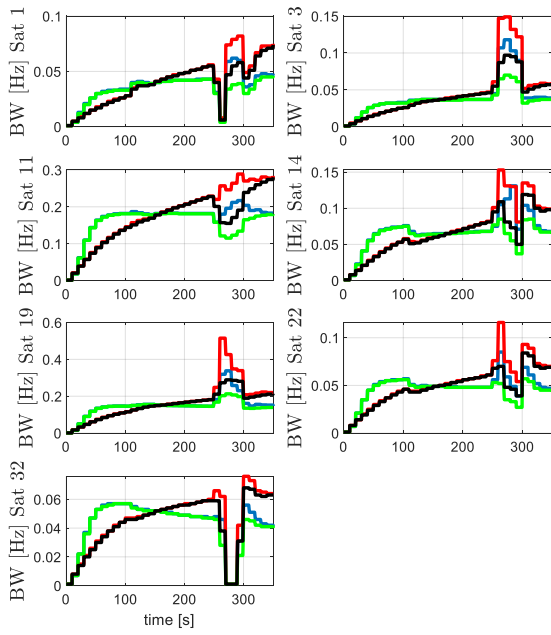


III-176 positioning error - DA42 - unjammed

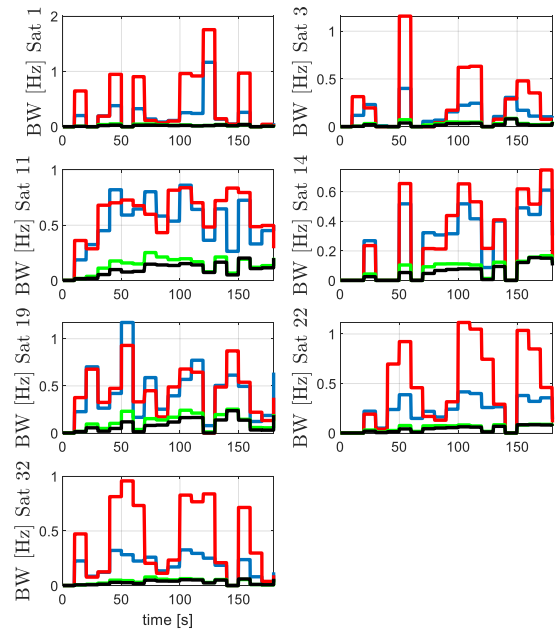
III-175 pseudorange error - high dynamic - unjammed



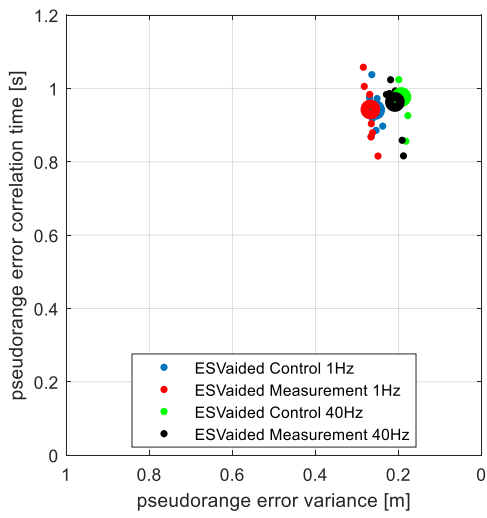
III-177 positioning error - high dynamic - unjammed



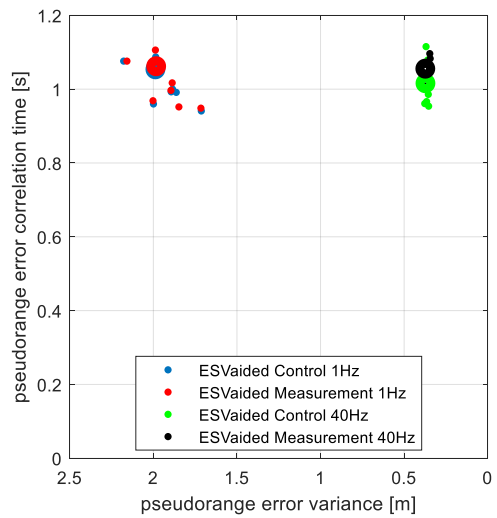
III-178 bandwidth - DA42 - unjammed



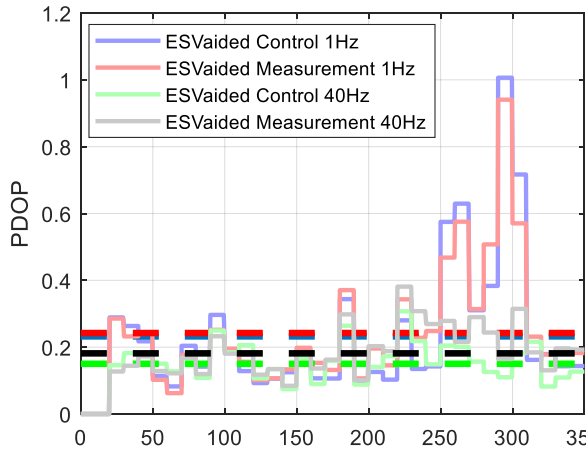
III-179 bandwidth - high dynamic - unjammed



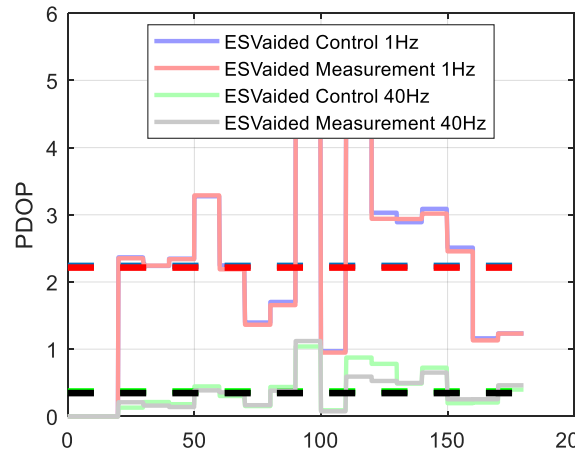
III-180 Comparing raw date - DA42 - unjammed



III-181 Comparing raw date - high dynamic - unjammed

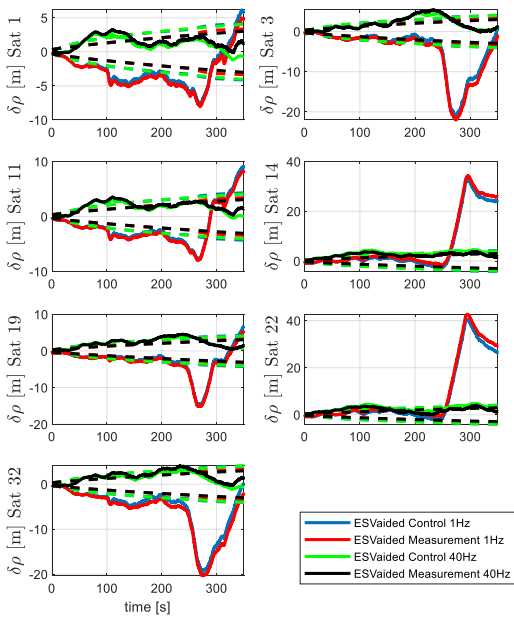


III-182 PDOP - DA42 - unjammed

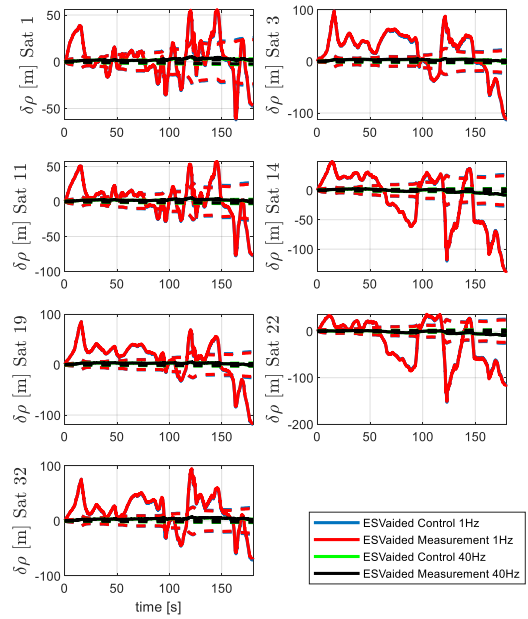


III-183 PDOP - high dynamic - unjammed

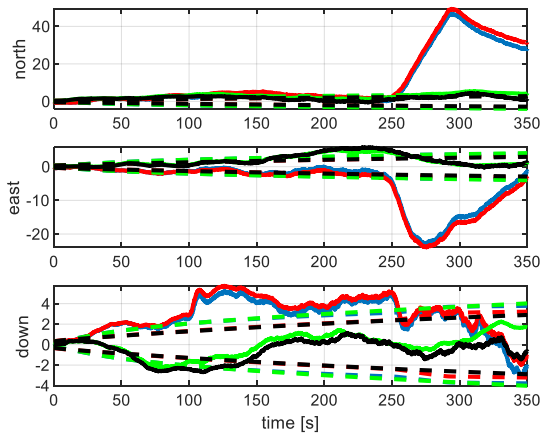
JAMMED



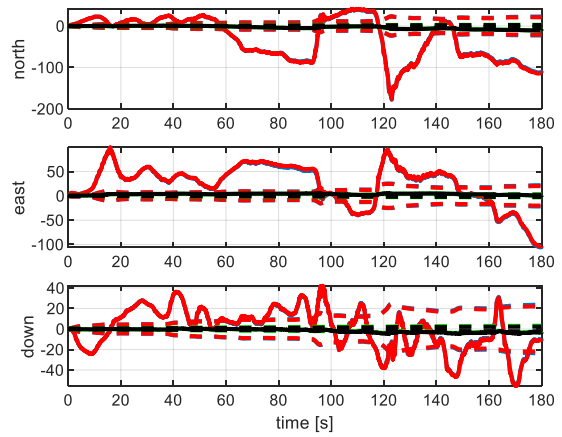
III-184 pseudorange error - DA42 - jammed



III-185 pseudorange error - high dynamic - jammed



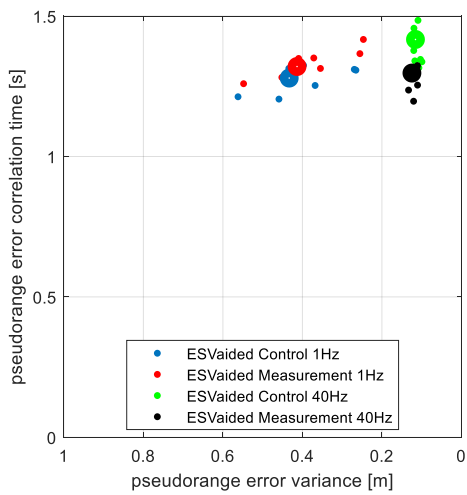
III-186 positioning error - DA42 - jammed



III-187 positioning error - high dynamic - jammed

Bandwidth is almost 0 Hz

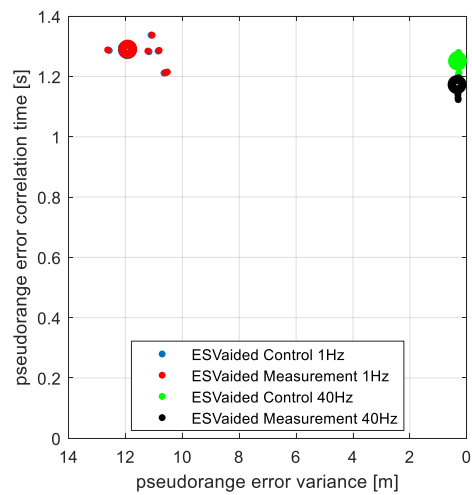
III-188 bandwidth - DA42



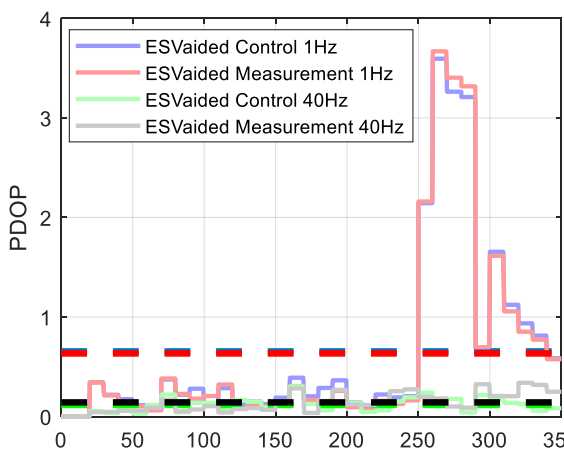
III-190 Comparing raw date – DA42 - jammed

Bandwidth is almost 0 Hz

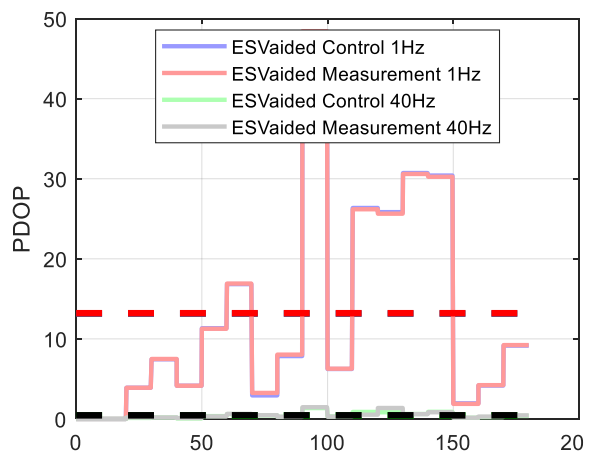
III-189 bandwidth - high dynamic



III-191 Comparing raw date - high dynamic - jammed



III-192 PDOP - DA42 - jammed



III-193 PDOP - high dynamic - jammed

CONCLUSION: Comparing “Aided Error State Vector Tracking – 1 Hz Aiding Rate” with “Aided Error State Vector Tracking – 40 Hz Aiding Rate”

The simulations above show that using aiding as a control input or aiding as a measurement gives almost equal tracking results. Therefore, it makes absolutely sense, using the architecture where aiding is used as a measurement, because in this case, also the acceleration can be estimated.

Using an appropriate aiding rate is important for good tracking results. As the simulations show, an aiding rate of 1Hz is insufficient, whereby 40Hz aiding gives even in jammed scenarios, good results.

Aiding as control input		Aiding as measurement	
-	Aiding of bad quality enters the tracking filter and worsens the overall performance. There is no filter mechanism to prevent bad aiding information.	+	Because aiding is applied as measurement, low quality aiding information are discarded.
+	Only one Kalman filter. No difficult coupling between the two filters	-	Coupling of two Kalman filters makes tuning more complicated and sensitive.
		+	Estimation of platform acceleration possible. This acceleration is vital for estimating tilt aiding errors and aiding delay.
		+	Estimation of platform acceleration is independent of actual signal to noise ratio

In the following sections on comparing different tracking architectures, only the approach using aiding as measurement is used, due to its capability of estimating the platform acceleration.

III - 10.7 Estimation of aiding delay

In the previous section, only noise like aiding errors together with a low aiding rate were considered. In this section, delay aiding errors will be introduced. As aiding rate, 40 Hz is used.

III - 10.7.1 Aiding delay

There exist different methods for considering delayed aiding in the innovation within the update step of a Kalman filter. Often used methods are given in [74] and [75]. These methods are compared in [33, p. 182]. An additional method is given by Larsen [76], which represents an suboptimal method with the basic idea, that the innovation at time $k - \frac{\Delta t_{OOS}}{\Delta T} = k - j$,

$$\mathbf{i}_{k-j} = \mathbf{y}_{k,k-j} - \mathbf{H} \cdot \mathbf{x}_{k-j} \quad (\text{III-294})$$

shall be the same as the innovation

$$\mathbf{i}_k = \mathbf{y}_k - \mathbf{H} \cdot \mathbf{x}_k \quad (\text{III-295})$$

with \mathbf{y}_k being an extrapolated measurement value. The term Δt_{OOS} represents the aiding delay. The measurement $\mathbf{y}_{k,k-j}$ is received at time step k, but valid for the time step k-j, being j samples in the past.

A slightly modified version of [76] is used in [75, p. 234].

But all these examples of dealing with delayed measurements, cannot be applied to Kalman filter-based tracking. Here, the aiding is not a direct measurement of the Kalman filter. As measurements for the Kalman filter, only the discriminator measurements are used. These discriminator measurements always show the phase error between the received code phase and replicated code phase at the valid time. The discriminator measurement is not “delayed”. But the aiding delay error is part of the discriminator measurement errors.

Therefore, the delay aiding error is at the measurement time of the discriminator measurement an in-time or in-sequence error.

This delayed aiding velocity error can be approximated in discrete time space as

$$\delta \hat{\mathbf{v}}_{e,\text{OOS},k}^e = \hat{\mathbf{a}}_{e,k}^{ee} \cdot \Delta \hat{T}_{\text{OOS}} \quad (\text{III-296})$$

Because the aiding delay $\Delta \hat{T}_{\text{OOS}}$ is a constant value, it makes sense to estimate this error in the error state Kalman filter. The estimation must be implemented within the error state Kalman filter, because only the discriminator measurements give the necessary constraints for observing the aiding delay.

From equation (III-296) it gets obvious, that for observing the aiding delay by measuring the velocity error, the actual platform acceleration is necessary. At this point it becomes obvious, why the aided error state vector tracking architecture, using the aiding as a measurement in an additional total state Kalman filter, makes sense. This architecture can estimate the actual line of sight acceleration.

III - 10.7.2 Filter architecture with consideration of delayed aiding

The error state dynamic model in this section is based on the error state dynamic model, derived in equation (III-275) . This model is expanded about the aiding delay error $\delta \Delta \hat{T}_{\text{OOS}}$. The aiding delay is modeled as a constant value.

$$\delta \dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta \hat{\mathbf{z}} + \mathbf{n}_{\delta z}$$

$$\begin{bmatrix} \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_{e,R,Aid}^p \\ \delta \Delta \dot{T}_{OOS} \\ c \cdot \delta \dot{t}_{clk} \\ c \cdot \delta \dot{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{xTS} & \mathbf{I}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R,Aid} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R,Aid}^e \\ \delta \Delta \hat{T}_{OOS} \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{x,TS} \\ \mathbf{n}_{\delta v,R,Aid} \\ n_{\Delta T,OOS} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (III-297)$$

$$\mathbf{A}_a = \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix}$$

For the Kalman filter being able to estimate the aiding delay, some uncertainty must be added to the dynamic model. This is realized by some white noise error belonging to the state $\delta \Delta \hat{T}_{OOS}$. (If no error variance would be realized, the Kalman filter assumes the model to be exact and there would be no necessity to correct the delayed aiding by the measurement within the Kalman filter update equation.

As measurement only the pseudorange errors at the output of the discriminators are available. In order to make this aiding delay error observable, it must be coupled with the positioning state. This is realized by multiplication with the actual total state acceleration estimation according to equation (III-296).

The error state observation matrix is given as

$$\mathbf{H}_{ES} = \begin{bmatrix} \left(\frac{\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e}{c \cdot \|\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & 0 & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e}{c \cdot \|\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & 0 & \frac{1}{c} & 0 \end{bmatrix} \quad (III-298)$$

As total state dynamic model, the already derived model from equation (III-263) is expanded about the estimated aiding delay.

$$\dot{\mathbf{z}}_{TS} = \mathbf{A}_{TS} \cdot \mathbf{z}_{TS} + \mathbf{n}_{TS}$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ \delta \dot{\hat{\mathbf{v}}}_{N,Aid} \\ \Delta \dot{\hat{T}}_{OOS} \\ \mathbf{c} \cdot \dot{\hat{t}}_{clk} \\ \mathbf{c} \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \hat{\mathbf{a}}_e^{ee} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta ve,N} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{N,Aid} \\ \Delta \hat{T}_{OOS} \\ \mathbf{c} \cdot \hat{t}_{clk} \\ \mathbf{c} \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v,N} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (III-299)$$

$$\mathbf{A}_a = \begin{bmatrix} \frac{-1}{\tau_{a,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{a,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{a,z}} \end{bmatrix} \quad \mathbf{A}_{\delta ve,N} = \begin{bmatrix} \frac{-1}{\tau_{\delta ve,N,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta ve,N,x}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta ve,N,x}} \end{bmatrix} \quad (III-300)$$

The resulting positioning error caused by the delayed aiding velocity is the integral of the following velocity error.

$$\Delta \mathbf{x}_{OOS}(t) = \int_{t_0}^t \hat{\mathbf{a}}_{e,k}^{ee}(t) \cdot \Delta \hat{T}_{OOS} dt \quad (III-301)$$

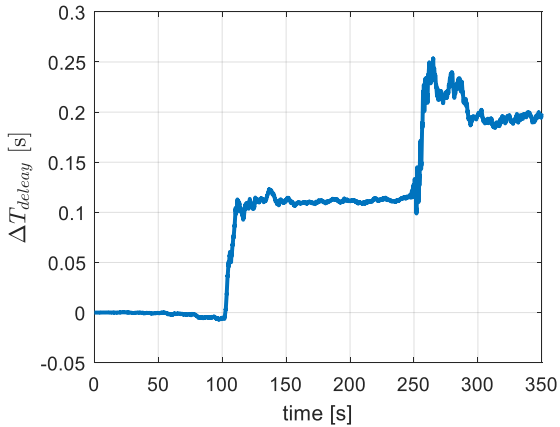
For compensation, it is therefore necessary to couple the estimated aiding delay into the positioning error differential equation for integration. A disadvantage is the resulting nonlinear total state dynamic model through the multiplication of $\hat{\mathbf{a}}_e^{ee} \cdot \Delta \hat{T}_{OOS}$.

The correction of the total states by the error states is given below.

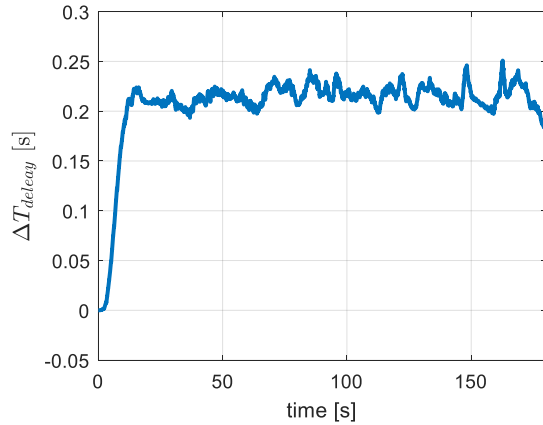
$$\begin{bmatrix} \hat{\mathbf{x}}_e \\ \Delta \hat{T}_{OOS} \\ \mathbf{c} \cdot \hat{t}_{clk} \\ \mathbf{c} \cdot \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_e \\ \Delta \hat{T}_{OOS} \\ \mathbf{c} \cdot \hat{t}_{clk} \\ \mathbf{c} \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \delta \hat{\mathbf{x}}_e \\ \delta \Delta \hat{T}_{OOS} \\ \mathbf{c} \cdot \delta \hat{t}_{clk} \\ \mathbf{c} \cdot \delta \hat{d}_{clk} \end{bmatrix} \quad (III-302)$$

In the previous sections on aided error state vector tracking architecture, no aiding delay error was stimulated for simulations. In the following examples, an aiding delay with $\Delta \hat{T}_{OOS} = 0.2s$ is stimulated.

The following figures show the aiding delay estimation.



III-194 Estimated aiding delay – DA42

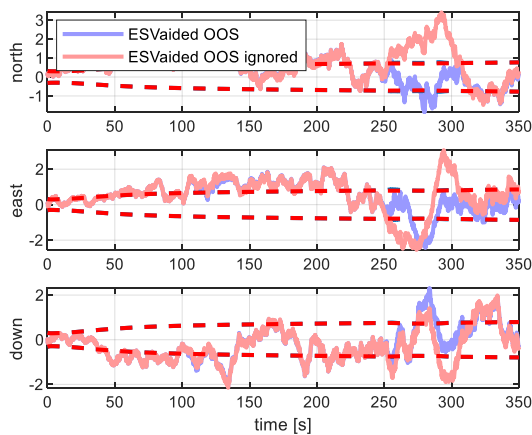


III-195 Estimated aiding delay - Fighter

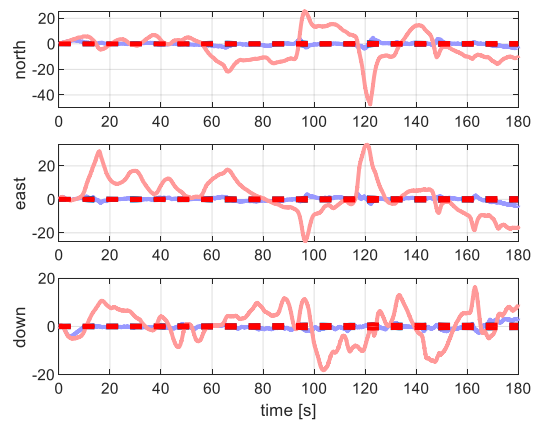
The developed tracking architecture can estimate the aiding delay. As equation (III-296) shows, the aiding delay is only observable, if the platform acceleration is not equal to zero. This can be observed very well from the estimated aiding delay in case of the DA42 scenario. Only after some dynamic, the delay is estimated properly.

The tuning of the fighter dynamic need to be modified in order to get appropriate values. Due to modeling the aiding delay as a constant value with additional white noise, the Kalman filter maps some clock error on to this aiding delay. The error variance for clock error must be reduced in case of high dynamic scenarios.

In order to show the necessity of estimating the aiding delay, the following simulations show the results, if the aided error state vector tracking architecture does not care about the aiding delay.



III-196 ESV without consideration of aiding delay - DA42



III-197 ESV without consideration of aiding delay - Fighter

In case of low dynamic trajectories, the delayed aiding causes only small positioning errors, but in high dynamic situation, figure III-197 shows that it's mandatory to estimate the aiding delay.

III - 10.8 Tilt error estimation

In this section, an additional aiding error will be introduced – the tilt aiding error. Typically, the aiding is provided in NES coordinates, \mathbf{v}_e^e . Tilt aiding errors occur, if GPS receiver expects aiding in NES coordinates, but the provided aiding is in a tilted n-frame, $\mathbf{V}_e^{\hat{n}}$. This is the case, if in the external aiding system, which could be a tightly coupled INS system, exist some tilt errors.

This section develops the corresponding filter architecture to estimate tilt aiding errors, given an aided error state vector tracking architecture.

III - 10.8.1 Filter architecture with integrated tilt error model

In order to get the dependency of the tilt angles, equation (III-233) is linearized about these angles.

$$\begin{aligned} \delta \mathbf{v}_e^e &= \frac{\partial \delta \mathbf{v}_e^e}{\partial (\Delta \hat{\Phi}, \Delta \hat{\Theta})} \bigg|_{\Delta \hat{\Phi}, \Delta \hat{\Theta}} \cdot \begin{bmatrix} \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \end{bmatrix} = \left[\left(\frac{\mathbf{R}_{en} \cdot \partial \mathbf{R}_{n\hat{n}}(\Delta \hat{\Phi}, \Delta \hat{\Theta})}{\partial \Delta \hat{\Phi}} \cdot \mathbf{v}_e^{\hat{n}} \right) \left(\frac{\mathbf{R}_{en} \cdot \partial \mathbf{R}_{n\hat{n}}(\Delta \hat{\Phi}, \Delta \hat{\Theta})}{\partial \Delta \hat{\Theta}} \cdot \mathbf{v}_e^{\hat{n}} \right) \right] \cdot \begin{bmatrix} \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Phi}_{\Delta \hat{\Phi}} & \boldsymbol{\Phi}_{\Delta \hat{\Theta}} \end{bmatrix} \cdot \begin{bmatrix} \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \end{bmatrix} \end{aligned} \quad (\text{III-303})$$

$$\frac{\partial \mathbf{R}_{n\hat{n}}(\Delta \hat{\Phi}, \Delta \hat{\Theta})}{\partial \Delta \hat{\Phi}} = \begin{bmatrix} 0 & \sin(\Delta \hat{\Theta}) \cdot \cos(\Delta \hat{\Phi}) & -\sin(\Delta \hat{\Theta}) \cdot \sin(\Delta \hat{\Phi}) \\ 0 & -\sin(\Delta \hat{\Phi}) & -\cos(\Delta \hat{\Phi}) \\ 0 & \cos(\Delta \hat{\Theta}) \cdot \cos(\Delta \hat{\Phi}) & -\cos(\Delta \hat{\Theta}) \cdot \sin(\Delta \hat{\Phi}) \end{bmatrix} \quad (\text{III-304})$$

$$\frac{\partial \mathbf{R}_{n\hat{n}}(\Delta \hat{\Phi}, \Delta \hat{\Theta})}{\partial \Delta \hat{\Theta}} = \begin{bmatrix} -\sin(\Delta \hat{\Theta}) & \cos(\Delta \hat{\Theta}) \cdot \sin(\Delta \hat{\Phi}) & \cos(\Delta \hat{\Theta}) \cdot \cos(\Delta \hat{\Phi}) \\ 0 & 0 & 0 \\ -\cos(\Delta \hat{\Theta}) & \sin(\Delta \hat{\Theta}) \cdot \sin(\Delta \hat{\Phi}) & -\sin(\Delta \hat{\Theta}) \cdot \cos(\Delta \hat{\Phi}) \end{bmatrix} \quad (\text{III-305})$$

The error state dynamic model in this section is based on the error state dynamic model, derived in equation (III-275). This error dynamic model is expanded about the derived tilt error model.

$$\delta \dot{\mathbf{z}} = \mathbf{A}_{\delta z} \cdot \delta \mathbf{z} + \mathbf{n}_{\delta z}$$

$$\begin{bmatrix} \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_{e,R,Aid}^e \\ \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \\ c \cdot \delta \dot{\hat{t}}_{clk} \\ c \cdot \delta \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathbf{x}TS} & \mathbf{I}_{3 \times 3} & \boldsymbol{\Phi}_{\Delta \hat{\Phi}} & \boldsymbol{\Phi}_{\Delta \hat{\Theta}} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta \mathbf{v},R,Aid} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_{e,R,Aid}^e \\ \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \\ c \cdot \delta \hat{t}_{clk} \\ c \cdot \delta \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{\mathbf{x},TS} \\ \mathbf{n}_{\delta \mathbf{v},R,Aid} \\ n_{\delta \Phi} \\ n_{\delta \Theta} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (\text{III-306})$$

The error angles are assumed as constant. In order to enable the Kalman filter to estimate these

errors, a small model uncertainty must be added, which is done by white noise.

The observation matrix is given as

$$\mathbf{H}_{\delta z} = \begin{bmatrix} \left(\frac{\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e}{C \cdot \|\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & 0 & 0 & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e}{C \cdot \|\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & 0 & 0 & \frac{1}{c} & 0 \end{bmatrix} \quad (\text{III-307})$$

The measurement error covariance matrix for the error state filter is equal to the previous chapter.

The update equation of the error state filter gets

$$\delta \mathbf{z}_{k+1}^+ = \delta \mathbf{z}_{k+1}^- + \mathbf{K}_{ES} \cdot \begin{bmatrix} \delta \tau_{\#1} \\ \vdots \\ \delta \tau_{\#Nsat} \end{bmatrix} \quad (\text{III-308})$$

The corresponding total state filter gets

$$\dot{\mathbf{z}} = \mathbf{A}_z \cdot \mathbf{z} + \mathbf{n}_z$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e^e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{N,Aid} \\ \Delta \hat{\Psi} \\ c \cdot \dot{\hat{t}}_{clk} \\ c \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta ve,N} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 2} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 2} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{N,Aid} \\ \Delta \hat{\Psi} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{0}_{\delta v,N} \\ \mathbf{0}_{2 \times 1} \\ 0 \\ 0 \end{bmatrix} \quad (\text{III-309})$$

$$\Delta \hat{\Psi} = \begin{bmatrix} \Delta \hat{\Phi} \\ \Delta \hat{\Theta} \end{bmatrix} \quad (\text{III-310})$$

The tilt aiding errors are modeled as constant values within the total state Kalman filter.

In order to estimate the tilt angles, they must be used to correct the external aiding. In contrast to the aiding delay, which was coupled within the total state dynamic model, the estimated tilt angle errors are coupled within the total state observation equation, by transforming the aiding velocity into the correct n-frame.

$$\hat{\mathbf{z}}_{k+1}^+ = \hat{\mathbf{z}}_{k+1}^- + \mathbf{K}_z \cdot \left(\mathbf{R}_{n\tilde{n}} \left(\Delta \hat{\Psi} \right) \cdot \tilde{\mathbf{v}}_e^{\tilde{n}} - \mathbf{H}_z \cdot \hat{\mathbf{z}}_{k+1}^- \right) \quad (\text{III-311})$$

The total state observation matrix is given as

$$\mathbf{H}_z = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (\text{III-312})$$

Important to note is, that the estimated tilt angles are used to correct directly the incoming velocity aiding. The tilt angles are not considered within the total state observation matrix.

The rate aiding error is only implemented as a consider state. The impact of the estimated rate aiding error on the measurement is considered in equation (III-312), but it is not used in the real feedback.

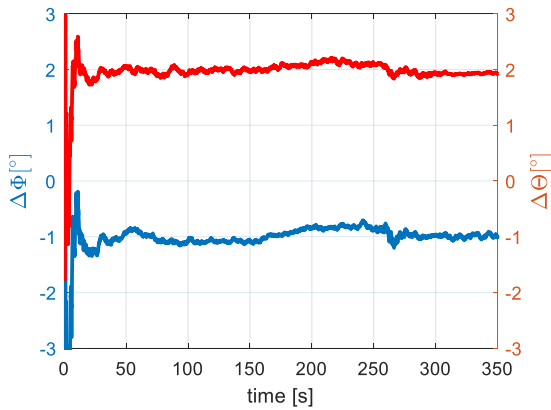
The total states are corrected by the error states according to the following equation.

$$\begin{bmatrix} \hat{\mathbf{x}}_e \\ \Delta \hat{\Phi} \\ \Delta \hat{\Theta} \\ \mathbf{c} \cdot \hat{\mathbf{t}}_{clk} \\ \mathbf{c} \cdot \hat{\mathbf{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_e \\ \Delta \hat{\Phi} \\ \Delta \hat{\Theta} \\ \mathbf{c} \cdot \hat{\mathbf{t}}_{clk} \\ \mathbf{c} \cdot \hat{\mathbf{d}}_{clk} \end{bmatrix} + \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \\ \mathbf{c} \cdot \delta \hat{\mathbf{t}}_{clk} \\ \mathbf{c} \cdot \delta \hat{\mathbf{d}}_{clk} \end{bmatrix} \quad (\text{III-313})$$

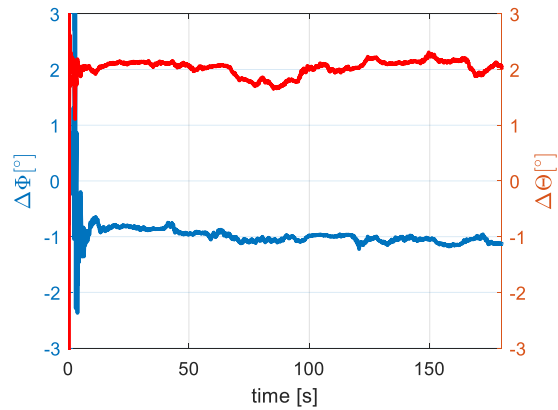
The following tilt errors are used in simulation.

$$\begin{aligned} \Delta \Phi &= -1^\circ \\ \Delta \Theta &= 2^\circ \end{aligned} \quad (\text{III-314})$$

The following figures show the estimated tilt errors, given the DA42 approach and the high dynamic trajectory.

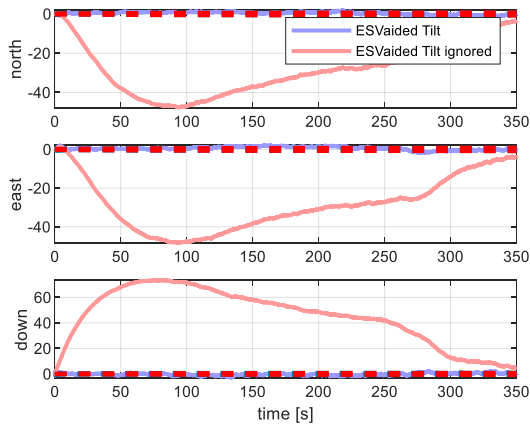


III-198 Estimated tilt errors - DA42

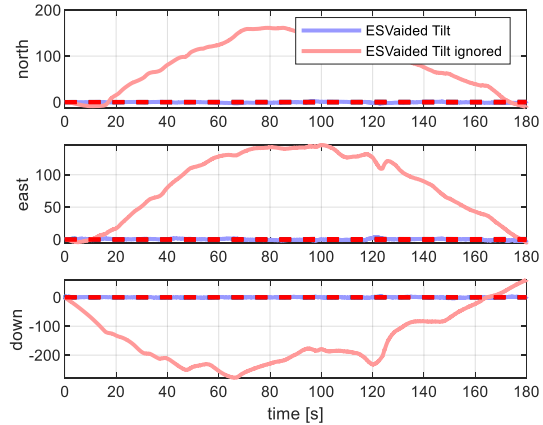


III-199 Estimated tilt errors - high dynamic scenario

In order to show the importance of estimating tilt errors within the tracking filter, the following simulations show the tracking behavior, if a tilt error is present, but not considered respectively estimated within the tracking filter.



III-200 positioning error - without tilt estimation - DA42



III-201 positioning error - without tilt estimation - Fighter

As the positioning errors in figures III-200 and III-201 show, already small tilt errors lead to unusable tracking results.

III - 10.9 Aided vs. unaided tracking

Out of the up to now analyzed unaided tracking architectures, the unaided total state vector tracking approach gives the best results. Therefore, this tracking architecture will be used as a representative of the unaided architectures.

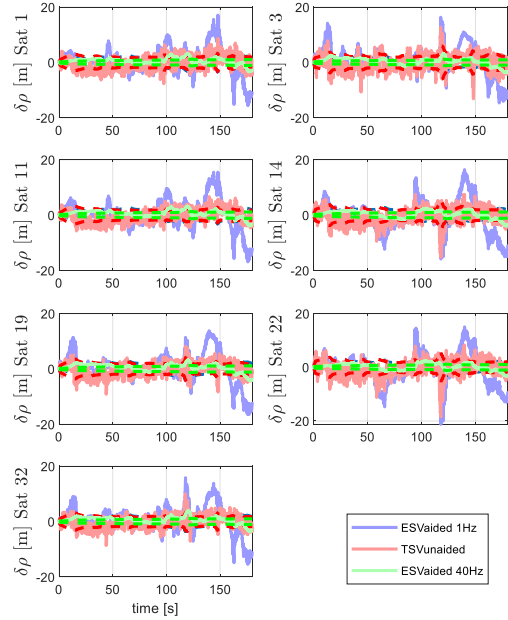
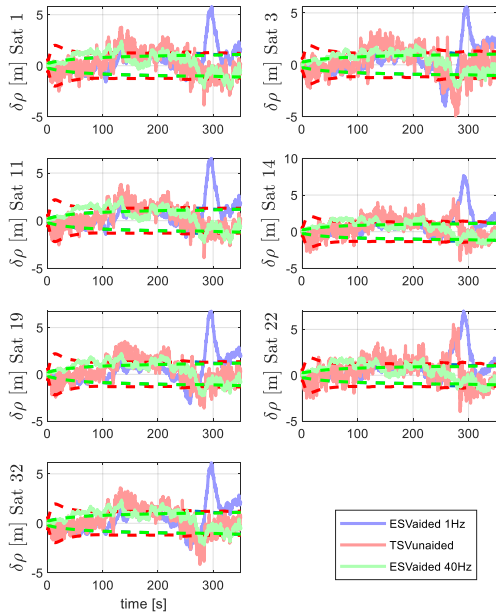
In case of aiding, the error state vector tracking architecture, where aiding is used as measurement, provides the best results.

Therefore, in this section, the unaided total state tracking architecture will be compared to the aided error state vector tracking architecture.

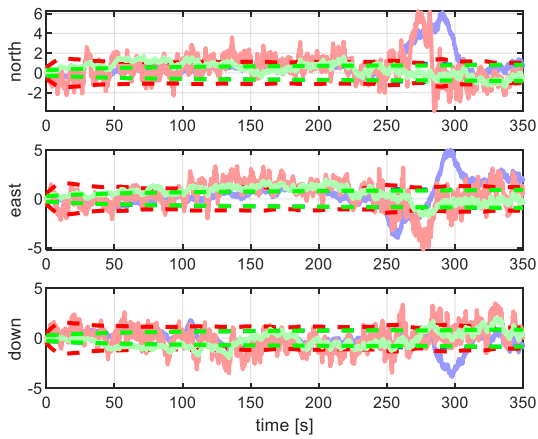
The previous comparing of two different aided error state vector tracking approaches, using aiding as a control input and aiding as a measurement, shows that both give similar results. Because the architecture, which applies aiding as a measurement also is able to estimate the platform acceleration, this architecture will be used as a representative for the aided error state vector tracking approach.

An important question in comparing aided and unaided tracking is the needed aiding rate, in order the aided architecture outperforms the unaided one. Therefore, in the following, unaided total state vector tracking is compared with aided error state vector tracking, given 1Hz aiding rate and 40Hz aiding rate.

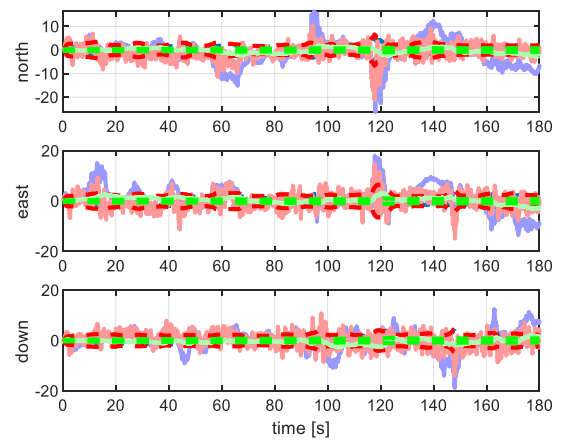
UNJAMMED



III-202 pseudorange error - DA42 - unjammed

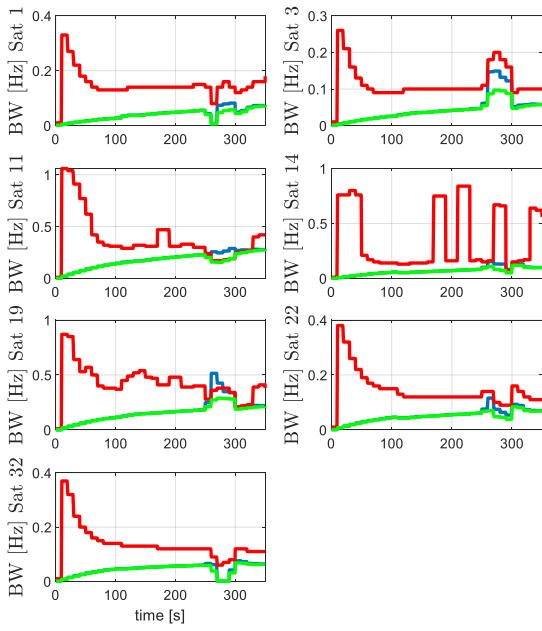


III-203 pseudorange error - high dynamic - unjammed

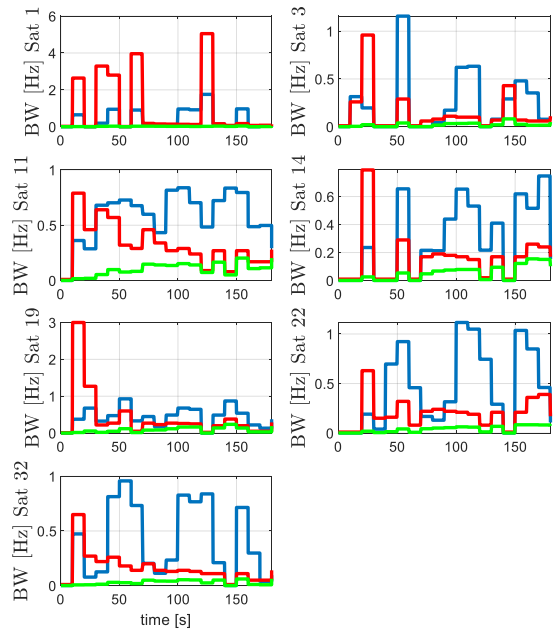


III-204 positioning error - DA42 - unjammed

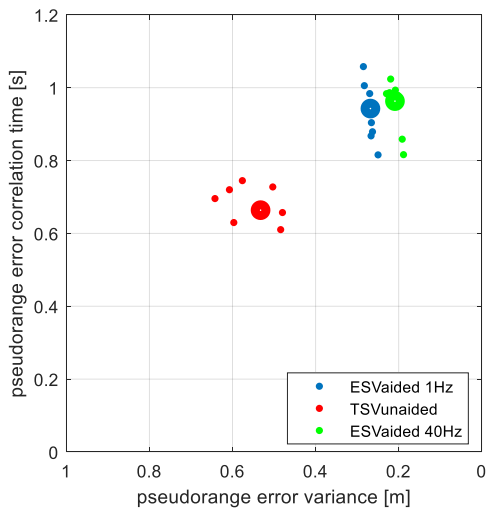
III-205 positioning error - high dynamic - unjammed



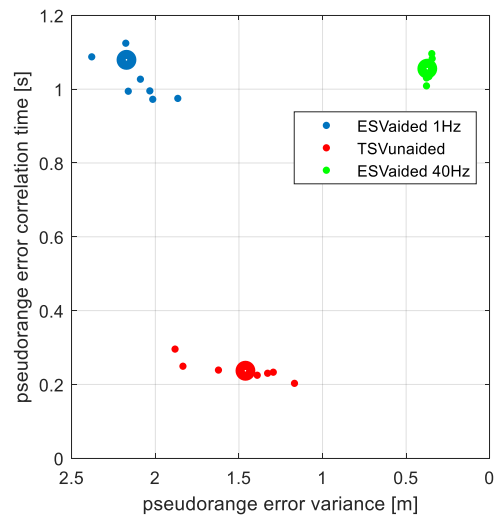
III-206 bandwidth - DA42 - unjammed



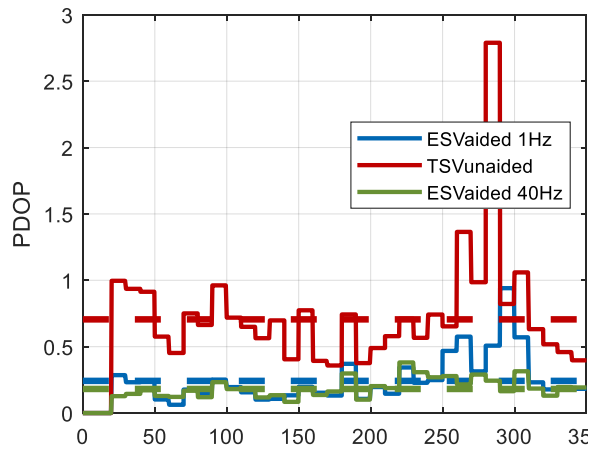
III-207 bandwidth - high dynamic - unjammed



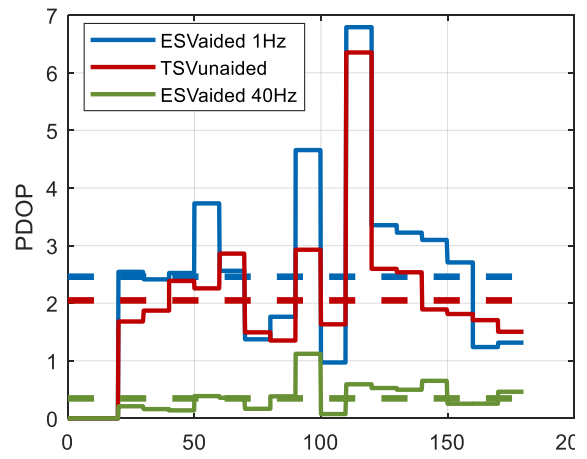
III-208 Comparing raw date - DA42 - unjammed



III-209 Comparing raw date - high dynamic - unjammed

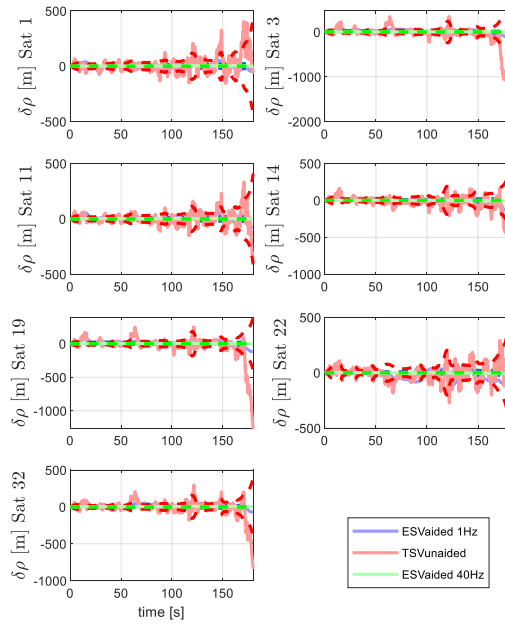
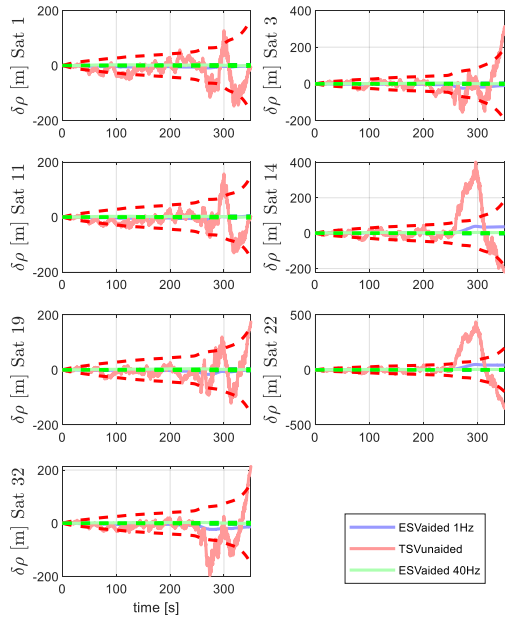


III-210 PDOP - DA42 - unjammed



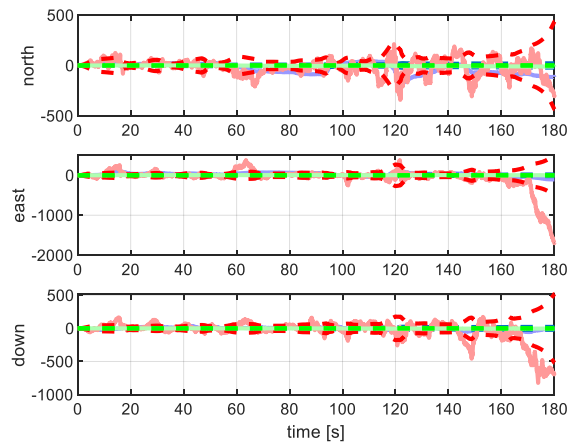
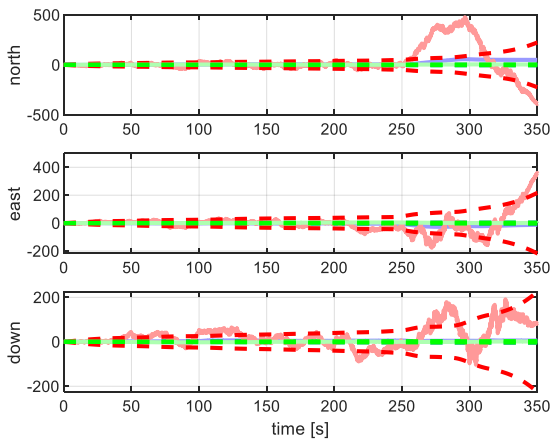
III-211 PDOP - high dynamic - unjammed

JAMMED



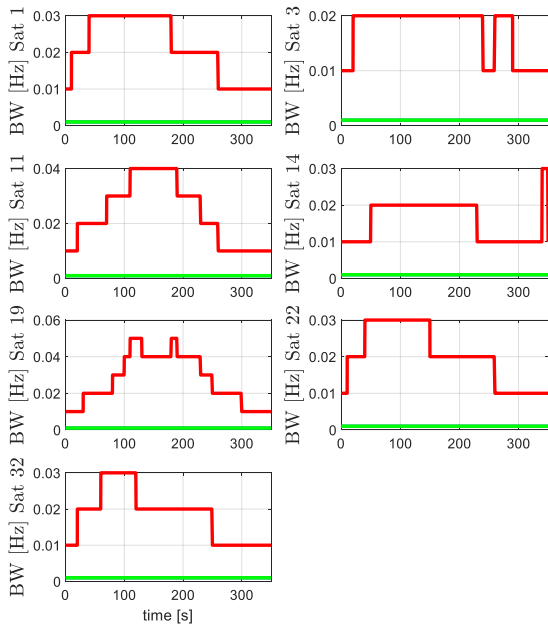
III-212 pseudorange error - DA42 - jammed

III-213 pseudorange error - high dynamic - jammed

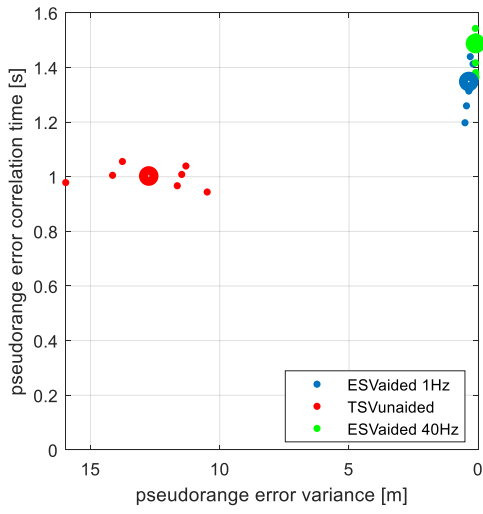


III-214 positioning error - DA42 - jammed

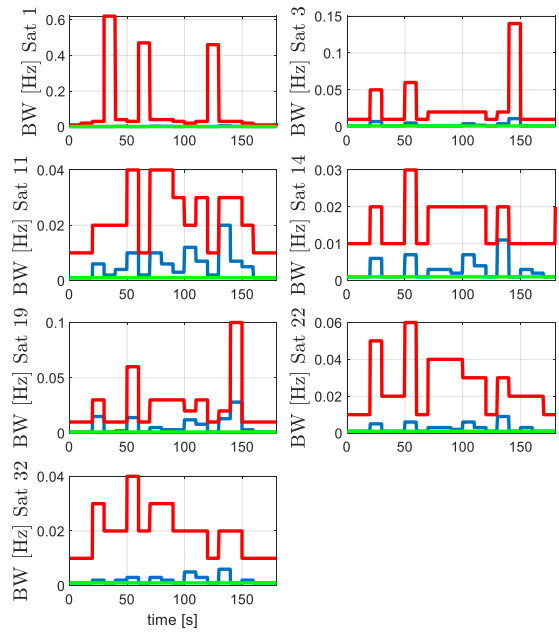
III-215 positioning error - high dynamic - jammed



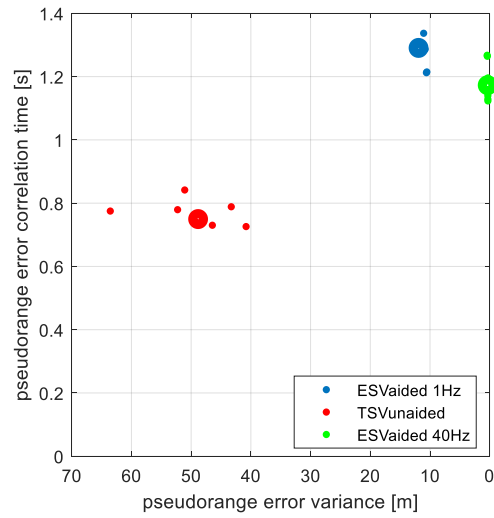
III-216 bandwidth - DA42 - jammed



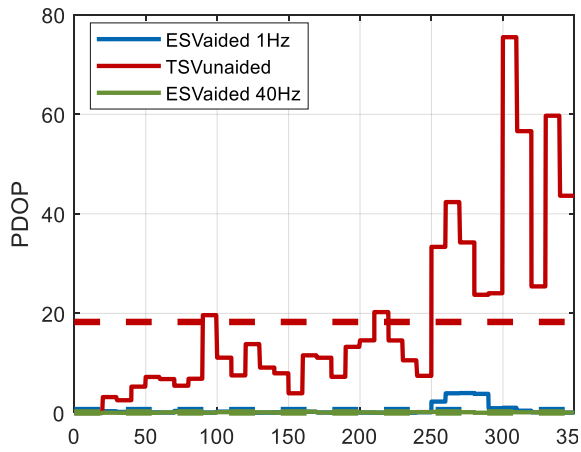
III-218 Comparing raw date - DA42 - jammed



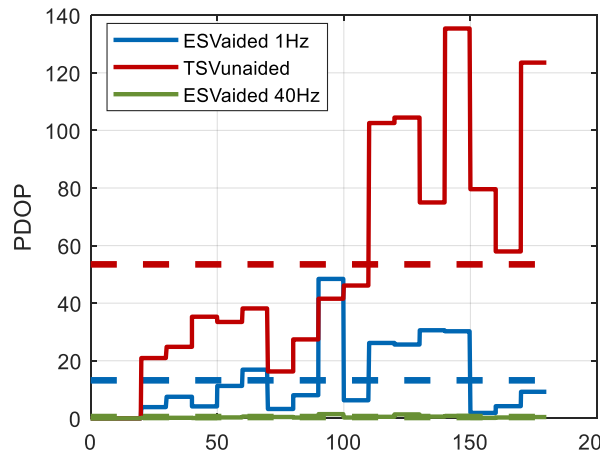
III-217 bandwidth - high dynamic - jammed



III-219 Comparing raw date - high dynamic - jammed



III-220 PDOP - DA42 - unjammed



III-221 PDOP - high dynamic - unjammed

CONCLUSION: Error State Vector aided vs. Total State Vector unaided

The unjammed simulation shows that the aided error state vector tracking approach, using a 40Hz aiding, gives in both simulation scenarios the best results.

If only a 1Hz aiding rate is used, in high dynamic situation, the unaided vector tracking approach gives better results. Using 1Hz aiding in low dynamic situations, the aided error state vector approach is almost similar to the unaided total state vector tracking approach. But even here, if short maneuvers with higher dynamic occur, the dynamic stress error of the unaided vector tracking approach is smaller than the error of 1Hz aided error state vector tracking architecture. (Even if the aided tracking architecture is also a vector tracking architecture, given only a 1 Hz aiding rate, it performs in general worse than the unaided architecture, because the aided architecture must consider also aiding errors and has a different tuning setup).

Considering the closed loop tracking bandwidth in the unjammed case in combination with low dynamic situations, the bandwidth of the aided error state approach is even in the 1Hz case, lower than the bandwidth of the unaided total state vector tracking approach. But in higher dynamic situations, when only 1Hz aiding is available, the closed loop tracking bandwidth is, despite the aiding, higher than in case of the unaided vector tracking approach.

If there is some jamming, the aided error state vector tracking approach, using 40Hz aiding, provides stable and good tracking results. The aided error state vector tracking, using 1Hz aiding, also provides stable tracking in all channels, without loss of lock, but with worse performance. The total state unaided architecture loses the lock, as the aircraft approaches the jammer and the jammer power becomes larger.

In jammed environments, the aided error state Kalman filter relies mostly on the aiding, which lowers the tracking bandwidth beyond the resolution limit of 0.001 Hz. Only in high dynamic scenarios and 1Hz aiding rate, the tracking bandwidth of the aided error state approach is slightly higher, but still below the bandwidth of the unaided vector tracking approach.

ESV aided		TSV unaided	
+	Almost no influence of jamming. Especially given a 40 Hz aiding rate.	+	Independent standalone architecture. Robust tracking results – even in case of low power jamming – without the need for an external aiding interface.
-	Aiding errors must be known and considered, respectively estimated for good results.		
-	Appropriate aiding rates are necessary. As simulations show, 1Hz is not enough.		

The following table compares the performance of unaided total state vector tracking against aided error state vector tracking in case of 1Hz aiding and 40 Hz aiding.

	TSV unaided	ESV aided 1Hz	ESV aided 40Hz	
Unjammed	+-	+-	++	DA42
	+-	-	++	Fighter
jammed	--	+-	++	DA42
	--	+-	++	Fighter

III-222 Conclusion unaided TSV - aided ESV (1Hz, 40Hz)

III - 11 Aided total state vector tracking

III - 11.1 Motivation

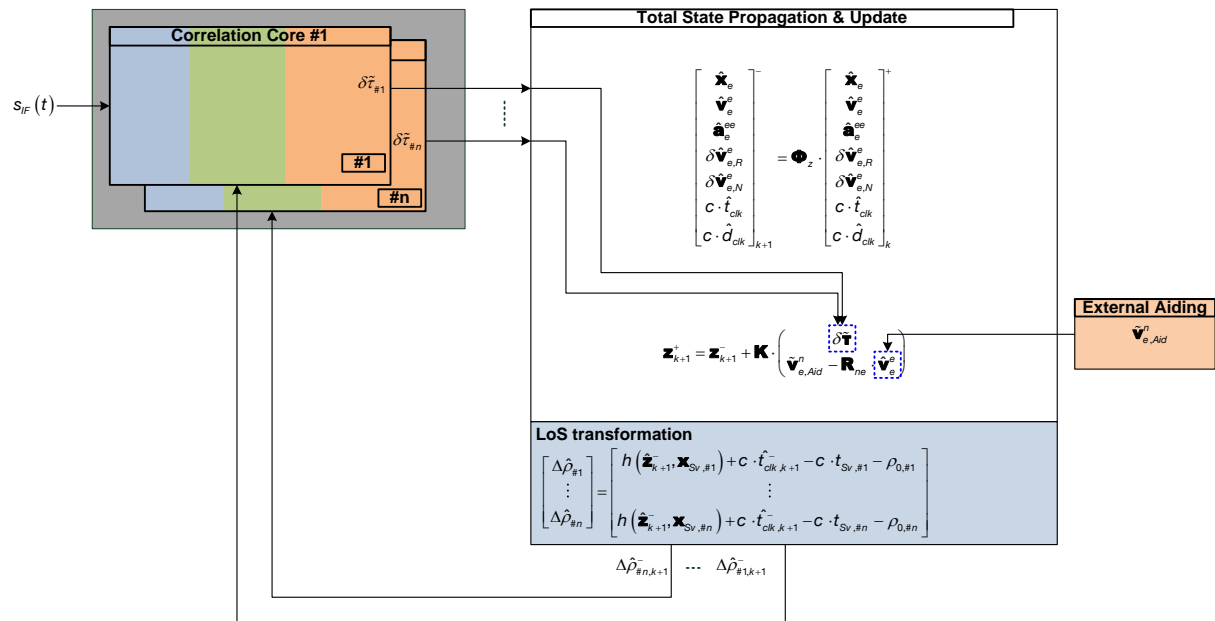
In the previous section, aided error state vector tracking was introduced together with methods for estimating, or at least considering different kind of aiding errors. The error state vector tracking approach might give the best results. Nevertheless, aiding is also possible for total state vector tracking architectures. In chapter III - 7 already an unaided total state tracking architecture was developed. Based on that, in this section, an aided total state vector tracking architecture will be developed.

For estimating aiding delay, the estimation of the platform acceleration is necessary. For that, in case of aided error state vector tracking, the aiding had to be applied as measurement to an additional total state Kalman filter. A disadvantage of this approach is the coupling between two Kalman filters, which requires special tuning measures in the error state Kalman filter.

The aided total state approach does not require an additional total state space filter. The aiding can directly be applied as measurement within the total state Kalman filter update equation.

III - 11.2 Tracking architecture

In this section at first the tracking architecture together with the Kalman filter equations are derived. The following figure shows the corresponding tracking architecture. At first only noise like aiding errors and rate aiding errors are introduced.



III-223 Total state vector tracking - aided

As tracking filter, also for this total state approach, a Kalman filter is used. The dynamic differential equation system is given below.

$$\dot{\mathbf{z}} = \mathbf{A}_z \cdot \mathbf{z} + \mathbf{n}_z$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e^e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ \delta \dot{\hat{\mathbf{v}}}_{e,R}^e \\ \delta \dot{\hat{\mathbf{v}}}_{e,N}^e \\ C \cdot \dot{\hat{t}}_{clk} \\ C \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{e,R}^e \\ \delta \hat{\mathbf{v}}_{e,N}^e \\ C \cdot \hat{t}_{clk} \\ C \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v R} \\ \mathbf{n}_{\delta v N} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (III-315)$$

$$\mathbf{A}_a = \begin{bmatrix} \frac{-1}{\tau_{a,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{a,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{a,z}} \end{bmatrix} \quad \mathbf{A}_{\delta v,R} = \begin{bmatrix} \frac{-1}{\tau_{\delta v,R,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v,R,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v,R,z}} \end{bmatrix} \quad \mathbf{A}_{\delta v,N}^n = \begin{bmatrix} \frac{-1}{\tau_{\delta v,N,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v,N,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v,N,z}} \end{bmatrix} \quad (III-316)$$

The matrix $\mathbf{A}_{\delta v,N}^e$ is derived in section III - 9.3 .

$$\mathbf{A}_{\delta v,N}^e = (\mathbf{R}_{en} \cdot \mathbf{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N} \cdot \mathbf{R}_{en}^T) \quad (III-317)$$

The noise like aiding error $\delta \hat{\mathbf{v}}_{e,N}^e$ and the rate aiding error $\delta \hat{\mathbf{v}}_{e,R}^e$ are not coupled within the dynamic equation. Both errors are only implemented as consider states and considered within the measurement equation.

The acceleration is assumed as Gauss Markov process, having a zero mean.

The corresponding model error covariance matrix is given as:

$$\mathbf{Q}_{TS} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v R} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v N} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & c^2 \sigma_{\delta tclk}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 d_{\delta tclk}^2 \end{bmatrix} \quad (III-318)$$

$$\mathbf{Q}_{\delta vN} = \mathbf{R}_{en} \cdot \begin{bmatrix} \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,n,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,n,N}^n}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,d,N}^n}^2 \cdot 2}{\tau_{\delta v_{e,d,N}^n}} \end{bmatrix} \cdot \mathbf{R}_{en}^T, \mathbf{Q}_{\delta vR} = \begin{bmatrix} \frac{\sigma_{\delta v_{e,x,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,x,R}^e}} & 0 & 0 \\ 0 & \frac{\sigma_{\delta v_{e,y,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,y,R}^e}} & 0 \\ 0 & 0 & \frac{\sigma_{\delta v_{e,z,R}^e}^2 \cdot 2}{\tau_{\delta v_{e,z,R}^e}} \end{bmatrix} \quad (\text{III-319})$$

This architecture uses two measurements within the update step of the Kalman filter.

The first measurement is the discriminator output $\delta \tilde{\mathbf{T}}$. This is already the innovation, which is calculated within the tracking core.

$$\delta \tilde{\mathbf{T}} = \begin{bmatrix} \Delta \tilde{\tau}_{\#1} \\ \vdots \\ \Delta \tilde{\tau}_{\#Nsat} \end{bmatrix} - \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T \\ \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T \end{bmatrix} \cdot \Delta \hat{\mathbf{x}}_e + \begin{bmatrix} \hat{t}_{clk} \\ \vdots \\ \hat{t}_{clk} \end{bmatrix} = \begin{bmatrix} \Delta \tilde{\tau}_{\#1} \\ \vdots \\ \Delta \tilde{\tau}_{\#Nsat} \end{bmatrix} - \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T \\ \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T \end{bmatrix} \cdot (\hat{\mathbf{x}}_e - \hat{\mathbf{x}}_{0,e}) + \begin{bmatrix} \hat{t}_{clk} \\ \vdots \\ \hat{t}_{clk} \end{bmatrix} \quad (\text{III-320})$$

The second measurement is the aiding velocity. The innovation for the aiding velocity is calculated within the update equation.

$$\delta \tilde{\mathbf{v}} = \tilde{\mathbf{v}}_{e,Aid}^n - \mathbf{R}_{ne} \cdot \hat{\mathbf{v}}_e^e \quad (\text{III-321})$$

The update equation gets

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tilde{\mathbf{T}} \\ \tilde{\mathbf{v}}_{e,Aid}^n - \mathbf{R}_{ne} \cdot \hat{\mathbf{v}}_e^e \end{pmatrix} \quad (\text{III-322})$$

In order to consider the rate aiding error and the noise like aiding error within the update equation, the observation matrix gets.

$$\mathbf{H} = \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (\text{III-323})$$

Because these two aiding error components shall only be realized as consider states, they are not used in the real update equation(III-321). They are only used within the observation matrix, used for Kalman gain calculation.

Because the transformation from the position state to the pseudoranges is nonlinear, the observation matrix (III-323) is linearized about the actual position $\hat{\mathbf{x}}_e$.

The real feedback for NCO command is calculated according to the next equation.

$$\hat{\rho}_{\#n,k+1}^- = h(\hat{\mathbf{z}}_{k+1}^-, \mathbf{x}_{Sv,\#n}) = \sqrt{(\mathbf{x}_{Sv,\#n} - \hat{\mathbf{x}}_{e,k+1}^-)^T \cdot (\mathbf{x}_{Sv,\#n} - \hat{\mathbf{x}}_{e,k+1}^-)} + \mathbf{C} \cdot \hat{t}_{clk,k+1}^- \quad (\text{III-324})$$

The considered rate and noise like aiding errors deteriorate the aiding quality, which in turn causes the Kalman filter to rely less on the aiding information and put higher weight on the discriminator measurement.

Because the noise like aiding error and rate aiding error are already considered within the observation equation, they do not appear in the measurement error variance matrix.

The measurement error variance matrix gets

$$\mathbf{R} = \begin{bmatrix} \sigma_{\delta r, \#1}^2 & \cdots & 0 & \mathbf{0}_{1 \times 3} \\ \vdots & \ddots & \vdots & \mathbf{0}_{1 \times 3} \\ 0 & \cdots & \sigma_{\delta r, \#Nsat}^2 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \cdots & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (\text{III-325})$$

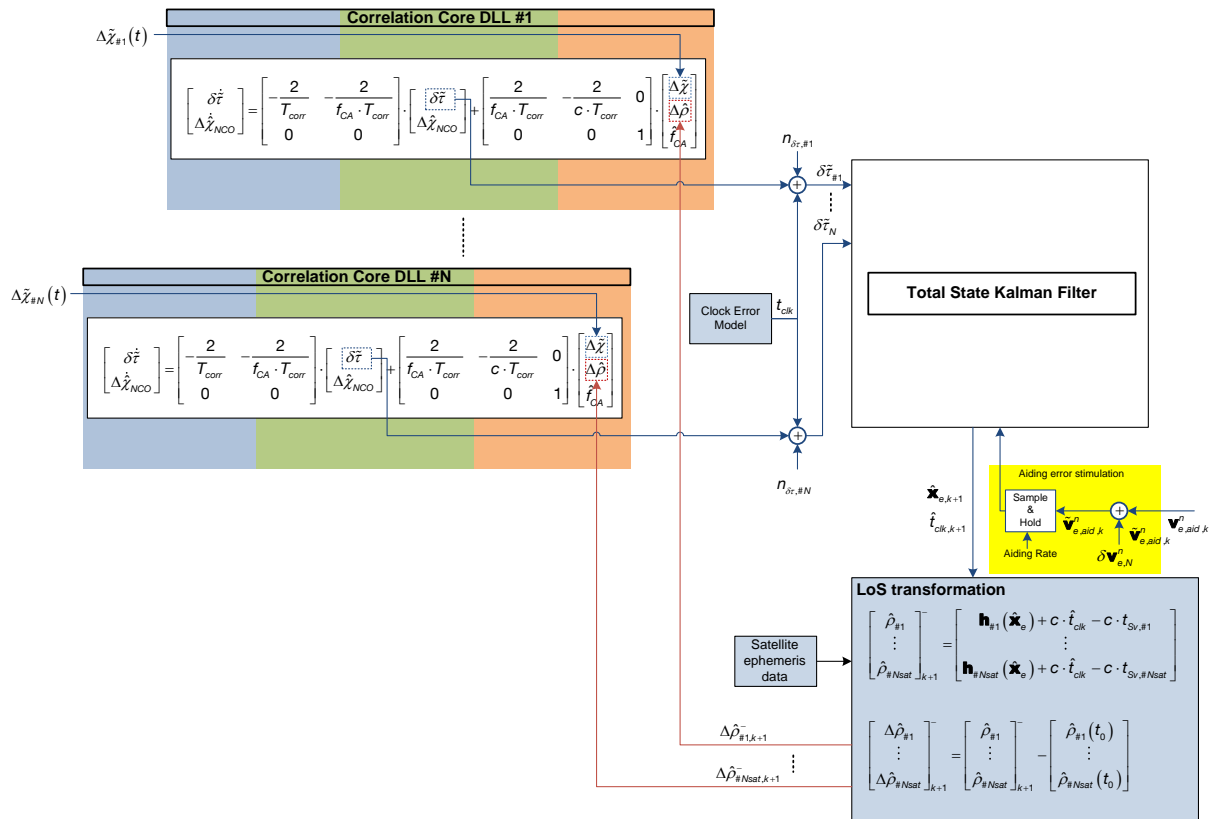
For tuning the Kalman filter, the already developed scenario matched tuning is applied together with the known information about the aiding errors.

III - 11.3 Equivalent base band dynamic state space simulation

III - 11.3.1 Distributed state space simulation

For equivalent base band simulation, the distributed state space formulation is used. There, besides the discriminator noise, also aiding together with aiding errors is simulated.

The following figure shows the corresponding concept.



III-224 Distributed state space form

III - 11.3.2 Centralized state space representation

In this section, the centralized state space form for the aided total state vector tracking approach will be derived.

Out of this, the closed loop tracking bandwidth for every satellite channel can be calculated.

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{u}$$

$$\begin{bmatrix} \dot{\delta\tilde{t}}_{\#1} \\ \dot{\Delta\tilde{\lambda}}_{NCO,\#1} \\ \vdots \\ \dot{\delta\tilde{t}}_{\#N} \\ \dot{\Delta\tilde{\lambda}}_{NCO,\#N} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}^{2 \times 2} & \dots & \mathbf{0}^{2 \times 2} & \mathbf{A}_{h,\#1}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}^{2 \times 2} & \dots & \mathbf{A}_{N,N}^{2 \times 2} & \mathbf{A}_{h,\#N}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} & \mathbf{0}^{2 \times 3} \\ \mathbf{A}_{Kx,\#1}^{3 \times 2} & \dots & \mathbf{A}_{Kx,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & -\mathbf{A}_{Kx,v}^{3 \times 3} + \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{Kv,\#1}^{3 \times 2} & \dots & \mathbf{A}_{Kv,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & -\mathbf{A}_{Kv,v}^{3 \times 3} & \mathbf{I}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{Ka,\#1}^{3 \times 2} & \dots & \mathbf{A}_{Ka,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & -\mathbf{A}_{Ka,v}^{3 \times 3} & \mathbf{A}_a^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{K\delta vR,\#1}^{3 \times 2} & \dots & \mathbf{A}_{K\delta vR,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & -\mathbf{A}_{K\delta vR,v}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_{\delta vR}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{K\delta vN,\#1}^{3 \times 2} & \dots & \mathbf{A}_{K\delta vN,\#N}^{3 \times 2} & \mathbf{0}^{3 \times 3} & -\mathbf{A}_{K\delta vN,v}^{3 \times 3} & \mathbf{0}^{3 \times 3} & \mathbf{A}_{\delta vN}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{A}_{K\delta tclk,\#1}^{1 \times 2} & \dots & \mathbf{A}_{K\delta tclk,\#N}^{1 \times 2} & \mathbf{0}^{1 \times 3} & -\mathbf{A}_{K\delta tclk,v}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \\ \mathbf{A}_{K\delta\delta clk,\#1}^{1 \times 2} & \dots & \mathbf{A}_{K\delta\delta clk,\#N}^{1 \times 2} & \mathbf{0}^{1 \times 3} & -\mathbf{A}_{K\delta\delta clk,v}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{t}_{\#1} \\ \Delta\tilde{\lambda}_{NCO,\#1} \\ \vdots \\ \delta\tilde{t}_{\#Nsat} \\ \Delta\tilde{\lambda}_{NCO,\#Nsat} \\ \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta\hat{\mathbf{v}}_{e,R}^e \\ \delta\hat{\mathbf{v}}_{e,N}^e \\ \mathbf{c} \cdot \delta\hat{t}_{clk} \\ \mathbf{c} \cdot \delta\hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} -\frac{2}{\mathbf{c} \cdot T_{corr}} \\ 0 \\ \vdots \\ -\frac{2}{\mathbf{c} \cdot T_{corr}} \\ 0 \\ \vdots \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_X^{2 \times N \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \\ \mathbf{0}^{3 \times N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0}^{2 \times N \times 3} \\ \mathbf{A}_{Kx,v}^{3 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{Kv,v}^{3 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{Ka,v}^{3 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{K\delta vR,v}^{3 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{K\delta vN,v}^{3 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{K\delta tclk,v}^{1 \times 3} \cdot \mathbf{R}_{en} \\ \mathbf{A}_{K\delta\delta clk,v}^{1 \times 3} \cdot \mathbf{R}_{en} \end{bmatrix} \cdot \begin{bmatrix} \Delta\tilde{\lambda}_{\#1} \\ \vdots \\ \Delta\tilde{\lambda}_{\#N} \\ \tilde{\mathbf{v}}_{n,Aid}^e \end{bmatrix}$$

$$\mathbf{A}_{1,1}^{2 \times 2} = \dots = \mathbf{A}_{N,N}^{2 \times 2} = \begin{bmatrix} -2 & -2 \cdot T_{C/A} \\ T_{corr} & 0 \end{bmatrix} \quad (\text{III-327})$$

$$\mathbf{A}_{h,\#1}^{2 \times 2} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{\partial h(\mathbf{x}_{Sv\#1}, \hat{\mathbf{x}}_e)}{\partial \hat{\mathbf{x}}_e} \Big|_{\hat{\mathbf{x}}_{e,k}} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} = \dots$$

$$\begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{\partial \sqrt{(\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)^T \cdot (\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e)}}{\partial \hat{\mathbf{x}}_e} \Big|_{\hat{\mathbf{x}}_{e,k}} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#1,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \quad (\text{III-328})$$

$$\mathbf{A}_{h,\#N}^{2 \times 2} = \begin{bmatrix} \frac{2}{c \cdot T_{corr}} \cdot \frac{(\mathbf{x}_{Sv\#N,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#N,k} - \hat{\mathbf{x}}_{e,k}\|} \\ \mathbf{0}^{1 \times 3} \end{bmatrix}$$

$$\mathbf{A}_{Kx,\#1}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{1,1} & 0 \\ k_{2,1} & 0 \\ k_{3,1} & 0 \end{bmatrix} \quad \mathbf{A}_{Kx,\#N}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{1,N} & 0 \\ k_{2,N} & 0 \\ k_{3,N} & 0 \end{bmatrix} \quad \mathbf{A}_{Kx,v}^{3 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{1,N+1} & k_{1,N+2} & k_{1,N+3} \\ k_{2,N+1} & k_{2,N+2} & k_{2,N+3} \\ k_{3,N+1} & k_{3,N+2} & k_{3,N+3} \end{bmatrix} \quad (\text{III-329})$$

$$\mathbf{A}_{Kv,\#1}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{4,1} & 0 \\ k_{5,1} & 0 \\ k_{6,1} & 0 \end{bmatrix} \quad \mathbf{A}_{Kv,\#N}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{4,N} & 0 \\ k_{5,N} & 0 \\ k_{6,N} & 0 \end{bmatrix} \quad \mathbf{A}_{Kv,v}^{3 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{4,N+1} & k_{4,N+2} & k_{4,N+3} \\ k_{5,N+1} & k_{5,N+2} & k_{5,N+3} \\ k_{6,N+1} & k_{6,N+2} & k_{6,N+3} \end{bmatrix} \quad (\text{III-330})$$

$$\mathbf{A}_{Ka,\#1}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{7,1} & 0 \\ k_{8,1} & 0 \\ k_{9,1} & 0 \end{bmatrix} \quad \mathbf{A}_{Ka,\#N}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{7,N} & 0 \\ k_{8,N} & 0 \\ k_{9,N} & 0 \end{bmatrix} \quad \mathbf{A}_{Ka,v}^{3 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{7,N+1} & k_{7,N+2} & k_{7,N+3} \\ k_{8,N+1} & k_{8,N+2} & k_{8,N+3} \\ k_{9,N+1} & k_{9,N+2} & k_{9,N+3} \end{bmatrix} \quad (\text{III-331})$$

$$\mathbf{A}_{K\delta vR,\#1}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{10,1} & 0 \\ k_{11,1} & 0 \\ k_{12,1} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta vR,\#N}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{10,N} & 0 \\ k_{11,N} & 0 \\ k_{12,N} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta vR,v}^{3 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{10,N+1} & k_{10,N+2} & k_{10,N+3} \\ k_{11,N+1} & k_{11,N+2} & k_{11,N+3} \\ k_{12,N+1} & k_{12,N+2} & k_{12,N+3} \end{bmatrix} \quad (\text{III-332})$$

$$\mathbf{A}_{K\delta vN,\#1}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{13,1} & 0 \\ k_{14,1} & 0 \\ k_{15,1} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta vN,\#N}^{3 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{13,N} & 0 \\ k_{14,N} & 0 \\ k_{15,N} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta vN,v}^{3 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{13,N+1} & k_{13,N+2} & k_{13,N+3} \\ k_{14,N+1} & k_{14,N+2} & k_{14,N+3} \\ k_{15,N+1} & k_{15,N+2} & k_{15,N+3} \end{bmatrix} \quad (\text{III-333})$$

$$\mathbf{A}_{K\delta tclk,\#1}^{1 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{16,1} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta tclk,\#N}^{1 \times 2} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{16,N} & 0 \end{bmatrix} \quad \mathbf{A}_{K\delta tclk,v}^{1 \times 3} = \frac{1}{T_{corr}} \cdot \begin{bmatrix} k_{16,N+1} & k_{16,N+2} & k_{16,N+3} \end{bmatrix} \quad (\text{III-334})$$

$$\mathbf{A}_{K\delta dclk,\#1}^{1 \times 2} = \frac{1}{T_{corr}} \cdot [k_{17,1} \quad 0] \quad \mathbf{A}_{K\delta dclk,\#N}^{1 \times 2} = \frac{1}{T_{corr}} \cdot [k_{17,N} \quad 0] \quad \mathbf{A}_{K\delta dclk,y}^{1 \times 3} = \frac{1}{T_{corr}} \cdot [k_{17,N+1} \quad k_{17,N+2} \quad k_{17,N+3}] \quad (\text{III-335})$$

$$\mathbf{A}_{\delta vR}^{3 \times 3} = \begin{bmatrix} \frac{-1}{\tau_{\delta v_e, \delta vR,x}} & 0 & 0 \\ 0 & \frac{-1}{\tau_{\delta v_e, \delta vR,y}} & 0 \\ 0 & 0 & \frac{-1}{\tau_{\delta v_e, \delta vR,z}} \end{bmatrix}, \quad \mathbf{A}_{\delta vN}^{3 \times 3} = (\mathbf{R}_{en} \cdot \mathbf{\Omega}_{en}^n \cdot \mathbf{R}_{en}^T + \mathbf{R}_{en} \cdot \mathbf{A}_{\delta v,N} \cdot \mathbf{R}_{en}^T) \quad (\text{III-336})$$

$$\mathbf{B}_X^{2 \cdot N \times N} = \begin{bmatrix} \mathbf{B}_{sub}^{2 \times 1} & \dots & \mathbf{0}^{2 \times 1} \\ \vdots & \ddots & \vdots \\ \mathbf{0}^{2 \times 1} & \dots & \mathbf{B}_{sub}^{2 \times 1} \end{bmatrix}, \quad \mathbf{B}_{sub}^{2 \times 1} = \begin{bmatrix} 2 \cdot T_{C/A} \\ T_{corr} \\ 0 \end{bmatrix} \quad (\text{III-337})$$

In order to get the closed loop tracking bandwidth for one satellite channel, the following SISO transfer function is needed.

$$\tilde{\chi}_{\#1} \rightarrow \frac{1}{\lambda_{C/A}} \cdot \hat{\rho}_{\#1} \Leftrightarrow \tilde{\chi}_{\#1} \rightarrow \hat{\chi}_{\#1}: \quad g_{\#1,\#1}(s) = \mathbf{c}_{\#1}^T \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}_{\#1} \quad (\text{III-338})$$

Due to position state formulation, the observation is nonlinear and needs to be linearized about the actual positioning estimation.

$$\begin{aligned} \hat{\chi}_{\#n} &= c(\mathbf{x}_{\#n}, \hat{\mathbf{z}}) = \sqrt{\|\mathbf{x}_{\#n} - \hat{\mathbf{x}}_e\|} \\ \delta \hat{\chi}_{\#n} &= \left. \frac{\partial c(\mathbf{x}_{\#n}, \hat{\mathbf{z}})}{\partial \hat{\mathbf{x}}_e} \right|_{\hat{\mathbf{x}}_e} = -\frac{1}{\lambda_{C/A}} \cdot \frac{(\mathbf{x}_{\#n} - \hat{\mathbf{x}}_e)^T}{\|\mathbf{x}_{\#n} - \hat{\mathbf{x}}_e\|} \cdot \delta \mathbf{x}_e \\ \mathbf{c}_k^T &= \begin{bmatrix} \mathbf{0}^{1 \times 2N_{sat}} & -\frac{1}{\lambda_{C/A}} \cdot \frac{(\mathbf{x}_{Sv\#n,k} - \hat{\mathbf{x}}_{e,k})^T}{\|\mathbf{x}_{Sv\#n,k} - \hat{\mathbf{x}}_{e,k}\|} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 & 0 \end{bmatrix} \end{aligned} \quad (\text{III-339})$$

With $\mathbf{x}_{\#n}$ being the actual satellite position of satellite n.

The corresponding input vector gets

$$\mathbf{b}_{\#n} = \begin{bmatrix} \vdots \\ \left[\begin{array}{c} 2 \cdot T_{C/A} \\ T_{corr} \\ 0 \end{array} \right] \\ \vdots \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \\ \mathbf{0}^{2 \times 1} \end{bmatrix} \quad (\text{III-340})$$

III - 11.4 Aiding delay estimation

The aiding delay $\delta \hat{\mathbf{v}}_{e,OOS,k}^e$ was already introduced in chapter III - 10.7.2.

The applied aiding, which is delayed about $\Delta \hat{T}_{OOS}$ can be written as:

$$\tilde{\mathbf{v}}_{e,k}^e = \mathbf{v}_{e,k}^e - \delta \mathbf{v}_{e,OOS,k}^e = \mathbf{v}_{e,k}^e - \mathbf{a}_{e,k}^{ee} \cdot \Delta T_{OOS} \quad (\text{III-341})$$

In order to estimate the aiding delay, ΔT_{OOS} is modeled as an own state in the dynamic model.

Regarding the feedback respectively coupling of the estimated aiding delay, there are two possibilities. The following state space differential equation shows the first one, where the estimated aiding delay is not coupled in the state matrix, but only within the observation equation.

$$\dot{\mathbf{z}}_{TS} = \mathbf{A}_{TS} \cdot \mathbf{z}_{TS} + \mathbf{n}_{TS}$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{e,R}^e \\ \delta \hat{\mathbf{v}}_{e,N}^e \\ \Delta \hat{T}_{OOS} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R}^e & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{e,R,Aid}^e \\ \delta \hat{\mathbf{v}}_{e,N,Aid}^e \\ \Delta \hat{T}_{OOS} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v,R,Aid} \\ \mathbf{n}_{\delta v,N,Aid} \\ n_{\Delta T,00s} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (III-342)$$

Despite $\Delta \hat{T}_{OOS}$ is typically a constant value, it has to be modeled as a Wiener process, to add a little uncertainty which enables the Kalman filter to estimate the value. If no noise term $n_{\Delta T,00s}$ would be added, the value remains zero or at its initial one.

According to equation (III-341), the estimated aiding delay is used within the innovation.

The update equation gets

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K} \cdot \left(\begin{array}{c} \delta \tilde{\mathbf{r}} \\ \tilde{\mathbf{v}}_{e,Aid}^n - (\mathbf{R}_{ne} \cdot \hat{\mathbf{v}}_e^e - \mathbf{R}_{ne} \cdot \hat{\mathbf{a}}_e^{ee} \cdot \Delta \hat{T}_{OOS}) \end{array} \right) \quad (III-343)$$

Based on (III-323), the observation matrix gets

$$\mathbf{H} = \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{R}_{ne} & -\mathbf{R}_{ne} \cdot \hat{\mathbf{a}}_e^{ee} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (III-344)$$

The noise like aiding error as well as the error due to a low aiding rate are only modeled as consider states. These terms are considered within the observation matrix \mathbf{H} but not in the real innovation in equation (III-237). Because the actual total state acceleration $\hat{\mathbf{a}}_e^{ee}$ is needed within the observation equation, it gets nonlinear.

The second possibility of coupling the estimated aiding delay with the other states is shown below.

$$\dot{\mathbf{z}}_{TS} = \mathbf{A}_{TS} \cdot \mathbf{z}_{TS} + \mathbf{n}_{TS}$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ \delta \dot{\hat{\mathbf{v}}}_{e,R} \\ \delta \dot{\hat{\mathbf{v}}}_{e,N} \\ \Delta \dot{\hat{T}}_{OOS} \\ c \cdot \dot{\hat{t}}_{clk} \\ c \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \hat{\mathbf{a}}_e^{ee} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R}^e & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e \\ \hat{\mathbf{a}}_e^{ee} \\ \delta \hat{\mathbf{v}}_{e,R,Aid} \\ \delta \hat{\mathbf{v}}_{e,N,Aid} \\ \Delta \hat{T}_{OOS} \\ c \cdot \hat{t}_{clk} \\ c \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v,R,Aid} \\ \mathbf{n}_{\delta v,N,Aid} \\ n_{\Delta T,oos} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (III-345)$$

In equation (III-345), the estimated aiding delay is coupled within the state transition matrix. The state transition matrix becomes nonlinear, through its dependence on the actual acceleration.

The estimated velocity $\hat{\mathbf{v}}_e$ is the delayed one. To get the correct velocity, it needs to be compensated by the estimated aiding delay.

$$\hat{\mathbf{v}}_e = \hat{\mathbf{v}}_e + \hat{\mathbf{a}}_e^{ee} \cdot \Delta \hat{T}_{OOS} \quad (III-346)$$

The corresponding observation equation and innovation gets

$$\mathbf{z}_{k+1}^+ = \mathbf{z}_{k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tilde{\mathbf{T}} \\ \tilde{\mathbf{v}}_{e,Aid}^n - \mathbf{R}_{ne} \cdot \hat{\mathbf{v}}_e \end{pmatrix} \quad (III-347)$$

$$\mathbf{H} = \begin{bmatrix} \frac{1}{c} \cdot \mathbf{e}_{LoS,\#1}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{c} \cdot \mathbf{e}_{LoS,\#Nsat}^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{c} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{ne} & \mathbf{R}_{ne} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (III-348)$$

The aiding error caused by low aiding rate and the noise like aiding error are only modeled as consider states. Therefore, in the innovation in equation (III-347), these two errors are not included.

A suitable tuning of the uncertainty $n_{\Delta T,oos}$ respectively its error variance $\sigma_{\Delta T,oos}^2$ in the model error covariance matrix \mathbf{Q}_{TS} is important.

$$\mathbf{Q}_{TS} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v R} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_{\delta v N} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \sigma_{\Delta T_{00s}}^2 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 \sigma_{\delta t_{clk}}^2 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & c^2 d_{\delta t_{clk}}^2 \end{bmatrix} \quad (\text{III-349})$$

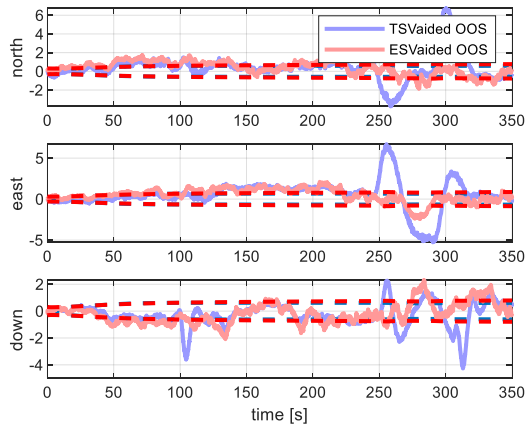
For \mathbf{Q}_a again the scenario matched tuning is used to determine the corresponding acceleration model error variance. The given state error covariance matrix is valid for both aiding delay error couplings.

Regarding $\sigma_{\Delta T_{00s}}^2$, a very low assumed error variance would lead to long estimation times until the correct aiding delay is estimated. Moreover, high velocity variations are necessary to estimate the aiding delay. On the other side, in case of a very high aiding delay error variance, some dynamic will be coupled into the delay estimation.

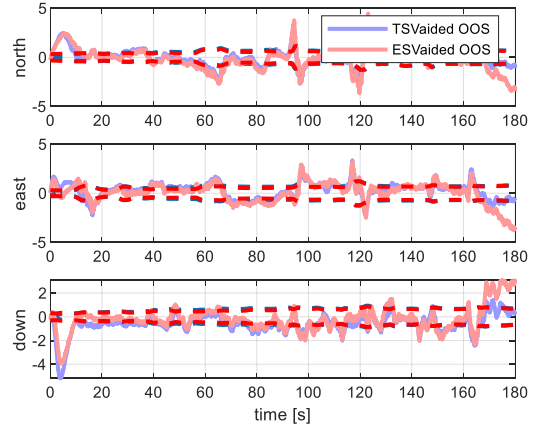
Comparing “aiding delay estimation” given the aided vector total state architecture and the aided vector error state architecture

Compared to the aided error state vector tracking approach, the tuning of the aided total state vector tracking approach is much more complicated, because the total state filter has to care about the whole platform dynamic and at the same, has to estimate the aiding and tracking errors.

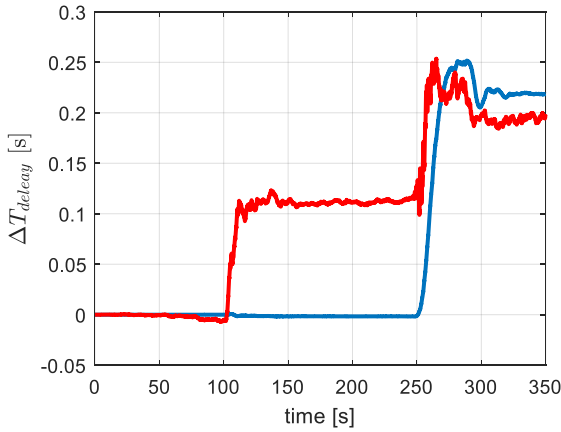
Given the aided vector total state approach, coupling the estimated aiding delay within the state transition matrix, turned out to be the better approach, especially in scenarios of higher dynamic. The figures below show the tracking results for the low dynamic DA42 approach and for the high dynamic scenario.



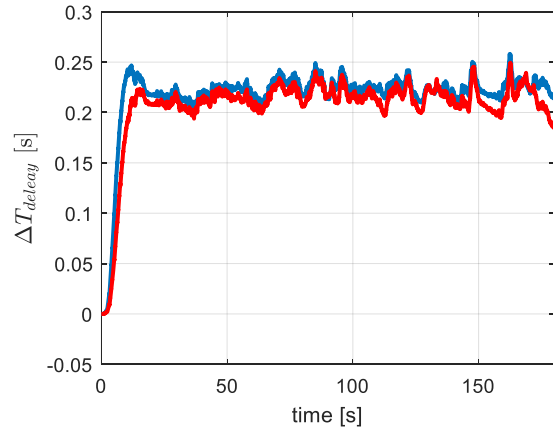
III-225 positioning error - aiding - OOS error - DA42



III-226 positioning error - aiding - OOS error - high dynamic



III-227 estimated aiding delay - DA42



III-228 estimated aiding delay - high dynamic

The simulations above show a worse aiding delay estimation performance for the DA42 approach. But this result is only valid for the actual used tuning. As already mentioned, the tuning of the aided total state tracking loop is very sensitive and may be further optimized to get similar estimation results.

But as a conclusion, the tuning sensitivity regarding aiding delay estimation, makes the aided total state tracking unpracticable for real scenarios, where the exact mission dynamic is not always known precisely, pre mission.

III - 11.5 Tilt error estimation

The tilt angle errors are already introduced in the previous section. The estimated tilt angle errors are modeled as constant values.

$$\begin{bmatrix} \delta\Delta\dot{\Phi} \\ \delta\Delta\dot{\Theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\Delta\hat{\Phi} \\ \delta\Delta\hat{\Theta} \end{bmatrix} + \begin{bmatrix} n_{\Delta\hat{\Phi}} \\ n_{\Delta\hat{\Theta}} \end{bmatrix} \quad (\text{III-350})$$

In order to enable the Kalman filter, estimating the constant tilt angle errors, some small portion of white noise must be added in order add some model uncertainty, which forces the Kalman filter to adapt the values, based on actual measurements. Without any white noise, the Kalman filter would consider the model as absolutely true and therefore, would not adapt the values.

The dynamic model used for the tracking filter, extended about the tilt angle errors, is given in the following equation.

$$\dot{\mathbf{z}}_{TS} = \mathbf{A}_{TS} \cdot \mathbf{z}_{TS} + \mathbf{n}_{TS}$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e^e \\ \dot{\hat{\mathbf{a}}}_{ee} \\ \delta \dot{\hat{\mathbf{v}}}_{e,R}^e \\ \delta \dot{\hat{\mathbf{v}}}_{e,N}^e \\ \Delta \dot{\hat{\Phi}} \\ \Delta \dot{\hat{\Theta}} \\ \mathbf{c} \cdot \dot{\hat{t}}_{clk} \\ \mathbf{c} \cdot \dot{\hat{d}}_{clk} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,R}^e & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 1 \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_e \\ \hat{\mathbf{v}}_e^e \\ \hat{\mathbf{a}}_{ee} \\ \delta \hat{\mathbf{v}}_{e,R}^e \\ \delta \hat{\mathbf{v}}_{e,N}^e \\ \Delta \hat{\Phi} \\ \Delta \hat{\Theta} \\ \mathbf{c} \cdot \hat{t}_{clk} \\ \mathbf{c} \cdot \hat{d}_{clk} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ \mathbf{n}_{\delta v,R,Aid} \\ \mathbf{n}_{\delta v,N,Aid} \\ n_{\Delta \hat{\Phi}} \\ n_{\Delta \hat{\Theta}} \\ n_{\delta tclk} \\ n_{\delta dclk} \end{bmatrix} \quad (\text{III-351})$$

Within the dynamic model, the tilt angle errors are not coupled. Instead they are used in the measurement equation to transform the estimated velocity in e-frame into the tilted n-frame of aiding velocity.

The nonlinear update equation gets

$$\begin{aligned} \mathbf{z}_{k+1}^+ &= \mathbf{z}_{k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tilde{\mathbf{r}} \\ \tilde{\mathbf{v}}_{e,Aid}^n - \mathbf{R}_{\tilde{n}} (\Delta \hat{\Phi}, \Delta \hat{\Theta}) \cdot \mathbf{R}_{ne} \cdot \hat{\mathbf{v}}_e^e \end{pmatrix} \\ &= \mathbf{z}_{k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tilde{\mathbf{r}} \\ \tilde{\mathbf{v}}_{e,Aid}^n - \mathbf{h}_{tilt} (\Delta \hat{\Phi}, \Delta \hat{\Theta}, \hat{\mathbf{v}}_e^e) \end{pmatrix} \end{aligned} \quad (\text{III-352})$$

For Kalman gain calculation, a linearized observation matrix is necessary.

$$\delta \hat{\mathbf{v}}_e^{\tilde{n}} = \left. \frac{\partial \mathbf{h}(\Delta \hat{\Phi}, \Delta \hat{\Theta}, \hat{\mathbf{v}}_e^e)}{\partial \Delta \hat{\Phi}} \right|_{\Delta \hat{\Phi}, \hat{\mathbf{v}}_e^e} \cdot \delta \Delta \hat{\Phi} + \left. \frac{\partial \mathbf{h}(\Delta \hat{\Phi}, \Delta \hat{\Theta}, \hat{\mathbf{v}}_e^e)}{\partial \Delta \hat{\Theta}} \right|_{\Delta \hat{\Theta}, \hat{\mathbf{v}}_e^e} \cdot \delta \Delta \hat{\Theta} \dots$$

$$\dots + \left. \frac{\partial \mathbf{h}(\Delta \hat{\Phi}, \Delta \hat{\Theta}, \hat{\mathbf{v}}_e^e)}{\partial \hat{\mathbf{v}}_e^e} \right|_{\Delta \hat{\Phi}, \Delta \hat{\Theta}} \cdot \delta \hat{\mathbf{v}}_e^e \quad (\text{III-353})$$

The linearized observation matrix, used for Kalman gain calculation gets:

$$\mathbf{H} = \begin{bmatrix} \left(\frac{\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e}{C \cdot \|\mathbf{x}_{Sv\#1} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & \frac{1}{C} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e}{C \cdot \|\mathbf{x}_{Sv\#Nsat} - \hat{\mathbf{x}}_e\|} \right)^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & \frac{1}{C} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{\tilde{r}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \mathbf{R}_{ne} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{\tilde{r}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \mathbf{R}_{ne} & \mathbf{R}_{\tilde{r}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta}) \cdot \mathbf{R}_{ne} & \boldsymbol{\Psi}_{\Delta\Phi} & \boldsymbol{\Psi}_{\Delta\Theta} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (\text{III-354})$$

With $\boldsymbol{\Psi}_{\Delta\Phi}$ and $\boldsymbol{\Psi}_{\Delta\Theta}$ are given below.

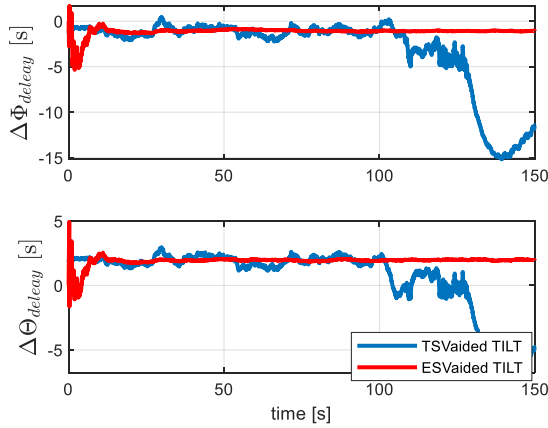
$$\boldsymbol{\Psi}_{\Delta\Phi} = \frac{\partial \mathbf{R}_{\tilde{r}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta})}{\partial \Delta\hat{\Phi}} \cdot \mathbf{R}_{ne} \cdot \mathbf{v}_e^e$$

$$\boldsymbol{\Psi}_{\Delta\Theta} = \frac{\partial \mathbf{R}_{\tilde{r}n}(\Delta\hat{\Phi}, \Delta\hat{\Theta})}{\partial \Delta\hat{\Theta}} \cdot \mathbf{R}_{ne} \cdot \mathbf{v}_e^e$$

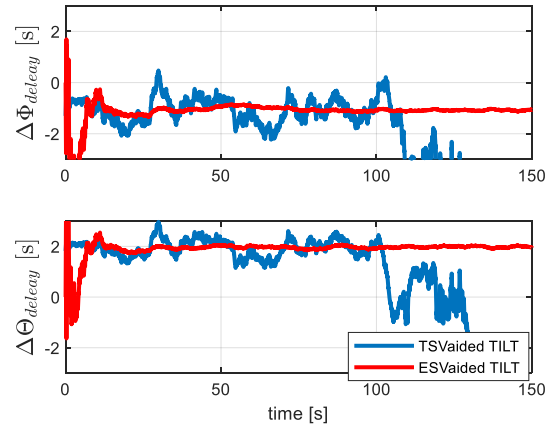
(III-355)

Equivalent base band simulations show that in general it is also possible to estimate tilt aiding errors using a total state vector tracking approach. But tuning is very sensitive for getting stable estimation results. In case of higher dynamic, the tilt error estimation becomes even unstable as the following simulation shows. The reason is given by the non-linearity of the observation matrix.

For comparison, the following figures show the estimated tilt angles of the total state vector tracking approach together with the much better tilt angle estimation of the error state vector tracking approach.



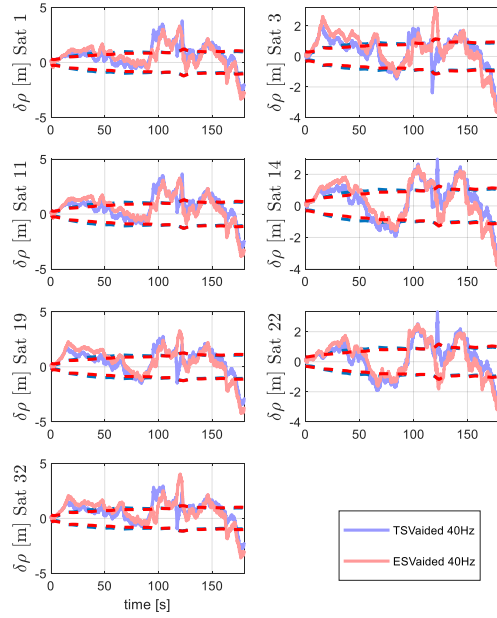
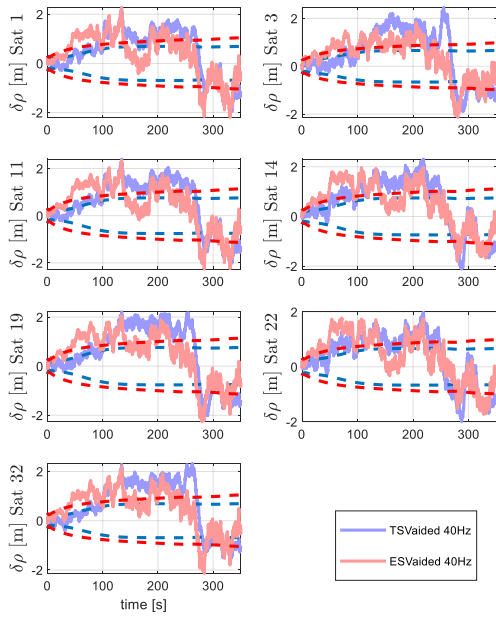
III-229 Estimated tilt angles - DA42 approach



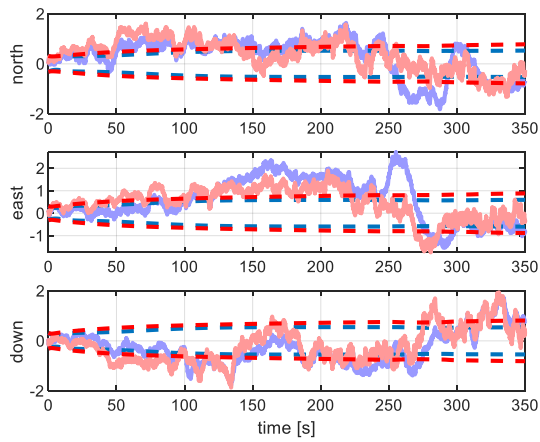
III-230 Estimated tilt angles - DA42 approach - zoomed

III - 11.6 Aided total state vector tracking vs. aided error state vector tracking

UNJAMMED

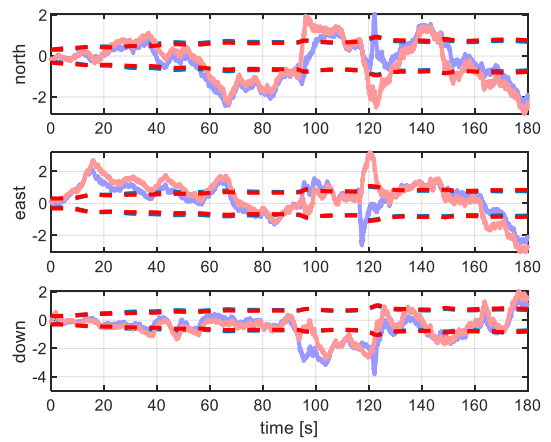


III-231 pseudorange error - DA42 - unjammed

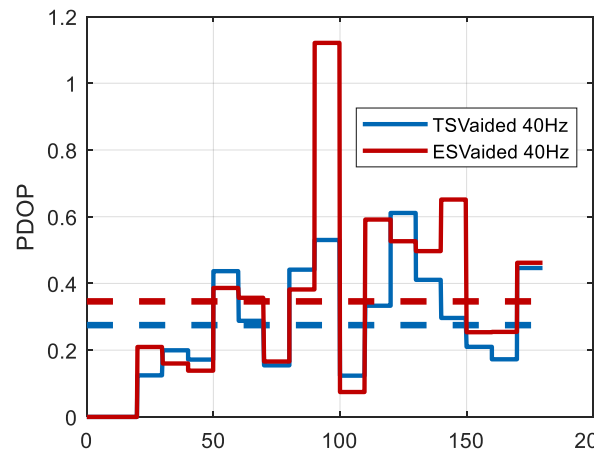
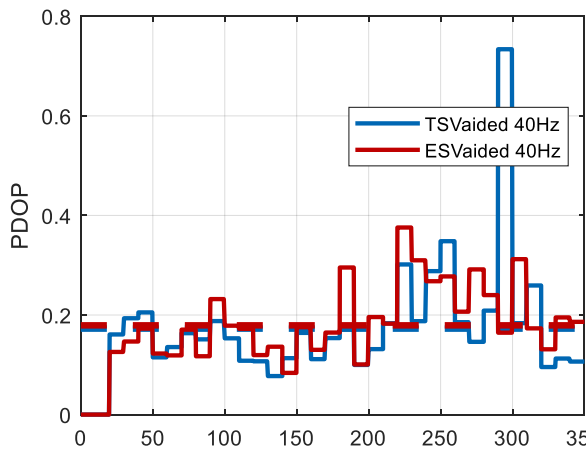
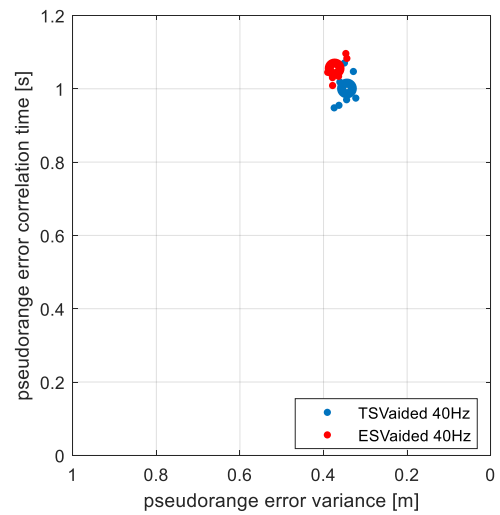
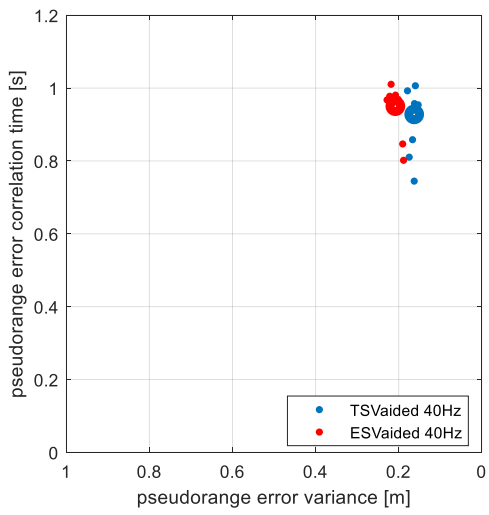


III-233 positioning error - DA42 - unjammed

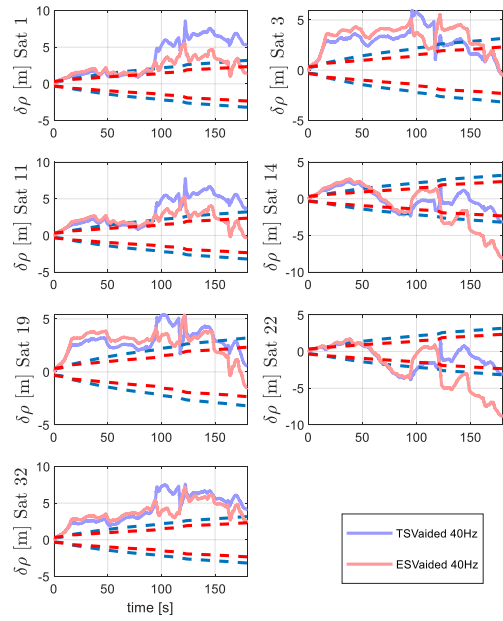
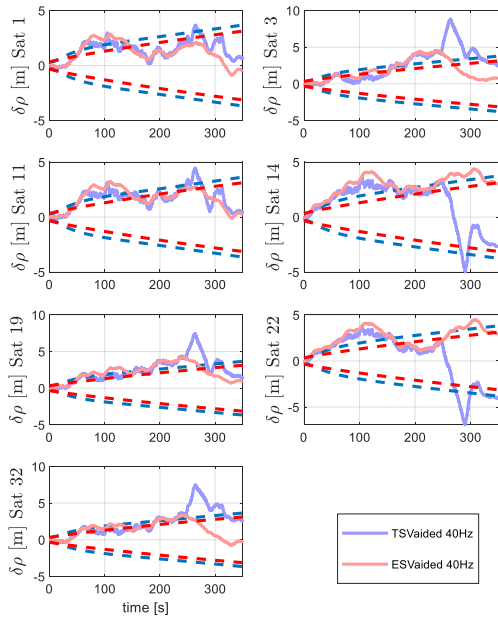
III-232 pseudorange error - high dynamic - unjammed



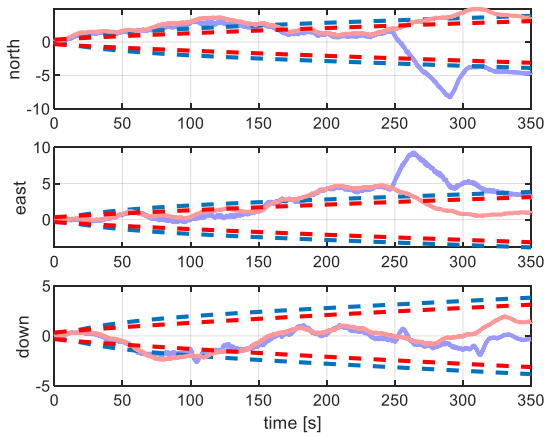
III-234 positioning error - high dynamic - unjammed



JAMMED

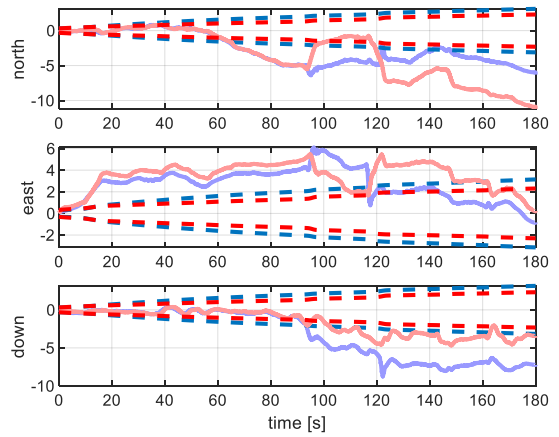


III-239 pseudorange error - DA42 - jammed

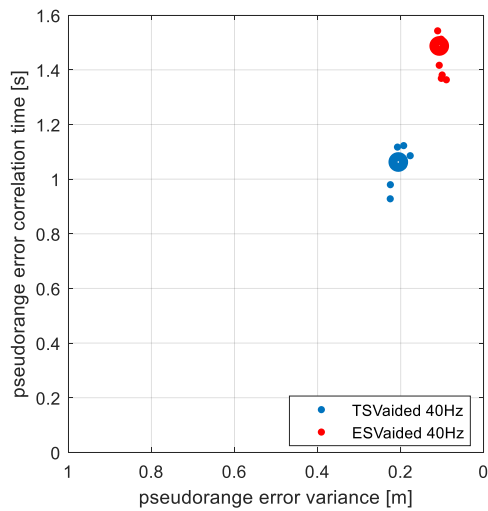


III-241 positioning error - DA42 - jammed

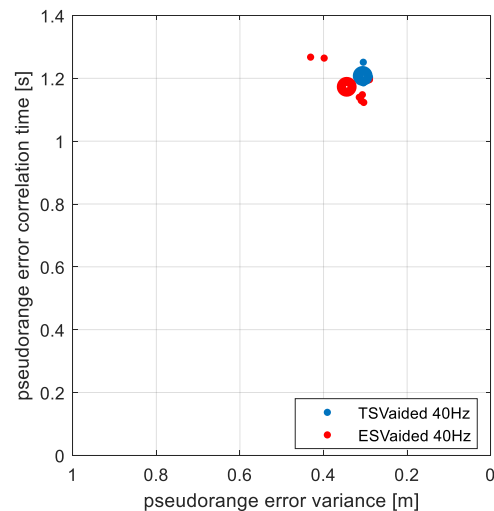
III-240 pseudorange error - high dynamic - jammed



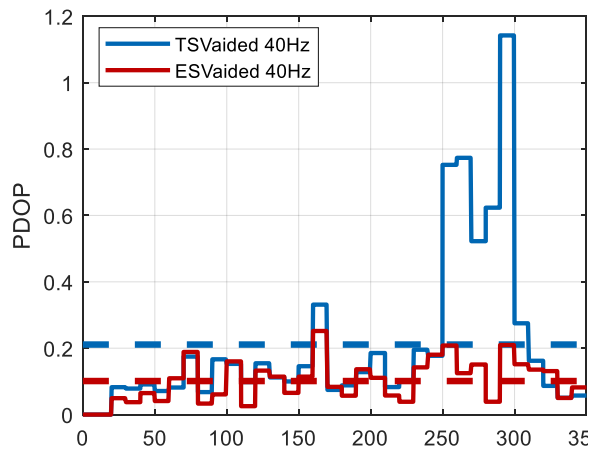
III-242 positioning error - high dynamic - jammed



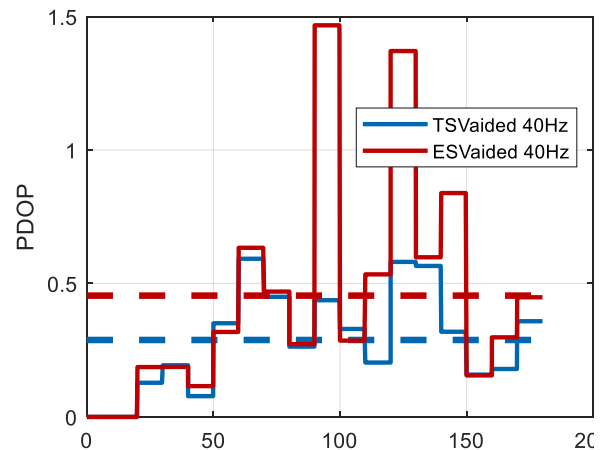
III-243 Comparing raw date - DA42 - jammed



III-244 Comparing raw date - high dynamic - jammed



III-245 PDOP - DA42 - jammed



III-246 PDOP - high dynamic - jammed

Conclusion: TSV aided against ESV aided

If there are only noise like aiding errors and rate aiding errors, the aided total state approach gives similar results as the aided error state approach.

Differences arise in case of aiding delays or tilt aiding errors.

Aiding delay can be estimated even with the total state approach, but worse, compared to error state approach.

The estimation of tilt aiding errors respectively tilt angles is in general possible in case of aided total state vector tracking. But the estimation is very instable and also the corresponding tuning is very sensitive. The dynamics within the total state filter are too restless for an appropriate tilt error estimation, respectively for mapping the tilt error parts within the discriminator measurements on the right state. As a conclusion, for aiding, always an error state vector tracking architecture should be used.

III - 12 Aided scalar state tracking

III - 12.1 Motivation

In the previous sections, aided vector tracking architectures were considered. This section develops an aided scalar tracking loop. Only a total state aided scalar tracking architecture is developed and analyzed. An error state aided scalar approach will not be considered, because it would need a cascade of three Kalman filters. The aided scalar total state approach already needs a cascaded Kalman filter structure and thus a mutual coupling. Already here, the outputs of the first stage – being the total state scalar filters – are correlated in time. This correlation must be considered in the superior positioning filter. In case of the error state approach, one coupling stage more would exist, causing a too strong temporal correlation.

Aided scalar tracking is a decentralized approach. There are local tracking filters for each satellite and a superior positioning filter. In [77] an unaided conjoint architecture of scalar and vector tracking filters are given. In this section, a similar conjoint approach for an aided scalar tracking is developed.

III - 12.2 Tracking architecture

For scalar tracking, always the combination of scalar tracking loops for each satellite are necessary, together with a subsequent positioning filter. The scalar tracking filter as well as the positioning filter are realized by using optimal filters.

This section develops an aided scalar tracking architecture by using the combination of a total state scalar tracking filter, where aiding is used as an additional measurement. The positioning solution is derived by a subsequent total state Kalman filter.

After deriving the tracking respectively filter equations, the reasons are explained, why especially this combination is chosen.

In this first section on aided scalar tracking, only rate aiding errors and noise aiding error are considered.

The following equation gives the total state dynamic model of the used Kalman filter for a single satellite channel.

$$\dot{\mathbf{z}}_{LoS} = \mathbf{A}_{LoS} \cdot \mathbf{z}_{LoS} + \mathbf{n}_{LoS}$$

$$\begin{bmatrix} \dot{\hat{\rho}}_{LoS} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \\ \delta \dot{\mathbf{v}}_{e,LoS,N}^e \\ \delta \dot{\mathbf{v}}_{e,LoS,R}^e \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \\ 0 & 0 & 1 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \\ 0 & 0 & \frac{-1}{\tau_{a,LoS}} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}^{3 \times 3} \\ \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 3} & \mathbf{A}_{\delta v,R}^e \end{bmatrix} \cdot \begin{bmatrix} \hat{\rho}_{LoS} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \\ \delta \mathbf{v}_{e,LoS,N}^e \\ \delta \mathbf{v}_{e,LoS,R}^e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \\ \mathbf{n}_{\delta vN,LoS} \\ \mathbf{n}_{\delta vR,LoS} \end{bmatrix} \quad (III-356)$$

The line of sight dynamic is modeled again as a second order dynamic, using a constant mean, Gauss Markov like acceleration process.

The noise like aiding errors are modeled as a Gauss Markov process $\delta\hat{\mathbf{v}}_{e,LoS,N}^e$. The characteristics of the noise like aiding error are given in n-frame, whereas the noise like aiding error is needed in e-frame coordinates. Therefore, equivalent to equation (III-265), a Gauss Markov process is developed, being able to transform within the differential equations, the n-frame noise characteristics into an e-frame Gauss Markov process. The noise like aiding error is considered within the observation equation. Thus, given very bad aiding quality with high noise, the aiding will be less considered, respectively the bad aiding is prevented from entering the tracking loop.

The rate aiding error $\delta\hat{\mathbf{v}}_{e,LoS,R}^e$ is also modeled as a Gauss Markov process. But in contrast to the noise like aiding error, the rate aiding error is mapped within the dynamic matrix \mathbf{A}_{LoS} onto to the pseudorange, by using the line of sight direction vector $\mathbf{e}_{LoS}^T(\mathbf{x}_{\#Sv}, \hat{\mathbf{x}})$.

Given high rate aiding error variances, respectively low aiding rates, the dynamic model becomes more uncertain and the Kalman filter puts more weight on the measurements. Due to the mapping on the pseudorange, only the estimated pseudoranges becomes more uncertain, the estimated line of sight velocity remains at the given uncertainty. That's why the Kalman filter puts only more weight on the discriminator measurements, while keeping the weights on the aiding inputs at the same level. Putting more weights on the discriminator measurements is the correct behavior, because the discriminator provides the necessary pseudorange correction, caused by low aiding rates.

If the rate aiding error would be considered within the observation equation, high rate aiding error variances would force the Kalman filter, to put less weight on the aiding input. But the weights on the discriminator measurement would remain at the same level. Given low aiding rates, aiding cannot cover higher dynamics. The Kalman filter must put higher weights on the discriminator measurements, which is reached by the coupling within the state dynamic matrix.

The following equation shows the Kalman filter update equation.

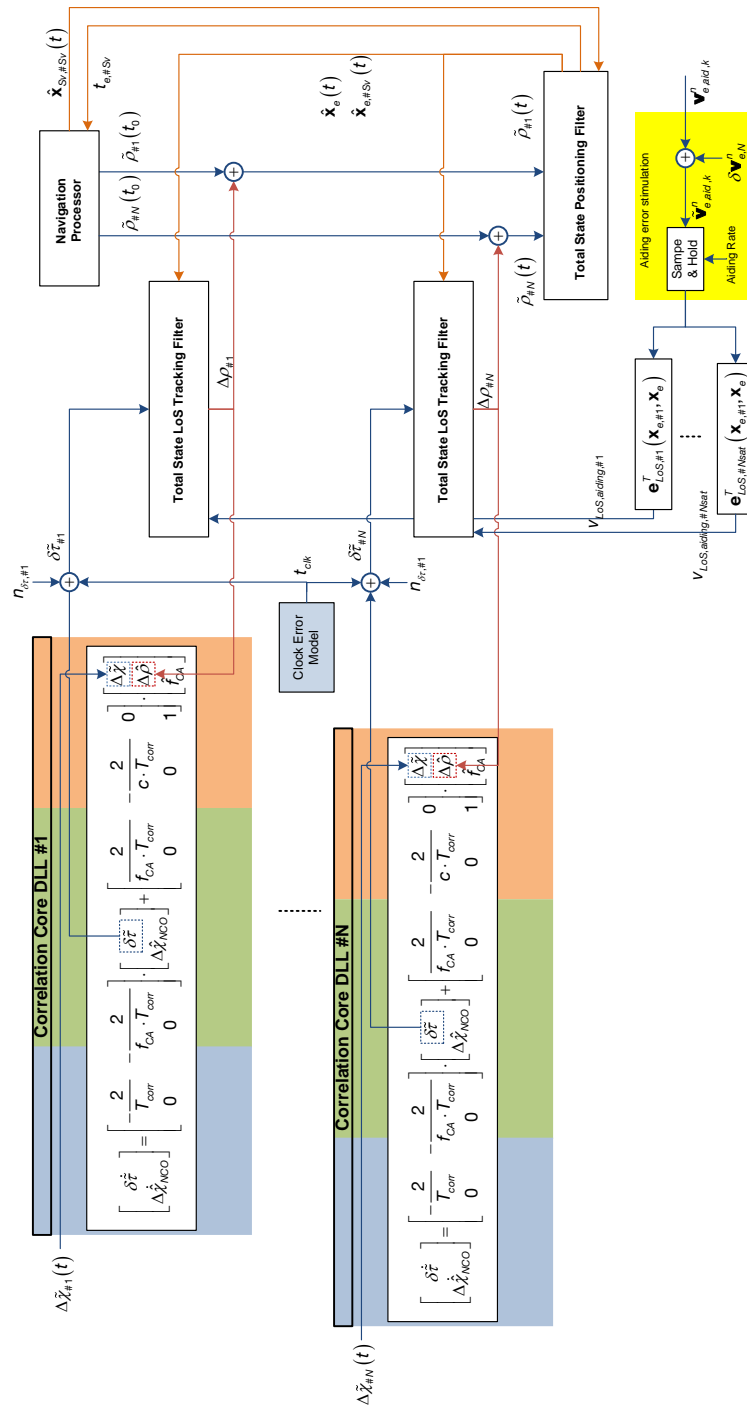
$$\mathbf{z}_{LoS,k+1}^+ = \mathbf{z}_{LoS,k+1}^- + \mathbf{K}_{LoS} \cdot \begin{pmatrix} \delta\tau \\ \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \cdot (\mathbf{v}_{e,Sv}^e - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,aiding}^n) - \hat{V}_{LoS} \end{pmatrix} \quad (III-357)$$

The estimated line of sight velocity \hat{V}_{LoS} contains the receiver velocity mapped onto the line of sight vector and the satellite velocity, also mapped onto the line of sight vector. The external aiding velocity $\tilde{\mathbf{v}}_{e,aiding}^n$ contains only the receiver velocity. In order to get the aided line of sight velocity, also the actual satellite velocity must be mapped onto the line of sight vector and added with the appropriate sign. The actual satellite velocity $\mathbf{v}_{e,Sv}^e$ is available through the ephemeris. The ephemeris must be evaluated at the signal transmission time t_{Sv} . This transmission time is available from the positioning filter.

$$\mathbf{H}_{LoS} = \begin{bmatrix} \frac{1}{c} & 0 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} \\ 0 & 1 & 0 & \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) & \mathbf{0}^{1 \times 3} \end{bmatrix} \quad (\text{III-358})$$

The observation matrix \mathbf{H}_{LoS} includes the mapping of the noise like aiding error, whereas within the update equation, the noise like aiding error is not mapped onto the estimated velocity. That's why the noise like aiding error is only realized as a consider state.

The following figure shows the scalar aided tracking architecture.



III-247 Scalar aided tracking architecture

As the architecture shows, the scalar tracking filter needs the satellite position and the actual estimated receiver position for mapping the e-frame noise like aiding error onto the line of sight dynamic.

The total state positioning solution is calculated by using a total state Kalman filter. The corresponding dynamic model and update equation are given below.

$$\dot{\mathbf{z}}_{POS} = \mathbf{A}_{POS} \cdot \mathbf{z}_{POS} + \mathbf{n}_{POS}$$

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_e \\ \dot{\hat{\mathbf{v}}}_e \\ \dot{\hat{\mathbf{a}}}_e^{ee} \\ C \cdot \dot{\hat{t}}_{clk} \\ C \cdot \dot{\hat{d}}_{clk} \\ \delta \hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{A}_{\delta p} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ n_{tclk} \\ n_{dclk} \\ \mathbf{n}_{\delta p} \end{bmatrix} \quad (\text{III-359})$$

The measurements of the positioning filter are the pseudoranges, which are states of the independent scalar tracking filters.

The pseudorange errors and therefore the measurement errors of the positioning filter are highly correlated in time due to the aiding of the scalar tracking loop.

The already derived positioning solution in section III - 6 has a similar structure and uses as measurements also pseudoranges from independent scalar tracking filters. But there, the pseudorange errors are much less correlated, because there is no aiding and therefore a higher closed loop bandwidth. The applied velocity aiding in this section leads to very low tracking bandwidth of the scalar tracking filter, which in turn causes long correlated pseudorange errors.

In order to consider this effect in the positioning solution, the measurement errors are modeled as long correlated Gauss Markov processes by $\delta \hat{\mathbf{p}}$.

$$\mathbf{A}_{\delta p} = \begin{bmatrix} \frac{1}{\tau_{\delta p, \#1}} & 0 & \dots & 0 \\ 0 & \frac{1}{\tau_{\delta p, \#2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\tau_{\delta p, \#N_{sat}}} \end{bmatrix} \quad (\text{III-360})$$

The state error covariance matrix is given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{\text{sat}}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{\text{sat}}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{Q}_a & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{\text{sat}}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & c^2 \cdot \sigma_{\text{clk}}^2 & 0 & \mathbf{0}_{1 \times N_{\text{sat}}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & c^2 \cdot \sigma_{\text{dclk}}^2 & \mathbf{0}_{1 \times N_{\text{sat}}} \\ \mathbf{0}_{N_{\text{sat}} \times 3} & \mathbf{0}_{N_{\text{sat}} \times 3} & \mathbf{0}_{N_{\text{sat}} \times 3} & \mathbf{0}_{N_{\text{sat}} \times 1} & \mathbf{0}_{N_{\text{sat}} \times 1} & \mathbf{Q}_{\delta\rho} \end{bmatrix}$$

$$\mathbf{Q}_a = \begin{bmatrix} \frac{2 \cdot \sigma_{ax}^2}{\tau_{ax}} & 0 & 0 \\ 0 & \frac{2 \cdot \sigma_{ay}^2}{\tau_{ay}} & 0 \\ 0 & 0 & \frac{2 \cdot \sigma_{az}^2}{\tau_{az}} \end{bmatrix} \quad \mathbf{Q}_{\delta\rho} = \begin{bmatrix} \frac{2 \cdot p_{\delta\rho, \text{LoS}, \#1}}{\tau_{\delta\rho, \#1}} & 0 & \dots & 0 \\ 0 & \frac{2 \cdot p_{\delta\rho, \text{LoS}, \#2}}{\tau_{\delta\rho, \#2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{2 \cdot p_{\delta\rho, \text{LoS}, \#N_{\text{sat}}}}{\tau_{\delta\rho, \#N_{\text{sat}}}} \end{bmatrix} \quad (\text{III-361})$$

For tuning, scenario matched tuning is used. The values $p_{\delta\rho, \text{LoS}, \#Sv}$ are taken from the state error covariance matrix, available in each scalar tracking filter.

The Kalman filter update equation of the positioning solution gets

$$\mathbf{z}_{k+1, \text{POS}}^+ = \mathbf{z}_{k+1, \text{POS}}^- + \mathbf{K}_{\text{POS}} \cdot (\tilde{\mathbf{p}} - \mathbf{h}_{\text{POS}}(\mathbf{z}_{k+1, \text{POS}}^-)) \quad (\text{III-362})$$

the innovation $\tilde{\mathbf{p}} - \mathbf{h}_{\text{POS}}(\mathbf{z}_{k+1, \text{POS}}^-)$ is calculated using the nonlinear relationship.

The nonlinear transformation from the filter states to the observation space is given by the following equation.

$$\delta\tilde{\mathbf{p}} = \tilde{\mathbf{p}} - \mathbf{h}_{\text{POS}}(\mathbf{z}_{k+1, \text{POS}}^-) = \tilde{\mathbf{p}} - \hat{\mathbf{p}}$$

$$= \begin{bmatrix} \tilde{p}_{\#1} \\ \vdots \\ \tilde{p}_{\#n} \end{bmatrix} - \begin{bmatrix} \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1, e'}(t_{e, \#1}) - \mathbf{x}_e(t_r)) + c \cdot \delta t_{\text{clk}} - c \cdot \delta t_{\#1}} \\ \vdots \\ \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \mathbf{x}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n, e'}(t_{e, \#n}) - \mathbf{x}_e(t_r)) + c \cdot \delta t_{\text{clk}} - c \cdot \delta t_{\#n}} \end{bmatrix} \quad (\text{III-363})$$

In order to set up the Kalman filter observation matrix \mathbf{H}_{POS} , the nonlinear transformation $\mathbf{h}_{\text{POS}}(\mathbf{z}_{k+1, \text{POS}}^-)$ is linearized at the every point on the trajectory.

The linearized observation equation gets

$$\mathbf{H}_{\rho OS} = \begin{bmatrix} \mathbf{h}_{\rho, \#1}(\hat{\mathbf{x}}_e) & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 1 & 1 & \mathbf{h}_{\tilde{\rho}, \#1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{h}_{\rho, \#Sv}(\hat{\mathbf{x}}_e) & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 1 & 1 & \mathbf{h}_{\tilde{\rho}, \#Sv} \end{bmatrix} \quad (\text{III-364})$$

$$\mathbf{h}_{\tilde{\rho}, \#1} = [1 \ 0 \ \dots \ 0] \quad (\text{III-365})$$

Within the linearized observation equation, the pseudorange errors are considered, but not in the real feedback calculation.

Calculating the estimated pseudorange requires the corresponding satellite position in the e-frame at time of signal reception in the receiver. For that, an iterative procedure is necessary, given in section III - 7 .

A comparison of the aided scalar approach with the aided vector tracking architectures will be given at the end of this chapter.

III - 12.3 Aiding delay estimation

Up to now, only noise like aiding errors and aiding errors caused by low aiding rates are considered. This section develops a method to estimate and also compensate aiding errors, caused by delayed aiding information.

Generally, the delay error is caused by transmission delays and can therefore be found in almost every navigation architecture. The delay error can be written as:

$$\delta \mathbf{v}_{e, OOS}^e(t) = \mathbf{v}_e^e(t) - \mathbf{v}_e^e(t - \Delta t_{OOS}) \quad (\text{III-366})$$

The corresponding line of sight delay velocity error gets

$$\delta v_{LoS, OOS}(t) = \mathbf{e}_{LoS, Sv}^T \cdot \delta \mathbf{v}_{e, OOS}^e(t) = \mathbf{a}_{LoS} \cdot \Delta t_{OOS} \quad (\text{III-367})$$

Equation (III-367) shows that for estimating the delay error Δt_{OOS} , the line of sight acceleration is necessary.

The delay is modeled as a constant value within the line of sight state dynamic model.

$$\dot{\mathbf{z}}_{LoS} = \mathbf{A}_{LoS} \cdot \mathbf{z}_{LoS} + \mathbf{n}_{LoS}$$

$$\begin{bmatrix} \dot{\hat{\rho}}_{LoS} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \\ \delta \dot{\hat{\mathbf{v}}}_{e,LoS,N}^e \\ \delta \dot{\hat{\mathbf{v}}}_{e,LoS,R}^e \\ \Delta \dot{\hat{t}}_{oos} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) & \hat{a}_{LoS} \\ 0 & 0 & 1 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 \\ 0 & 0 & \frac{-1}{\tau_{a,LoS}} & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 \\ \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{A}_{\delta v,N}^e & \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 1} \\ \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 1} & \mathbf{0}^{3 \times 3} & \mathbf{A}_{\delta v,R}^e & \mathbf{0}^{3 \times 1} \\ 0 & 0 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\rho}_{LoS} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \\ \delta \hat{\mathbf{v}}_{e,LoS,N}^e \\ \delta \hat{\mathbf{v}}_{e,LoS,R}^e \\ \Delta \hat{t}_{oos} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_{a,LoS} \\ \mathbf{n}_{\delta vN,LoS} \\ \mathbf{n}_{\delta vR,LoS} \\ n_{\Delta toos} \end{bmatrix} \quad (III-368)$$

The coupling of the delay is realized within the state dynamic matrix, marked by the red entry. Due to this coupling, the state dynamic matrix becomes nonlinear.

The Kalman filter update equation gets

$$\mathbf{z}_{LoS,k+1}^+ = \mathbf{z}_{LoS,k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tau \\ \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \cdot (\mathbf{v}_{e,Sv}^e - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,aiding}^n) - \hat{v}_{LoS} \end{pmatrix} \quad (III-369)$$

with the corresponding observation matrix

$$\mathbf{H}_{LoS} = \begin{bmatrix} \frac{1}{c} & 0 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 \\ 0 & 1 & 0 & \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) & \mathbf{0}^{1 \times 3} & 0 \end{bmatrix} \quad (III-370)$$

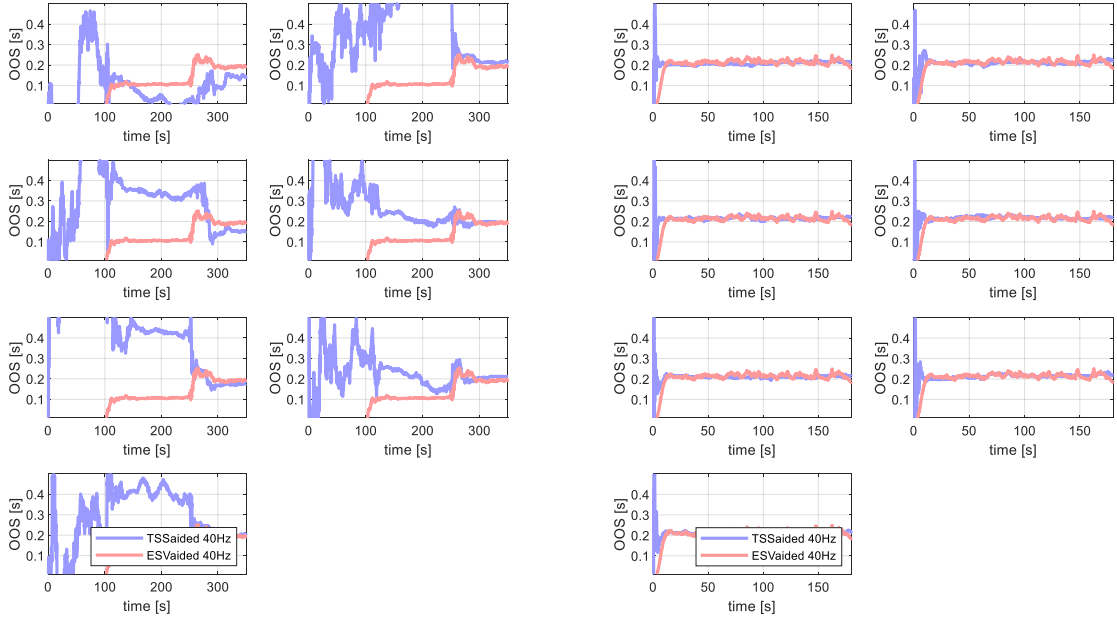
An alternative approach is, coupling the delay within the observation equation instead of the system dynamic model. The system dynamic matrix would in this case be linear, instead the observation matrix \mathbf{H}_{LoS} would become nonlinear, as shown below.

$$\mathbf{z}_{LoS,k+1}^+ = \mathbf{z}_{LoS,k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta \tau \\ \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \cdot (\mathbf{v}_{e,Sv}^e - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,aiding}^n) - (\hat{v}_{LoS} - \hat{a}_{LoS} \cdot \Delta \hat{t}_{oos}) \end{pmatrix} \quad (III-371)$$

$$\mathbf{H}_{LoS} = \begin{bmatrix} \frac{1}{c} & 0 & 0 & \mathbf{0}^{1 \times 3} & \mathbf{0}^{1 \times 3} & 0 \\ 0 & 1 & 0 & \mathbf{e}_{LoS}^T(\mathbf{x}_{Sv}, \hat{\mathbf{x}}) & \mathbf{0}^{1 \times 3} & -\hat{a}_{LoS} \end{bmatrix}$$

An observability analysis of the last realization shows that the delay error is not observable. The reason is that there is no coupling of delay error with the pseudorange and therefore the discriminator error measurement. Only the discriminator error measurement poses the needed constraint to make the aiding delay observable.

The following two figures compare aiding delay estimation of the aided ESV and aided TSS architectures.



III-248 Estimation "aiding delay" - DA42

III-249 Estimation "aiding delay" - High Dynamic

The DA42 scenario shows that an estimation of the delay is only possible if there is some dynamic. Both architectures are able to estimate the delay error. In case of the aided scalar approach, every scalar tracking filter estimates the delay. In case of aided error state vector approach, there is only one estimation of the aiding delay.

The simulations show that the aided error state vector approach shows better aiding delay estimation performance in the selected examples. Important to notice is that the estimation performance depends on the selected tuning parameter and also the special scenario. Therefore, the given examples are not valid in general.

III - 12.4 Tilt error estimation

Each scalar tracking loop is aided separately within the local update equation of the scalar tracking according to equation (III-357), by mapping the n-frame aiding velocity onto the corresponding line of sight vector.

$$v_{LoS,Aiding} = \mathbf{e}_{LoS}^T (\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \cdot \mathbf{v}_{e,Sv}^e - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,aiding}^n \quad (III-372)$$

Within the local scalar tracking loop, there is no possibility to estimate tilt aiding errors, respectively tilt angles. Through the Kalman filter update equation

$$\mathbf{z}_{LoS,k+1}^+ = \mathbf{z}_{LoS,k+1}^- + \mathbf{K}_{LoS} \cdot \left(\begin{array}{c} \delta\tau \\ \mathbf{e}_{LoS}^T (\mathbf{x}_{Sv}, \hat{\mathbf{x}}) \cdot (\mathbf{v}_{e,Sv}^e - \mathbf{R}_{en} \cdot \tilde{\mathbf{v}}_{e,aiding}^n) - \hat{v}_{LoS} \end{array} \right) \quad (III-373)$$

the pseudoranges become false due to the wrong transformation of aiding velocity into e-frame. The discriminator measurement $\delta\tau$ would give the needed correction. Because the tilt aiding error is not modeled, the Kalman filter does not fully correct the line of sight states. Part's of the tilt aiding error are within the estimated pseudorange states.

$$\hat{\rho} = \rho + \mathbf{c} \cdot \Delta \hat{t}_{clk} + \delta\rho_{tilt} + n_{\delta\rho} + \dots \quad (\text{III-374})$$

As positioning filter, an error state structure will be used.

The error state dynamic model is given as

$$\delta \dot{\mathbf{z}}_{POS} = \mathbf{A}_{\delta z} \cdot \delta \mathbf{z}_{POS} + \mathbf{n}_{POS} \quad (\text{III-375})$$

$$\begin{bmatrix} \delta \dot{\mathbf{x}}_e \\ \delta \dot{\mathbf{v}}_e^e \\ \delta \dot{\mathbf{a}}_e^{ee} \\ \mathbf{c} \cdot \delta \dot{\hat{t}}_{clk} \\ \mathbf{c} \cdot \delta \dot{\hat{d}}_{clk} \\ \delta \Delta \dot{\hat{\Phi}} \\ \delta \Delta \dot{\hat{\Theta}} \\ \delta \dot{\hat{\rho}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \boldsymbol{\Psi}_{\Delta \hat{\Phi}} & \boldsymbol{\Psi}_{\Delta \hat{\Theta}} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}_a & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 & 0 & 0 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 & \mathbf{0}_{1 \times N_{sat}} \\ \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 3} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{0}_{N_{sat} \times 1} & \mathbf{A}_{\delta\rho} \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{x}_e \\ \delta \mathbf{v}_e^e \\ \delta \mathbf{a}_e^{ee} \\ \mathbf{c} \cdot \delta \hat{t}_{clk} \\ \mathbf{c} \cdot \delta \hat{d}_{clk} \\ \delta \Delta \hat{\Phi} \\ \delta \Delta \hat{\Theta} \\ \delta \hat{\rho} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{n}_a \\ n_{tclk} \\ n_{dclk} \\ n_{\Delta\Phi} \\ n_{\Delta\Theta} \\ \mathbf{n}_{\delta\rho} \end{bmatrix}$$

The terms $\boldsymbol{\Psi}_{\Delta \hat{\Phi}}$ and $\boldsymbol{\Psi}_{\Delta \hat{\Theta}}$, being used to couple the tilt angles with the positional error.

The Kalman filter update equation gets

$$\delta \mathbf{z}_{POS,k+1}^+ = \delta \mathbf{z}_{POS,k+1}^- + \mathbf{K} \cdot \begin{pmatrix} \delta\rho_{\#1} \\ \vdots \\ \delta\rho_{\#N_{sat}} \end{pmatrix} \quad (\text{III-376})$$

A total state model integrates the error states.

$$\dot{\mathbf{z}}_{POS} = \mathbf{A}_z \cdot \mathbf{z}_{POS} \quad (\text{III-377})$$

$$\begin{bmatrix} \dot{\mathbf{x}}_e \\ \mathbf{c} \cdot \dot{\hat{t}}_{clk} \\ \mathbf{c} \cdot \dot{\hat{d}}_{clk} \\ \Delta \dot{\hat{\Phi}} \\ \Delta \dot{\hat{\Theta}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_e \\ \mathbf{c} \cdot \hat{t}_{clk} \\ \mathbf{c} \cdot \hat{d}_{clk} \\ \Delta \hat{\Phi} \\ \Delta \hat{\Theta} \end{bmatrix}$$

The measurements $\delta\rho_{\#Sv}$ of the error state Kalman filter are calculated separately out of the difference between the estimated pseudoranges provided by the scalar tracking loops and the estimated pseudoranges, calculated out of the estimated total state position.

$$\begin{bmatrix} \delta\rho_{\#1} \\ \vdots \\ \delta\rho_{\#Nsat} \end{bmatrix} = \begin{bmatrix} \hat{\rho}_{\#1} \\ \vdots \\ \hat{\rho}_{\#Nsat} \end{bmatrix} - \begin{bmatrix} \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e'}(t_{e,\#1}) - \hat{\mathbf{x}}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#1}) \cdot \mathbf{x}_{\#1,e'}(t_{e,\#1}) - \hat{\mathbf{x}}_e(t_r)) + c \cdot \hat{t}_{clk}} \\ \vdots \\ \sqrt{(\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e'}(t_{e,\#n}) - \hat{\mathbf{x}}_e(t_r))^T \cdot (\mathbf{C}_{e'}^e(\tau_{\#n}) \cdot \mathbf{x}_{\#n,e'}(t_{e,\#n}) - \hat{\mathbf{x}}_e(t_r)) + c \cdot \hat{t}_{clk}} \end{bmatrix} \quad (\text{III-378})$$

The total states are updated by the error states

$$\begin{aligned} \hat{\mathbf{x}}_e &= \hat{\mathbf{x}}_e + \delta\hat{\mathbf{x}}_e \\ \hat{t}_{clk} &= \hat{t}_{clk} + \delta\hat{t}_{clk} \\ \hat{d}_{clk} &= \hat{d}_{clk} + \delta\hat{d}_{clk} \\ \Delta\hat{\Phi} &= \Delta\hat{\Phi} + \delta\Delta\hat{\Phi} \\ \Delta\hat{\Theta} &= \Delta\hat{\Theta} + \delta\Delta\hat{\Theta} \end{aligned} \quad (\text{III-379})$$

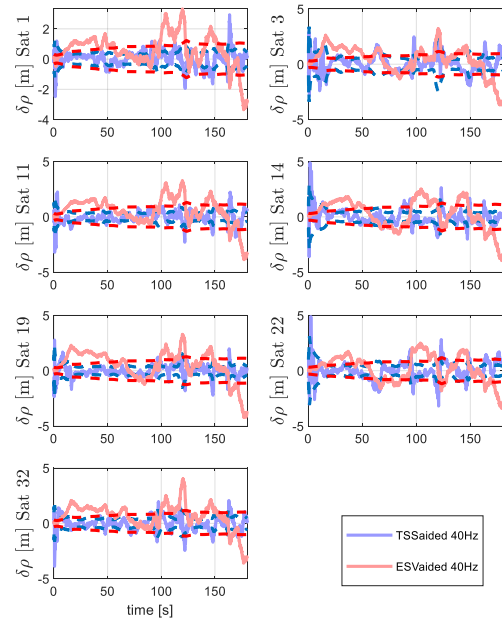
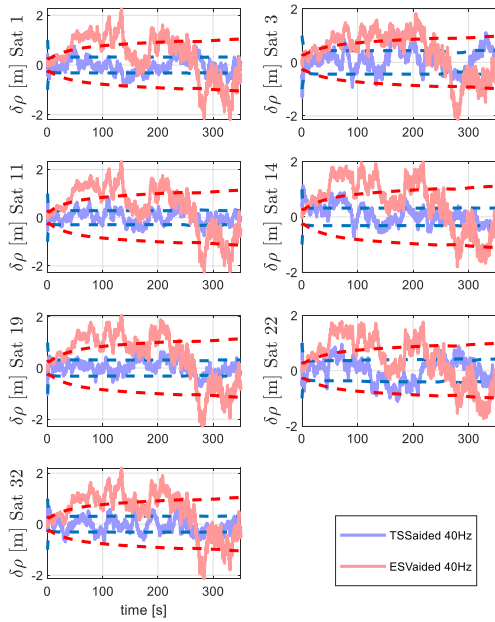
Simulations show that using this approach, no estimation of tilt angles is possible. The reason is, that within the estimated pseudoranges, provided by the independent aided scalar tracking loops, only parts of the tilt aiding error are included. Partly the aiding errors are already corrected by the discriminator measurements.

An estimation of tilt aiding errors within the independent scalar tracking loops is also not possible, because there, the tilt angles are not observable.

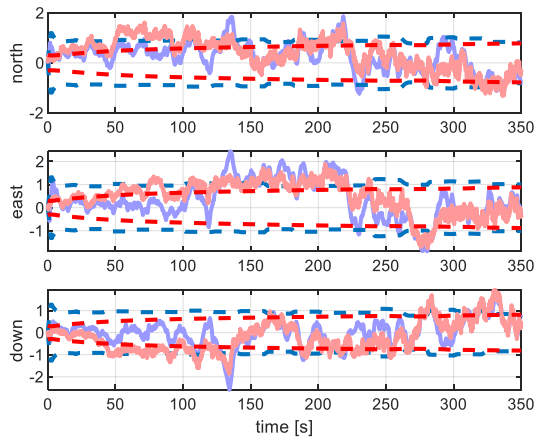
III - 12.5 Aided scalar tracking vs. aided vector tracking

This section compares the aided scalar tracking architecture with the aided error state vector tracking architecture. Both architectures will be compared for aiding without tilt aiding errors, because the scalar aided architecture is not able to estimate them.

UNJAMMED

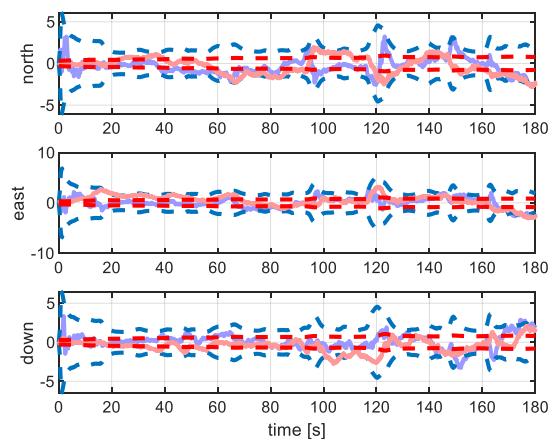


III-250 pseudorange error - DA42 - unjammed

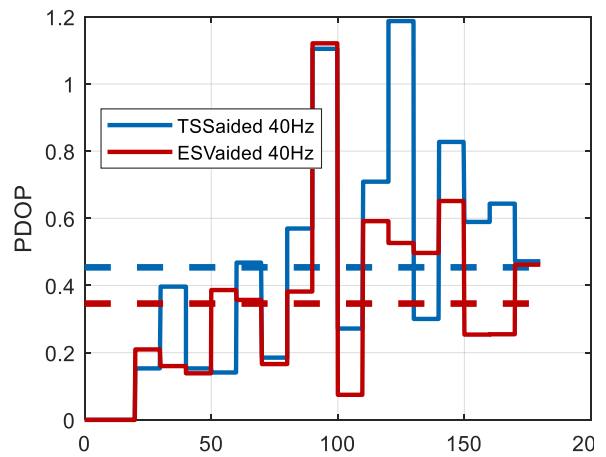
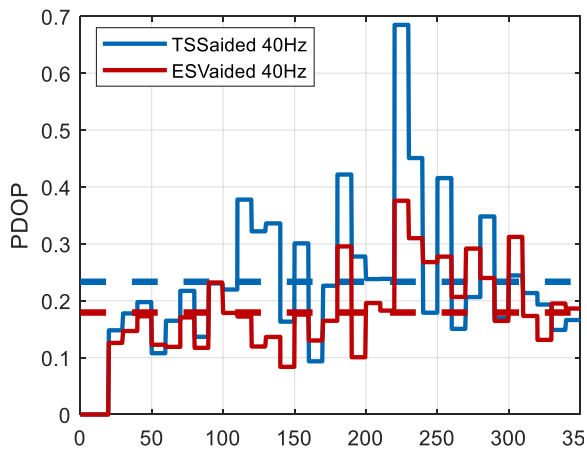
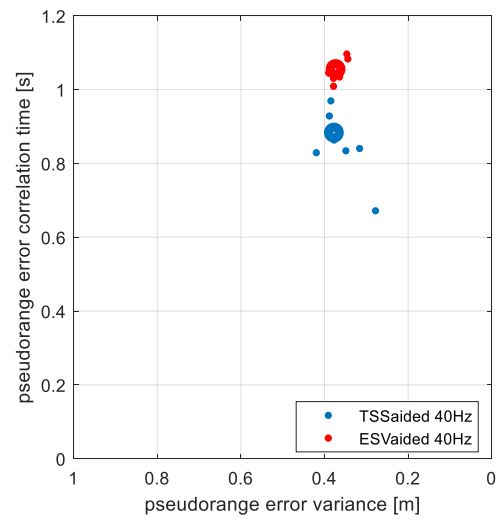
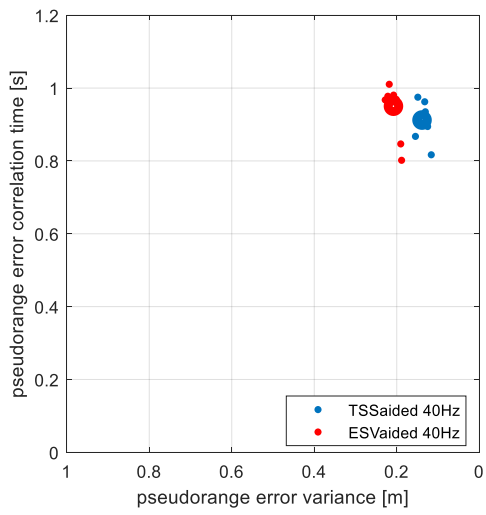


III-252 positioning error - DA42 - unjammed

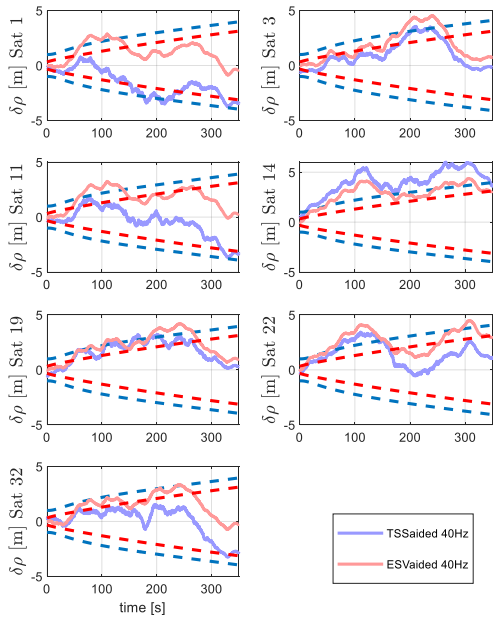
III-251 pseudorange error - high dynamic - unjammed



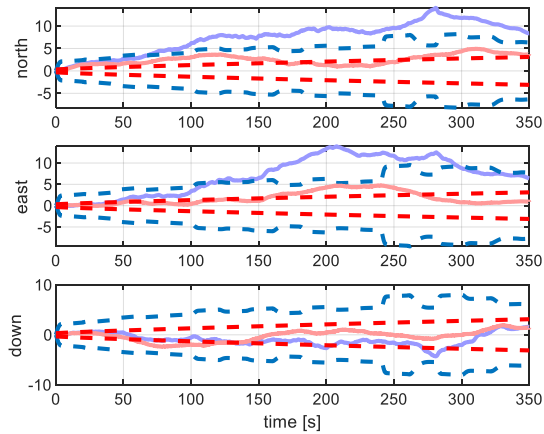
III-253 positioning error - high dynamic - unjammed



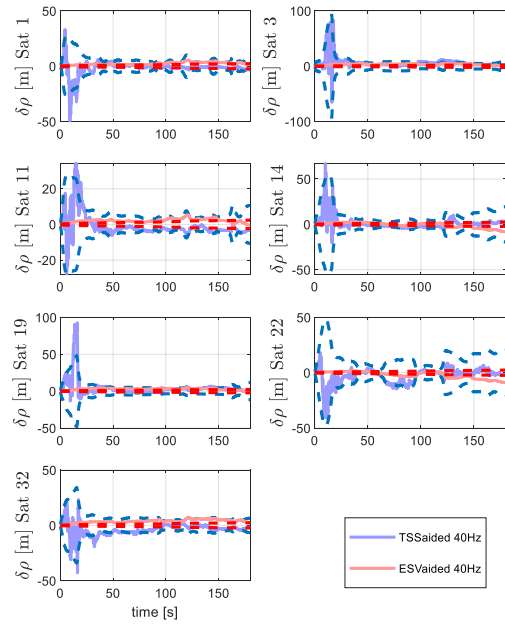
JAMMED



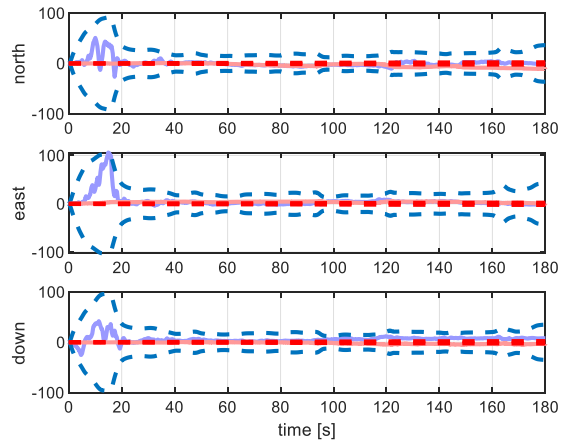
III-258 pseudorange error - DA42 - jammed



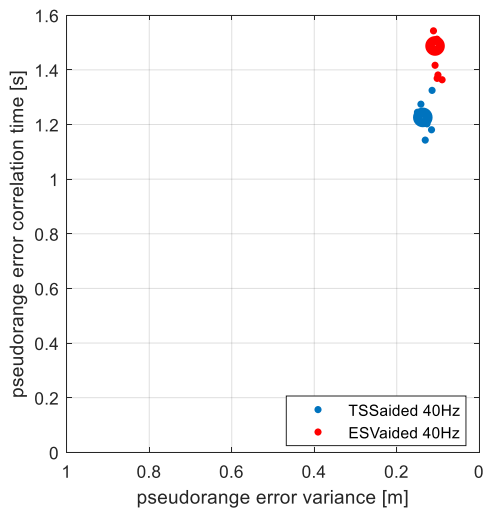
III-260 positioning error - DA42 - jammed



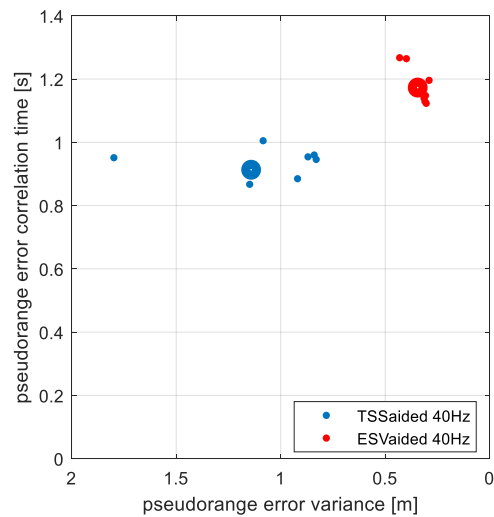
III-259 pseudorange error - high dynamic - jammed



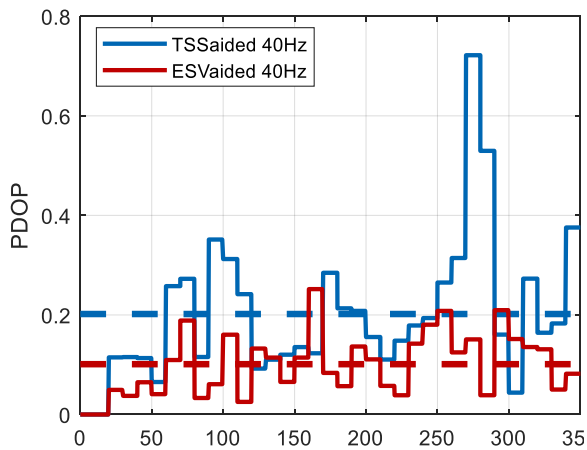
III-261 positioning error - high dynamic - jammed



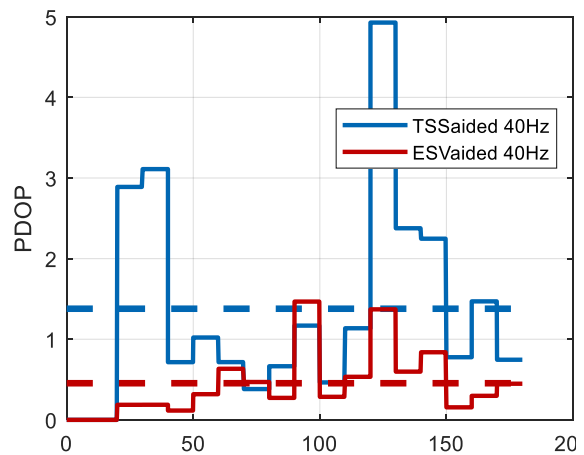
III-262 Comparing raw date - DA42 - jammed



III-263 Comparing raw date - high dynamic - jammed



III-264 PDOP - DA42 - unjammed



III-265 PDOP - high dynamic - unjammed

CONCLUSION: Aided scalar tracking vs. aided error state vector tracking

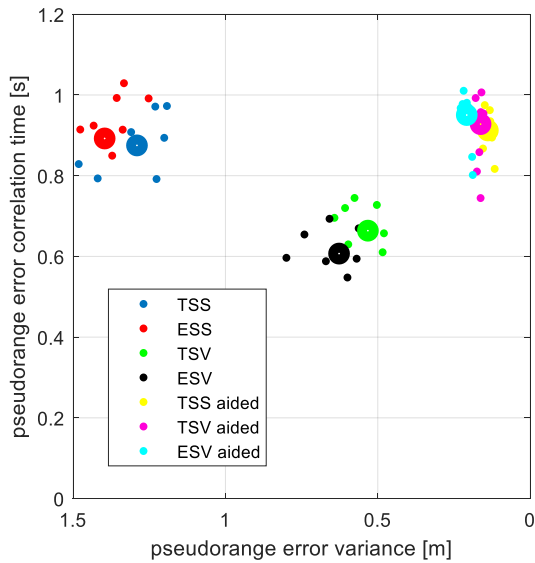
Both tracking architectures give good tracking results, as well in the unjammed as in the jammed scenario. But simulations show that the aided error state vector tracking approach gives definitely better tracking results for the positioning solution and in jammed scenarios.

Moreover, the aided scalar approach is not able to estimate and compensate tilt aiding errors. Therefore, if cannot be excluded that the aiding source has no tilt errors, the aided error state vector tracking approach should be chosen.

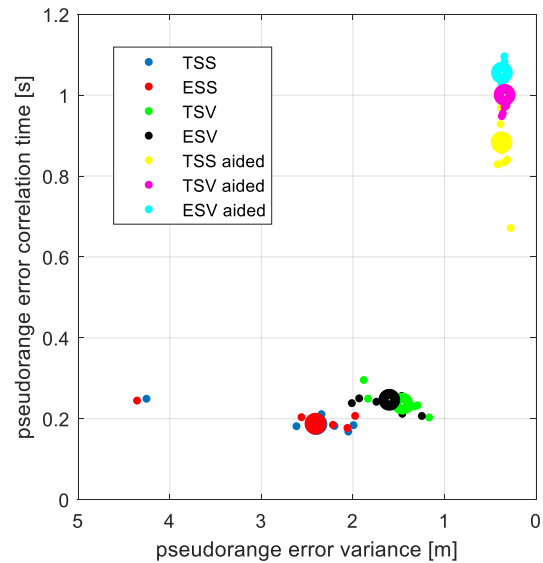
III - 13 Summary of tracking architectures

Many different tracking architectures were derived in this chapter. The tracking architectures are compared to each other step wise, starting from unaided scalar tracking, unaided vector tracking to aided scalar and aided vector tracking.

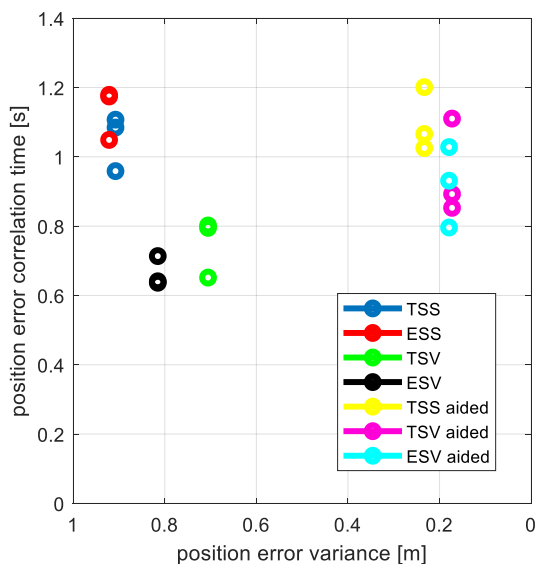
In order to get a comprehensive and summary overview, this section compares all tracking architectures by using the raw data error variance and error correlation times for pseudoranges and the error variances and error correlation times of the ECEF frame positioning solution.



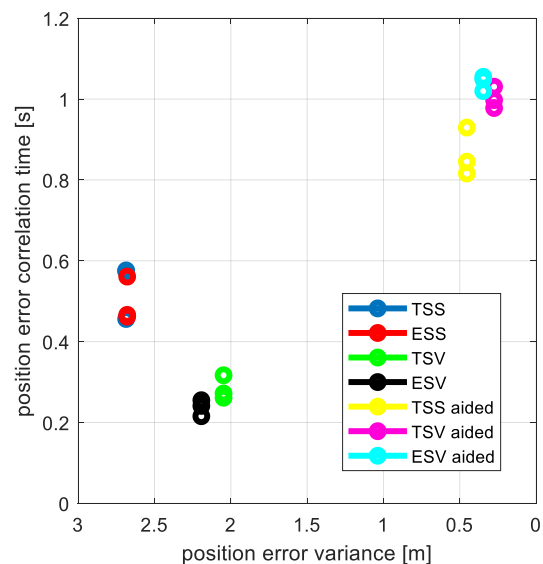
III-266 Comparison RAW Data - DA42 - unjammed



III-267 Comparison RAW Data - high dynamic - unjammed



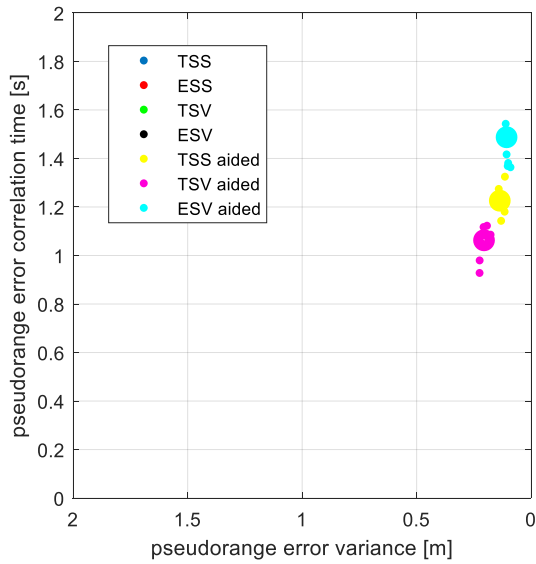
III-268 Comparison POS - DA42 - unjammed



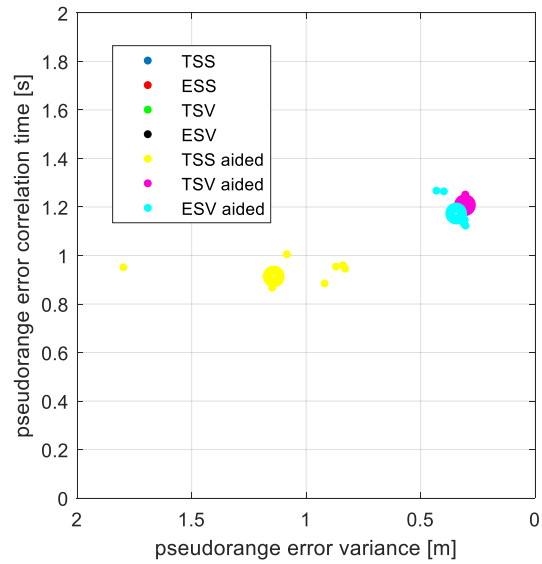
III-269 Comparison POS - high dynamic - unjammed

In the unjammed case, all architectures provide appropriate tracking solution. The unaided scalar total state and unaided scalar error state give the worst solutions. An improved tracking solution is provided by the unaided vector tracking architectures.

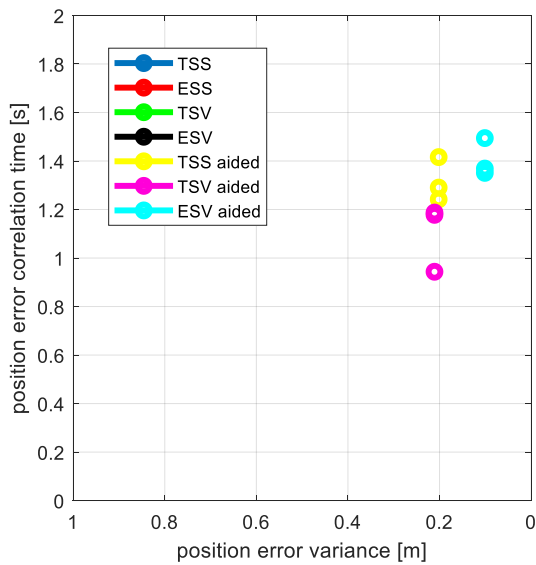
A significant improvement is possible by using aided tracking architectures. All three analyzed aided variants show a very small pseudorange and position error variance, together with long error correlation times.



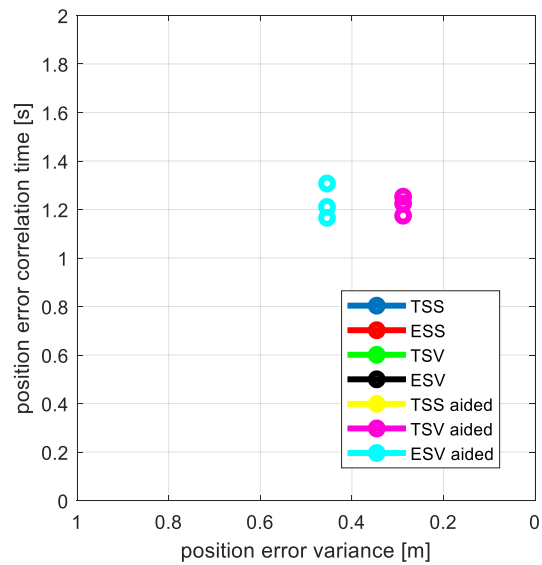
III-270 Comparison RAW Data - DA42 - jammed



III-271 Comparison RAW Data - high dynamic - jammed



III-272 Comparison POS - DA42 - jammed



III-273 Comparison POS - high dynamic - unjammed

In case of jamming, all unaided architectures lose the lock. The unaided vector tracking architectures can withstand longer, respectively are able to keep lock at higher jamming to signal ratios as the unaided scalar tracking approaches.

But as for simulations, the jammer is modeled at the destination and the jamming power becomes higher the smaller the distance, only the aided architectures keep lock.

In jammed scenarios, especially in close distance to the jammer, the observed raw data error characteristics and position error characteristics are mainly caused by the aiding errors. The

discriminator errors only have a negligible influence due to the very low closed loop tracking bandwidth.

IV GPS Modeling and Simulation

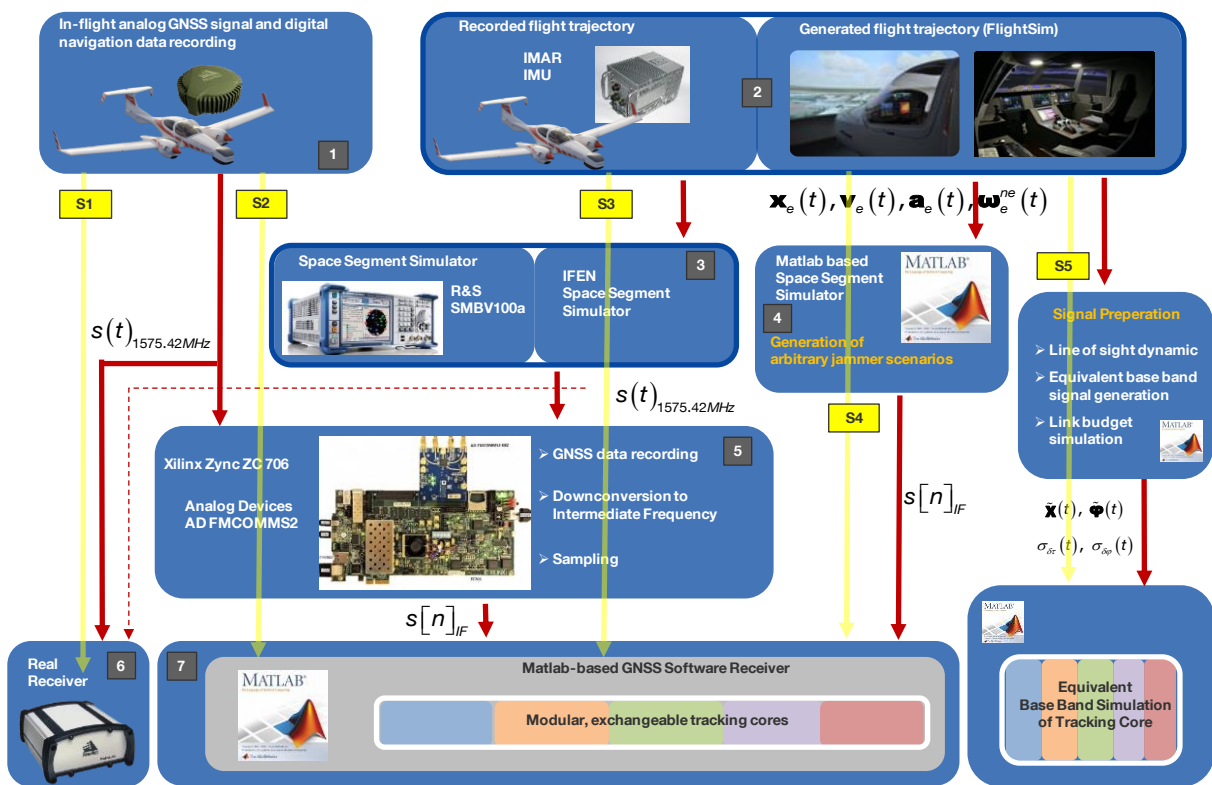
IV - 1 Motivation

GPS navigation plays an always increasing role in many applications. Starting from civil applications like car navigation to military applications like navigation of missiles. Especially upcoming applications, like autonomous driving or autonomous air taxis, rely partly on GPS navigation. For all these applications, it is important to know pre mission, the navigation performance, or to know how robust the navigation is, in case of potential threats, like jamming. For developers of GPS tracking kernels, it is important to know pre mission, the performance of the selected tracking architecture itself, or how good is the selected tuning of the tracking loop.

All this can be evaluated pre mission, by using appropriate simulation concepts, which will be developed and introduced in this chapter.

The major contribution of this chapter are different holistic tracking simulation concepts for different analysis purposes.

In the following, different approaches are described, with each approach having its own application focus.



IV-1GPS simulation approaches

Simulation concept S1 | Concept 1 is the realistic GPS positioning, using real GPS signals and a real GPS receiver. The positioning is done in real time.

Simulation concept S2 | The simulation concept is already a mixture between real hardware and software receiver components. The real GPS signal is recorded using a mounted antenna on the vehicle respectively aircraft. The received GPS signal is down converted to an intermediate

frequency (IF), sampled and stored. Afterwards, the stored and sampled GPS signal at IF frequency is forwarded to a software-based GPS receiver in Matlab, which could be run in post processing on a standard computer or in real time, if the Matlab based software receiver is realized on a FPGA board or digital signal processor. The Matlab based software receiver, developed in the context of this work, allows the exchange of tracking modules. This means, different tracking cores, from classical scalar fixed gain tracking, scalar optimal filter-based tracking over total state unaided vector tracking up to aided vector error state tracking, can be used within the software receiver. By using concept S2|, different tracking architectures can be compared to state-of-the-art receivers.

Simulation concept S3| This concept does not record real GPS signals. The GPS signals are generated using a space segment simulator. The flight trajectories can be recorded, using a high-performance navigation system or can be generated by using flight simulators. Using flight simulators offers the advantage of generating different trajectories, even with high dynamic ranges, where real recording would be very expensive and elaborate. The space segment simulator generates the real GPS signals, which can afterwards be forwarded to the FPGA for down conversion and sampling.

Simulation concept S4| IF band space segment and GPS software receiver S4 is purely software based. The flight trajectories are generated using flight simulators, which provide the possibility to generate very challenging trajectories for testing the limits of the tracking architecture. The flight trajectory is forwarded to a Matlab based space segment simulator. There, real GPS signals are generated, but already at an intermediate frequency level and already sampled. This is a huge advantage compared to S3|, because no high frequency analog electronic is needed. The sampled GPS signals at intermediate frequency can directly be used for tracking in the Matlab based software receiver. A further advantage of this concept is the arbitrary generation of different jamming scenarios. This concept is explained in detail in this chapter.

Simulation concept S5| Equivalent base band simulation. This concept is also purely software based. The simulation is realized in equivalent base band, where only the code phase and carrier phase dynamic matters. This approach enables fast tracking simulation of whole missions. This concept is also optimal for pre mission tuning and tracking stability verification. This concept is also explained in detail in this chapter.

IV - 2 Equivalent base band dynamic state space simulation

IV - 2.1 Motivation

Standalone GPS navigation or integrated GPS navigation systems are currently used at an increasingly number in safety critical applications or at least applications, with safety related requirements to GPS navigation. These include for example auto landing of civil aircrafts, UAV missions in inhabited areas or even autonomous driving cars or positioning of high speed trains, used within the train guard system. Even in military applications, GPS navigation plays an important role, whereby the requirements in this context are mainly concerned about the robustness in case of high noise environments or even GPS performance and robustness in case of jamming and spoofing.

In all these applications, there are minimum requirements on the capability to follow a certain trajectory dynamic and also there are requirements on the maximum allowed dynamic stress and error variance of raw data and positioning solution. Moreover, especially for safety of life applications, the performance of tracking in jammed environments is of great interest and to which jamming power, the tracking can withstand.

All this information must be known before the real application of the receiver or before any real mission. Therefore, in this section an equivalent base band simulation concept is developed, providing the possibility to do a tracking simulation much faster than real time, before mission. Moreover, a simulation concept is developed, providing deep insight into the tracking behavior, like closed loop tracking bandwidth calculation along a selected mission and also analytical stability verification using eigenvalue calculations.

The whole GPS tracking and positioning, respectively the whole chain, starting from the satellite signals to GPS position and velocity solution, can be simulated and evaluated pre mission in a very short time. In figure IV-2 this simulation architecture is shown. This pre mission simulation is equivalent to real GPS receiver tracking. It considers real mission scenarios. This includes real satellite positions, the planned flight trajectories, the received signal power at the antenna interface depending on elevation and the real antenna gain pattern. Also, real noise environments are considered and even jammers can be simulated, while exact jammer signal characteristics and the real or assumed jammer positions are considered.

Especially for performance and robustness under jamming, this simulation approach provides the huge advantage of short simulation times. Given some jamming signals, it is hardly possible to calculate analytically the maximum possible jamming power before loss of lock occurs. For that, a stochastic differential equation would have to be solved, which is done for few scenarios in [78] by using the Langevin differential equation. Due to short simulation times of the equivalent base band approach, many simulation runs are possible in a short time and therefore providing the possibility, determining the maximum jamming power causing loss of lock, by simulation.

One major intention of this work is the comparison of different tracking architectures, starting from classical scalar fixed tracking loops, scalar optimal gain tracking loops, to the extent of different forms of aided and non-aided vector tracking loops. For each tracking architecture, a corresponding equivalent base band simulation model is developed and used for deep analysis of tracking behavior, performance and robustness.

Another question is the correct tuning of all kinds of tracking loops. The introduced simulation concept allows the validation of the tuning, pre mission. Tuning of a GPS tracking loop is always a balance between optimal trajectory following capability and minimum tracking error variance and thus robustness. Here also the introduced simulation concept allows the evaluation of the chosen tuning settings. At each point on the trajectory, the trajectory following can be checked and also the tracking error variance. Due to very short simulation times provided by this equivalent base band approach, a lot of different tuning settings and parameter variations can be simulated pre mission in short time.

The complete GPS simulation gives also the possibility to assess the impact of a suddenly appearing jammer or unforeseen high dynamic maneuvers.

Also, if the GPS navigation is used in auto landing, where a minimum performance and accuracy of the positioning solution has to be guaranteed under all possible circumstances, the derived simulation concept can be used in the clearing process. The state space modeling provides the possibility of analytical, eigenvalue based, tracking stability verifications.

Moreover, it can be used in a Monte Carlo approach, where many different worst-case parameter variations and signal to noise or jamming to signal conditions are evaluated. Using this approach, worst case performance and robustness boundaries of the tracking can be identified.

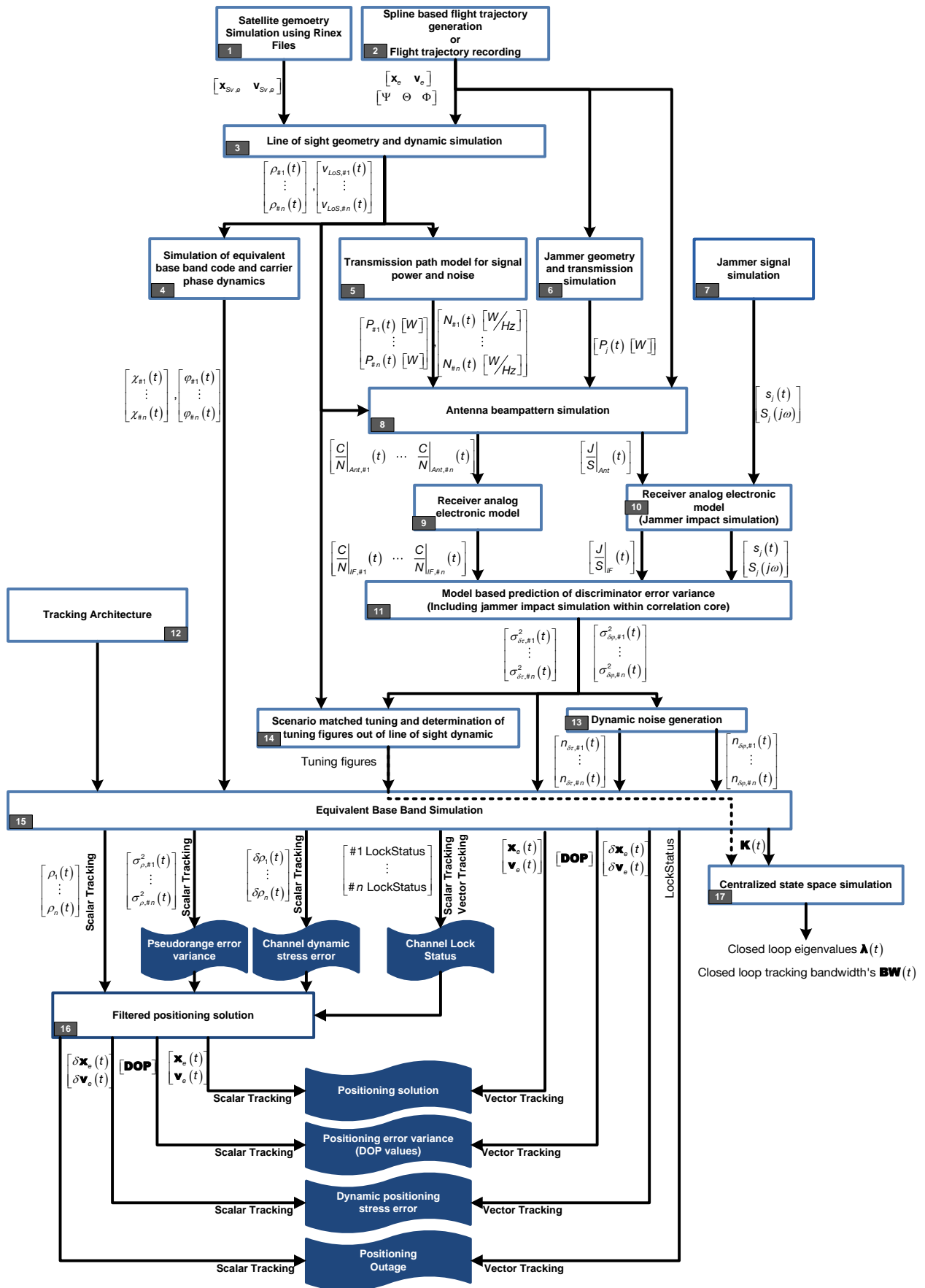
IV - 2.2 Simulation architecture and concept

The characteristic feature of equivalent base band simulation is the pure usage of code phase and carrier phase dynamics. No carrier signal or code signal is used. In a first stage, used for tracking simulation and mission planning, the correlation core for each satellite is represented as a first order differential equation system, using the satellite code phase dynamic as one excitation and the replicated respectively estimated code phase dynamic as second excitation. The output of this first differential equation system is the misalignment between replicated and received code phase. Upon this misalignment, real discriminator noise is superimposed, based on signal to noise models between satellite and receiver and also based on jamming to signal models between defined jammers and the receiver. The resulting real discriminator output is used by the tracking filter as measurement. The tracking filters replique control – being the estimated code frequency or code phase – is fed back and used as a time variable excitement in the correlation core model.

In a second stage, the differential equations of the correlation cores and the differential equations of the tracking filter and the respective coupling between all equations are combined in one linearized first order differential equation system. This combined or centralized differential equation system describes the whole satellite signal tracking and can be used for closed loop tracking bandwidth calculation of each satellite channel – even in case of vector tracking – and can also be used for analytical stability evaluations by using eigen value decompositions of this differential equation system.

For generating the real line of sight code and carrier phase dynamic, the whole signal transmission chain between satellite and receiver must be modeled and considered. Moreover, for calculating and simulating a realistic discriminator error variance, also a detailed signal to noise ratio model of the whole signal transmission chain must be developed. In case of jamming, also a jamming to signal model for the whole transmission chain must be developed.

This section develops the corresponding phase dynamic and signal to noise ratio models, as well the differential equation system for a selected tracking architecture. Because for each tracking architecture, a different differential equation system is necessary, the individual equivalent base band equations are developed alongside of each introduced tracking architecture in chapter III.



IV-2 Dynamic pre mission GPS simulation architecture

1| For line of sight geometry, respectively line of sight dynamic calculation, the actual satellite position and satellite velocity is necessary. The GPS satellites are located at defined orbits around the earth. On each orbit, 5 satellites are distributed. The actual position on the corresponding orbit and the orbit location itself are specified by Keplerian orbit elements. These Keplerian elements are transmitted via navigation message to the receiver. Additionally, they can be taken out of so called “Receiver independent exchange format files” (Rinex). On [“ftp://cddis.nasa.gov/gnss/data/”](ftp://cddis.nasa.gov/gnss/data/) Rinex files of past weeks with different accuracy can be downloaded. A detailed explanation of the Rinex file content is given in [79]

Calculation of the corresponding satellite position, as it is used in this thesis, is already explained in detail in [5, p. 114].

2| This component generates the flight trajectory and platform dynamic. There are different possibilities for trajectory generation. The trajectory can be recorded in flight, by using high performance navigation systems or can also be generated by a flight simulator. In all cases, the trajectory is sampled with a defined sampling rate, which usually is different than the sampling rate used for simulation. In order to match the sampling rates, the recorded or simulated trajectory is approximated by splines, which allow resampling at any sampling rates needed. (IV - 2.3

3| Using the satellite position and velocity, together with the platform position and velocity, the line of sight range and line of sight dynamic is calculated.

4| The line of sight dynamic respectively velocity causes a Doppler shift of carrier and C/A-Code frequency. Knowing the line of sight dynamic, the equivalent base band code- and carrier phase progress can be calculated, depending on time and line of sight velocity.

5| For simulation, besides the code- and carrier phase progress, the signal to noise level C/N at the antenna interface is an important figure. For that, a signal propagation model from satellite antenna to receiver antenna is used. The major part of the noise at antenna interface is thermal noise, having its origin in the antenna itself, but also additional noise sources contribute to the noise at the antenna interface.

6| In order to consider also jammers within the holistic simulation approach, besides the line of sight geometry between aircraft and satellites, also the line of sight geometry between the aircraft and the jammer must be derived.

7| Because there are many different possible jamming signals, a separate module for simulating the wanted jammer type with the corresponding signal characteristics is used. This is important for modeling the jammer impact within the signal processing steps of the GPS receiver. Different jamming signals give different impacts.

8| A huge influence on the received signal to noise level poses the gain pattern of the GPS reception antenna. Classical antennas have a fixed gain pattern, whereby the reception gain is a function of the satellite elevation in the antenna coordinate system. In order to increase the signal to noise level or to decrease in case of jamming, the jamming to signal level, controlled radiation patterns are used. There are different methods that can be applied. Methods like nulling don't need external information regarding elevation of the concerning signals, whereby beam steering or beamforming require this information. The holistic simulation concept in this thesis simulates also the corresponding CRPA algorithms. Therefore, the simulation module needs information about

line of sight geometry to the satellites and also to jammers. By considering these CRPA algorithms in the simulation, it is possible to evaluate pre mission, the improvement of different beam steering, nulling or beam forming algorithms.

9| After having calculated the signal to noise level at the antenna interface, also the effect of the receiver analog electronic must be considered, especially its influence on signal to noise level.

10| The effect of the receiver analog electronic on jammers need a separate analysis. Different jammer signals with different signal characteristics and different frequency characteristics are influenced in different ways by the receiver analog electronic and IF bandpass filter, which is located at the end of the analog signal processing unit.

11| Knowing the signal to noise level, respectively the jamming to signal level at the GPS receiver internal interface to the tracking core, the noise like errors at the output of the discriminators (the discriminator error variance) can be predicted, based on correlation core models.

12| The tracking kernel itself, consisting out of a correlation core and the tracking filters, are a major part of the GPS receiver and need to be simulated within the holistic equivalent base band simulation concept. For simulation, for each tracking kernel derived in chapter III,a distributed state space representation is developed, enabling tracking simulation in equivalent base band.

13| The simulation provides tracking results which are equivalent to real software receiver tracking. In order to reach this goal, also realistic noise is stimulated. For that, based on the predicted measurement error variances, at the output of the discriminators, noise is applied. (IV - 2.7)

14| A feature of this dissertation is the usage of scenario matched tuning of the tracking loop, based on pre mission available dynamic information and signal to noise prediction - also based on corresponding signal propagation models. Even if pre mission tactical reconnaissance about jammer is available, the tuning can be adapted to it. Besides pre mission tuning, also online adaptive scenario matched tuning is realized within this simulation sub module, by considering online variations of platform dynamic patterns and changing signal to noise environments. (III - 3.4 0)

15| This simulation module is the core module. The correlation core module together with the tracking filter are stimulated by the derived code phase and carrier phase dynamic and also the corresponding noise.

16| Given a vector tracking architecture, simulation module 15| already provides a positioning solution together with error information and tracking status information. In case of scalar architectures, an additional positioning filter is necessary to get a positioning solution. This is realized within this simulation sub module.

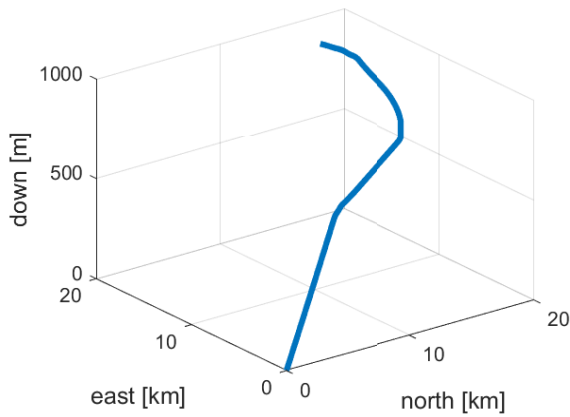
17| Besides the tracking simulation, realized by simulation sub module 16|, a centralized state space approach is developed. For that the whole tracking loop, including correlation core and tracking filter are described as a differential equation system 1st order. Nonlinear equations are linearized about the actual states for each point on trajectory. The resulting linear matrix differential equation system is used to calculate closed loop tracking bandwidth for every satellite channel. Also, the stability can be analyzed by eigenvalue prediction. A detailed derivation is given in III - 3.5.3.

IV - 2.3 Flight trajectory generation

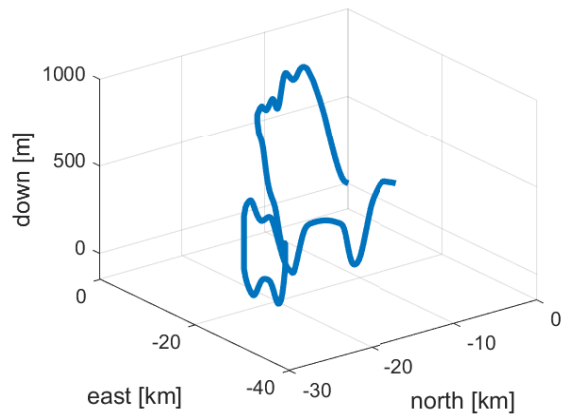
There are basically two different approaches to get flight trajectories. The first approach is flight trajectory recording. For that, typically inertial measurement units of high accuracy are used to record position, velocity, acceleration together with attitude, skew rates and further derivatives. The data are recorded at sampling rates, defined by the used inertial measurement unit. Becoming independent of these rates is possible by using splines. The sampled data are approximated by splines of high order. Afterwards, these splines can be sampled by using individual sampling rates, corresponding to the simulation rate.

The second possibility is, generating the flight trajectory by simulation. In [80], a high accurate approach is described, for getting real synthetic flight trajectories.

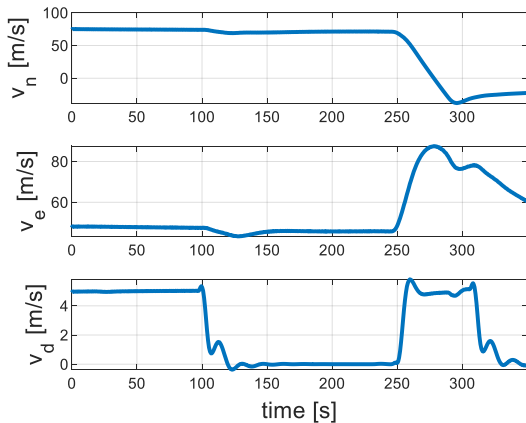
In the following, the flight trajectory of a DA42 approach and a high dynamic flight trajectory are described, used in this thesis for tracking architecture evaluation.



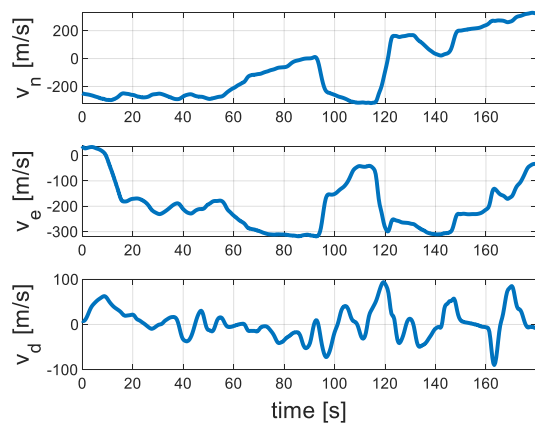
IV-3 DA42 approach - trajectory



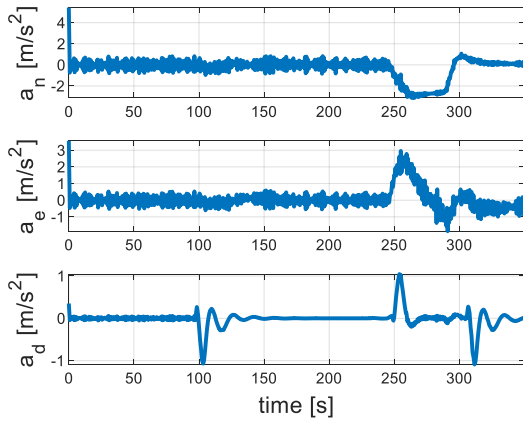
IV-4 High dynamic - trajectory



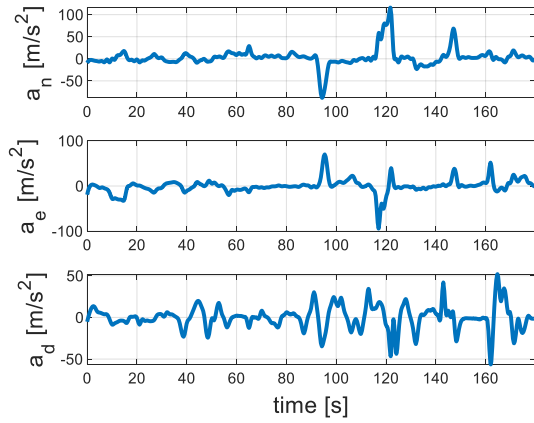
IV-5 DA42 approach - velocity



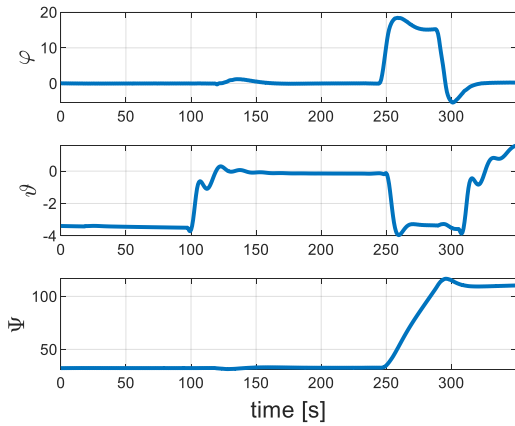
IV-6 High dynamic - velocity



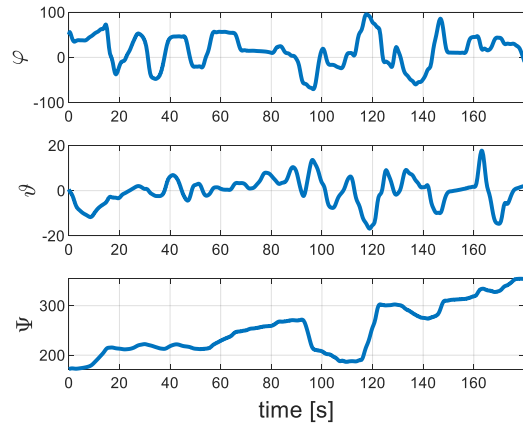
IV-7 DA42 approach - acceleration



IV-8 High dynamic - acceleration



IV-9 DA42 approach - attitude



IV-10 High dynamic - attitude

IV - 2.4 Line of sight geometry

Essential for equivalent base band code phase and carrier phase dynamic calculation is the line of sight dynamic. The carrier and code phase dynamic are the integrated code and carrier Doppler frequency, which in turn are a result of the line of sight dynamic.

At first the line of sight range $\|\mathbf{r}_{LoS,\#Sv}\|$ and the line of sight directional vector $\mathbf{e}_{LoS} = \frac{\mathbf{r}_{LoS,\#Sv}}{\|\mathbf{r}_{LoS,\#Sv}\|}$ are calculated.

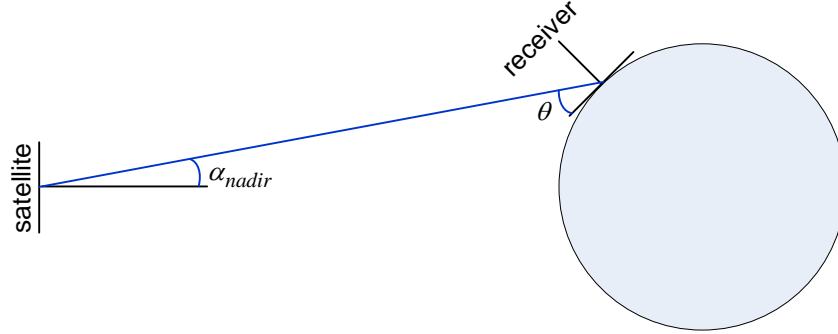
The satellite position and receiver position are given as

$$\text{satellite: } \mathbf{x}_e^{\#Sv} = \begin{bmatrix} x_e^{\#Sv} \\ y_e^{\#Sv} \\ z_e^{\#Sv} \end{bmatrix} \quad \text{receiver: } \mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad (IV-1)$$

The line of sight range gets

$$\mathbf{r}_{e,LoS,\#Sv} = \mathbf{x}_e^{\#Sv} - \mathbf{x}_e \quad (IV-2)$$

The following figures shows the basic relations for elevation and nadir angle.



IV-11 nadir and elevation definition

For IF link budget calculation, the elevation of the satellite in the receiver antenna coordinate system is necessary. For that, the line of sight range vector is written in b-frame coordinates of the antenna coordinate system.

$$\mathbf{r}_{b,LoS,\#Sv} = \mathbf{R}_{bn}(\phi_b, \theta_b, \Psi_b) \cdot \mathbf{R}_{ne}(\lambda, \phi) \cdot \mathbf{r}_{e,LoS,\#Sv} \quad (IV-3)$$

Using equation (IV-3), the elevation θ_b can be calculated as

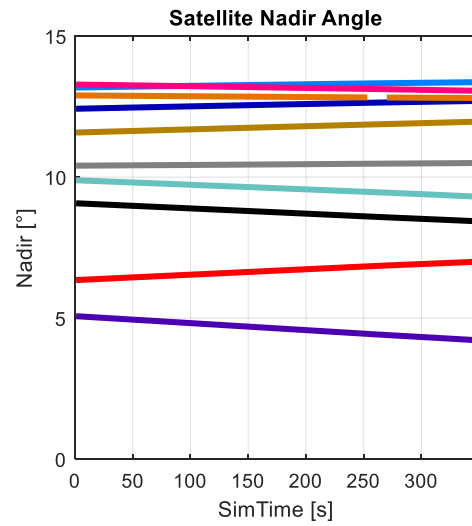
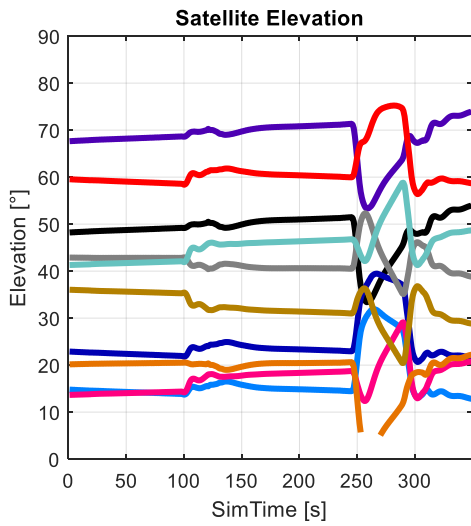
$$\theta_b = \sin^{-1} \left(\frac{-r_{bx,LoS,\#Sv}}{\|\mathbf{r}_{b,LoS,\#Sv}\|} \right) \quad (IV-4)$$

For calculating the nadir angle, the line of sight vector is written in n-frame coordinates of the satellite position, assuming that the antenna main direction is directed towards the d-axis of the n-frame.

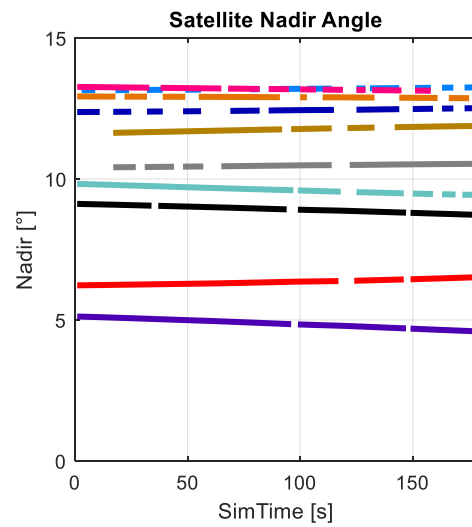
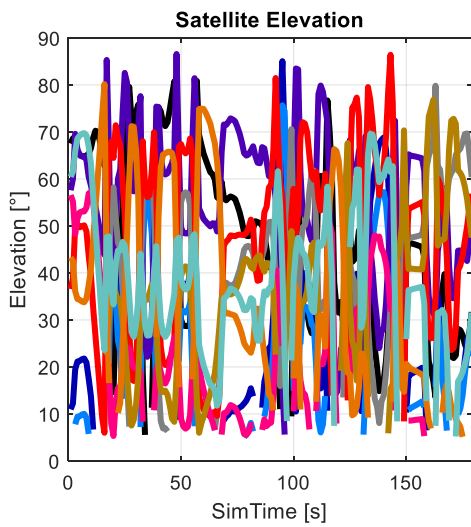
$$\mathbf{r}_{n,LoS,\#Sv} = \mathbf{R}_{ne}(\lambda^{\#Sv}, \phi^{\#Sv}) \cdot \mathbf{r}_{e,LoS,\#Sv} \quad (IV-5)$$

$$\alpha_{nadir} = \left(\frac{\pi}{2} \right) - \sin^{-1} \left(\frac{-r_{nx,LoS,\#Sv}}{\|\mathbf{r}_{n,LoS,\#Sv}\|} \right) \quad (IV-6)$$

The following figures show the elevations and nadir angles for all satellites, having an elevation of at least 5 degrees in the local antenna coordinate system and at the same time in the n-frame.

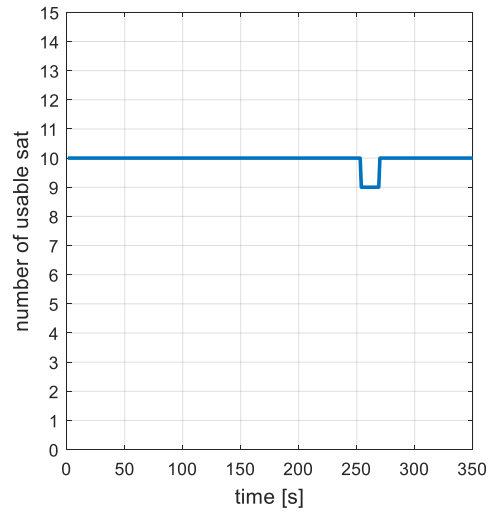
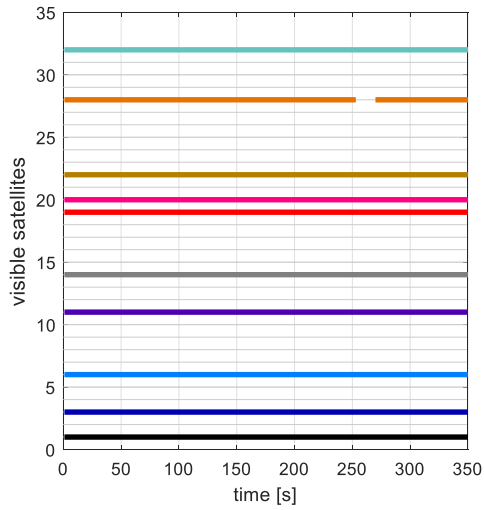


IV-12 elevation & nadir angle satellite (DA42 approach)

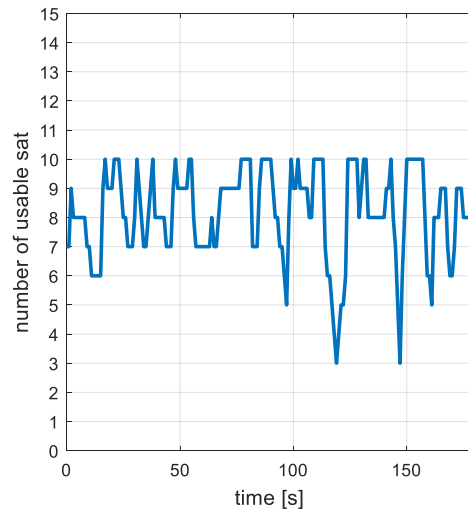
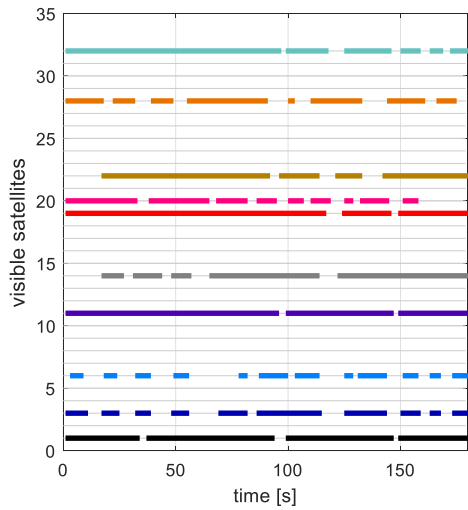


IV-13 elevation & nadir angle satellite (high dynamic)

A summary of the usable satellites in both scenarios is given by the following figures.



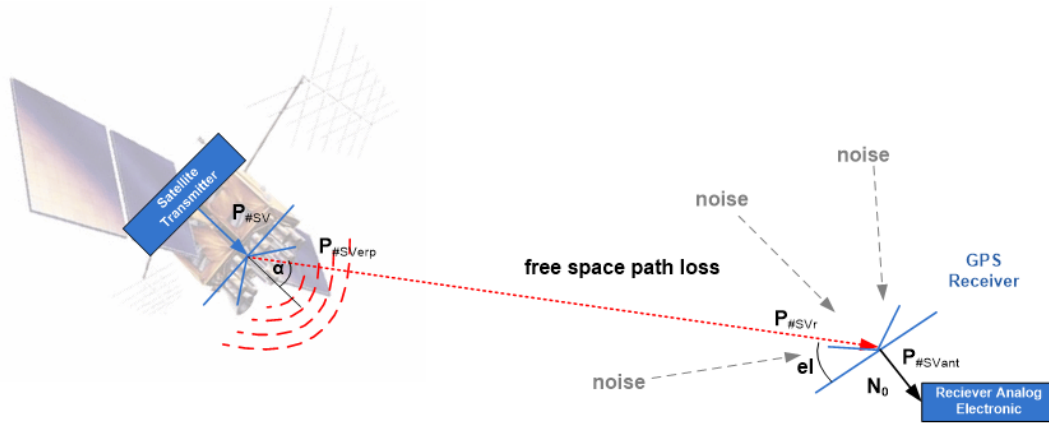
IV-14 usable satellites unjammed (DA42 approach)



IV-15 usable satellite unjammed (high dynamic)

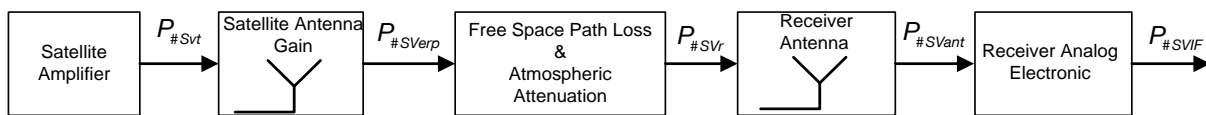
IV - 2.5 Satellite link budget model

In this section, an IF link budget model for a defined satellite will be introduced. The link budget model describes the power relations between the satellite transmitter and the IF interface within a GPS receiver.



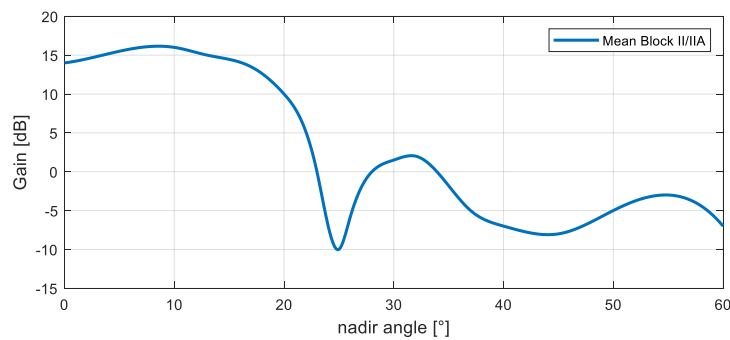
IV-16 satellite receiver scenery

The transmission antenna feeding power within the GPS receiver depends on the GPS satellite generation and on the satellite age. The feeding power is denoted as $P_{\#Svt}$.



IV-17 generic satellite link budget model

The effective radiated power $P_{\#SVerp}$ depends on the directivity of the satellite antenna and the antenna gain. The transmitted power is focused on a defined nadir angle range, as shown in the following example satellite antenna gain pattern.



IV-18 satellite antenna gain distribution

The effective radiated power $P_{\#SVerp}$ gets

$$P_{\#SVerp} = G_{\#Sv}(\alpha_{nadir}) \cdot P_{\#Svt} \quad (IV-7)$$

The main signal power loss is caused by the free space loss, given below.

$$\Delta P_{\#SVf_{sp}} = \frac{1}{4 \cdot \pi \cdot \rho_{\#Sv}^2} \quad (IV-8)$$

Additional loss is caused by atmospheric attenuation. There is no mathematical expression, just some empirical attenuation data, which can be described by an elevation dependent loss function.

$$G_{atm}(\theta) \quad (IV-9)$$

The atmospheric attenuation depends on the signal travel distance through atmosphere, which in turn can be expressed as a function of the satellite elevation θ within the receiver antenna coordinate system.

The available satellite signal power, received at the receiver antenna gets

$$P_{\#SVr} = P_{\#SVep} \cdot G_{atm}(\theta) \cdot \frac{1}{4 \cdot \pi \cdot \rho_{\#Sv}^2} \quad (IV-10)$$

The available satellite signal power at the antenna interface depends on the receiver antenna gain, which can be a fixed one in case of normal GPS antennas or can be variable, in case of controlled radiation pattern antennas (CRPA's), being introduced in a following chapter.

The antenna gain consists out of the directional gain and the reciprocity respectively effective antenna aperture.

$$P_{\#SVant} = P_{\#SVr} \cdot G_{ant}(\phi, \theta) \cdot \frac{\lambda^2}{4 \cdot \pi} \quad (IV-11)$$

A detailed consideration of the antenna gain $G_{ant}(\phi, \theta)$ in case of different controlled radiation pattern antennas and algorithms, is provided by a following chapter.

The received signal after the antenna is still at L1 frequency. Before tracking, the signal must be down converted to an intermediate frequency f_{IF} and amplified.

This down-conversion stage is called the receiver analog electronic, consisting out of at least three down conversion, filtering and amplification stages. At the end is a band pass, being essential for discriminator error variance.

What matters for tracking performance and robustness, is not the absolute power at IF frequency, but the signal to noise ratio SNR_{IF} .

For calculating the SNR_{IF} , besides the signal power at the output of the antenna interface, also the noise power spectral density at the output of the antenna interface is necessary. The noise at antenna interface is mainly caused by the thermal noise of the antenna itself. Additional sources are radio interference and astronomical background radiation. In case of jamming, the jamming signal provides the main part of the noise.

The thermal noise power spectral density at the antenna interface gets

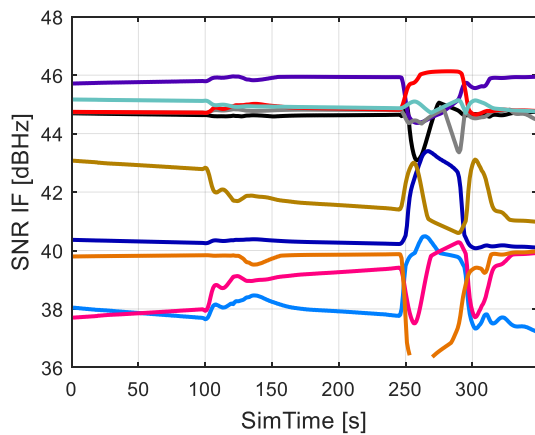
$$N_{ant} = k \cdot T_{ant} \quad (IV-12)$$

with k being the Boltzmann constant and T_{ant} being the physical antenna temperature. The following example shows a typical noise power spectral density

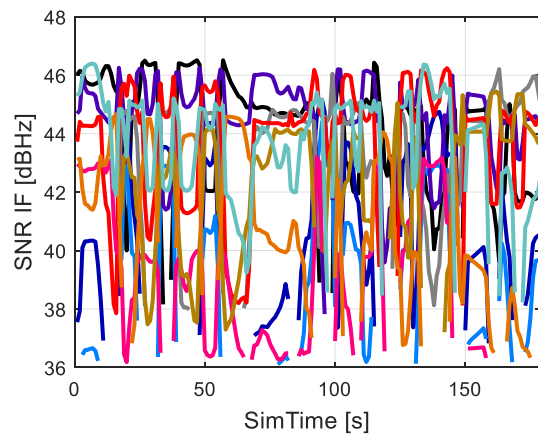
$$N_A = k \cdot T_A = 1.3806503 \cdot 10^{-23} \text{Ws/K} \cdot 290\text{K} = 4 \cdot 10^{-21} \text{Ws} = -204 \text{dBW/Hz} \quad (IV-13)$$

In case of controlled radiation pattern antennas, the IF signal power and therefore the SNR_{IF} changes.

The following figures show the SNR at the IF interface for the DA42 approach and the high dynamic fighter trajectory, using a standard GPS reception antenna gain pattern, given in V - 6.1 and a receiver analog electronic noise figure of 2 dB.



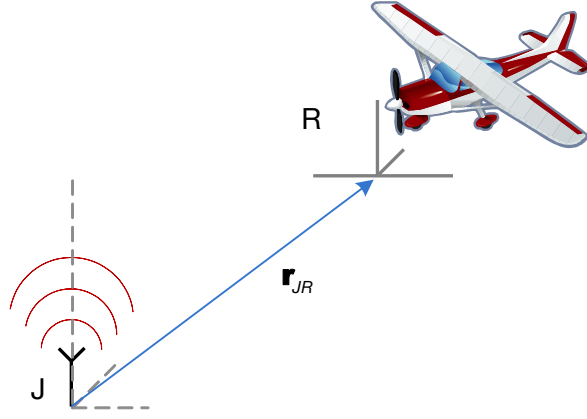
IV-19 SNR IF interface - DA42 - unjammed



IV-20 SNR IF interface - high dynamic - unjammed

IV - 2.6 Jammer link budget model

For the jammer link budget calculation, at first the line of sight geometry between the jammer and the GPS receiver antenna is necessary. The following figure shows the corresponding geometry.



IV-21 Jammer - Receiver geometry

The jammer position and GPS receiver antenna position are defined as

$$\mathbf{x}_{e,J} = \begin{bmatrix} x_{e,J} \\ y_{e,J} \\ z_{e,J} \end{bmatrix} \quad \mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad (IV-14)$$

The line of sight vector is defined as

$$\mathbf{r}_{e,JR} = \mathbf{x}_e - \mathbf{x}_{e,J} \quad (IV-15)$$

Important for the link budget calculation is the line of sight vector between jammer and receiver antenna, in the local receiver antenna coordinate system. This local system is called the antenna body system.

$$\mathbf{r}_{b,JR} = \mathbf{R}_{bn}(\psi_R, \theta_R, \phi_R) \cdot \mathbf{R}_{ne}(\lambda, \mu) \cdot \mathbf{r}_{e,JR} \quad (IV-16)$$

The elevation θ_R , under which the jamming signal is received in the local antenna coordinate system, can be calculated by using the components of $\mathbf{r}_{b,JR}$.

$$\theta_R = \arcsin\left(\frac{r_{bz,JR}}{\|\mathbf{r}_{b,JR}\|}\right) \quad (IV-17)$$

The jammer azimuth angle ψ_R in the local antenna coordinate system is given as

$$\psi_R = \arctan\left(\frac{r_{by,JR}}{r_{bx,JR}}\right) \quad (IV-18)$$

The received jamming power at the receiver antenna can be calculated by using the free space transmission equation as given in [81, p. 398].

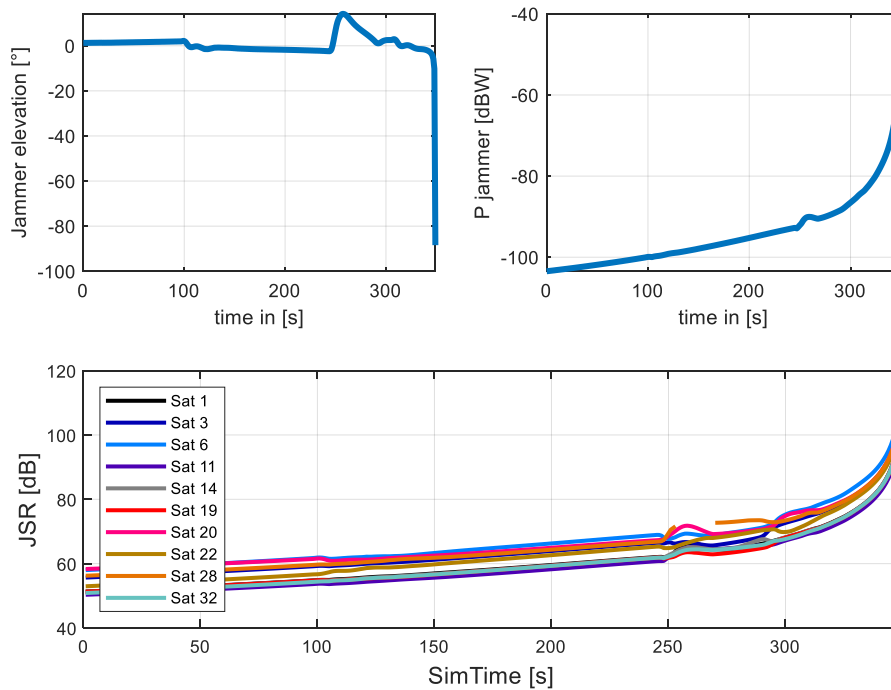
$$P_{J,ant} = P_J \cdot G_J(\psi_J, \theta_J) \cdot \frac{1}{4 \cdot \pi \cdot \|r_{b,JR}\|^2} \cdot G_R(\psi_R, \theta_R) \cdot \frac{\lambda^2}{4 \cdot \pi} \quad (IV-19)$$

$$\begin{aligned} P_J &= \text{initial jammer transmission power} \\ G_J(\psi_J, \theta_J) &= \text{jammer transmission antenna gain} \\ G_R(\psi_R, \theta_R) &= \text{receiver reception antenna gain} \end{aligned} \quad (IV-20)$$

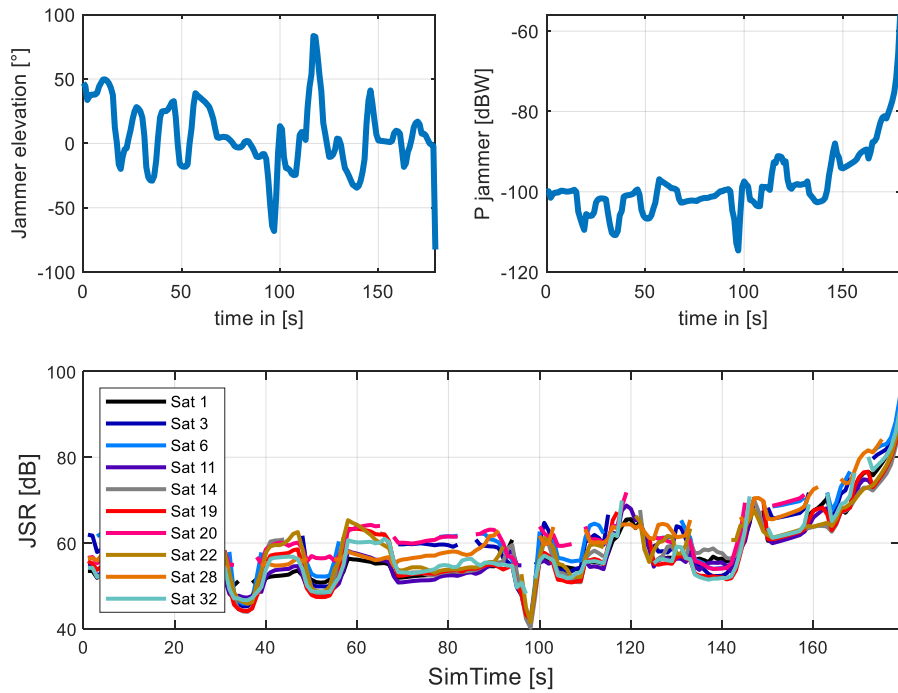
Using the derived satellite signal power from the previous section together with equation (IV-19), the jamming to signal ratio at the antenna interface can be predicted along the trajectory. The following figures show the predicted jamming to signal ratio for the DA42 approach and the high dynamic trajectory.

In this section, as receiving antenna characteristic, a normal GPS antenna according to section V - 6.1 is used. The advantage of this simulation approach is, that also the characteristics of CRPA's can be used. At every point on trajectory the dynamic beam forming or nulling pattern can be calculated and the corresponding gain towards satellite and jammer can be used for signal to noise ratio or jamming to signal ratio calculation.

In the following, the jamming to signal ratio is only shown for the satellites in view, having an appropriate elevation. The given jamming to signal ratio and jamming power is valid for the receiving antenna interface



IV-22 Jamming to signal ratio at antenna interface - DA42



IV-23 Jamming to signal ratio at antenna interface - Fighter

The figures above assume a jammer transmission power of 1 kW. As jammer transmission antenna gain pattern, a spherical gain pattern with 0 dB is assumed.

IV - 2.7 Discriminator error variance prediction

The discriminator output provides the actual code and carrier phase deviation between the replicated code or carrier phase and the received ones. It is used as measurement in the subsequent tracking filter. The discriminator error variance is a vital tuning figure, if optimal filter-based tracking is used. This error variance corresponds directly to the measurement error variance within the matrix **R** of the tracking filter.

Pre mission prediction of the discriminator error variance is therefore essential for pre mission tracking simulation and evaluation.

Within literature [82], the following formulas are given for calculating the discriminator error variance, based on the known signal to noise ratio.

Early-Late code phase discriminator error variance:

$$\sigma_{n,\delta\tau}^2 = \frac{d \cdot T_{CAchip}^2}{4 \cdot T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}} \right)} \quad (IV-21)$$

Early-Late-Power code phase discriminator error variance:

$$\sigma_{n,\delta r}^2 = \frac{T_{CAchip}^2}{4 \cdot T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}}\right)} \cdot \left(1 + \frac{2}{T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}}\right)}\right), \quad d=1 \quad (IV-22)$$

The equations above can be used for pre mission discriminator error variance prediction. The satellite link budget model is used to predict C_{IF} . (Additionally the influence of the analog signal processing unit on C_{IF} and N_{IF} must be considered, which is derived in V - 5 .

A further deeper analysis of discriminator error variances and especially jammer impact on it, is given in section V - 2.5 .

IV - 2.8 Code and carrier phase dynamic

The received signal at antenna interface is down converted to an intermediate frequency IF. The following equation describes the signal for one satellite.

$$s_{IF}(t) = A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \sin(\varphi_{IF}(t)) + J(t) + n(t)$$

A_{IF} : signal amplitude
 D : navigation message
 C : C/A code (IV-23)
 J : jammer
 χ : received code phase of CA-Code
 φ_{IF} : received carrier phase

The code phase dynamic and carrier phase dynamic depend on the line of sight dynamic between the GPS receiver and the corresponding satellite.

$$\chi(t) = \chi(t_0) + f_{C/A} \cdot (t - t_0) + \frac{f_{C/A}}{c} \int_{t_0}^t v_{LoS,s}(t) dt + \Delta\chi_{clk,sv}(t, t_0) + \Delta\chi_{clk,R}(t, t_0) + \Delta\chi_{e,s}(t, t_0) \quad (IV-24)$$

The clock drift within the satellite and the clock drift of the receiver NCO causes a phase shift

$$\Delta\chi_{clk,sv}(t) = f_{C/A} \cdot \delta t_{sv}(t) = f_{C/A} \cdot (t_{sv}(t) - t) \quad (IV-25)$$

$$\Delta\chi_{clk,R}(t) = f_{C/A} \cdot \delta t_R(t) = f_{C/A} \cdot (t_R(t) - t) \quad (IV-26)$$

One can see, that a constant line of sight velocity gives a linear code phase progression. An accelerated line of sight dynamic gives a quadratic code phase progression.

The carrier phase dynamic can be written in a similar way.

$$\varphi(t) = \varphi(t_0) + 2 \cdot \pi \cdot f_{L1} \cdot (t - t_0) + \frac{2 \cdot \pi \cdot f_{L1}}{c} \int_{t_0}^t v_{LoS,s}(t) dt + \Delta\varphi_{clk,sv}(t, t_0) + \Delta\varphi_{clk,R}(t, t_0) + \Delta\varphi_{\epsilon,s}(t, t_0) \quad (IV-27)$$

The clock drift within the satellite and the clock drift of the receiver NCO causes a phase shift

$$\Delta\varphi_{clk,sv}(t) = 2 \cdot \pi \cdot f_{L1} \cdot \delta t_{sv}(t) = 2 \cdot \pi \cdot f_{L1} \cdot (t_{sv}(t) - t) \quad (IV-28)$$

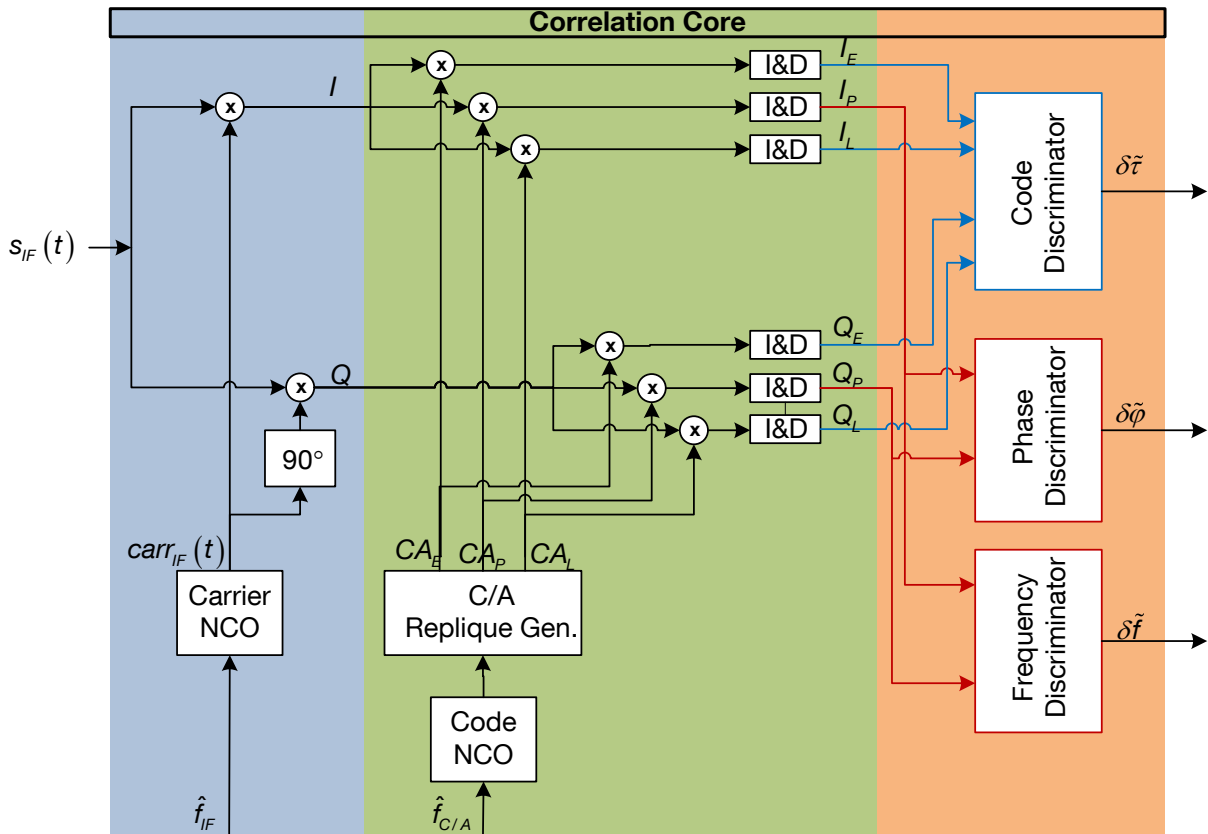
$$\Delta\varphi_{clk,s}(t) = 2 \cdot \pi \cdot f_{L1} \cdot \delta t_R(t) = 2 \cdot \pi \cdot f_{L1} \cdot (t_R(t) - t) \quad (IV-29)$$

For equivalent base band simulation, only the code phase and carrier phase dynamic are necessary, together with the signal amplitude respectively power. No time space signal representation is necessary.

IV - 2.9 State space representation of carrier wipe out, correlation and discrimination

This section derives an equivalent base band representation of the tracking loop. For that, the tracking loop is divided in two parts. The correlation core and the loop filter. For both components, a state space representation is derived.

The following figure shows the correlation core.



IV-24 Correlation core

In the equivalent base band, the time space GPS carrier and code signal are not of interest. Only the phase progress or variation of received C/A code phase and carrier phase are relevant.

This is due to the actual operation of the correlation core and its discriminator. The task of the correlation core is to measure the phase difference between the received C/A code and the replicated one as well as the phase difference between the received carrier and the replicated carrier.

The correlation core consists out of three components which must be considered in the equivalent base band model.

Replique generation

The carrier NCO generates at its output a sine signal, which frequency and phase is controlled to match the received carrier at IF frequency. Also, the code NCO generates at its output the C/A code frequency, which is also controlled to generate a C/A code replique, matching the received C/A code in phase and frequency. The transfer functions of the carrier and code NCO get

$$G_{NCO,L1IF}(s) = \frac{2 \cdot \pi}{s} \quad G_{NCO,C/A}(s) = \frac{1}{s} \quad (IV-30)$$

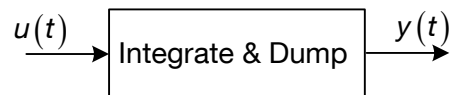
The replicated carrier is used for down conversion of the received signal.

For equivalent base band simulation, the time space representation of the replicated carrier and code signals are not necessary. Only the actual replicated code phase and carrier phase are needed.

Because the discriminator calculates the code phase error and the carrier phase error between the replicated and the received signal, in equivalent base band, this can be realized instantaneously by using the received code and carrier phase together with the replicated code and carrier phase.

Integrate and dump operation

The correlation consists out of the multiplication of the replicated code signal with the received code signal. Afterwards the product is integrated over a defined time - the correlation time. The multiplication is irrelevant in the base band model. Only the integration must be considered due to its influence on dynamic behavior and bandwidth.



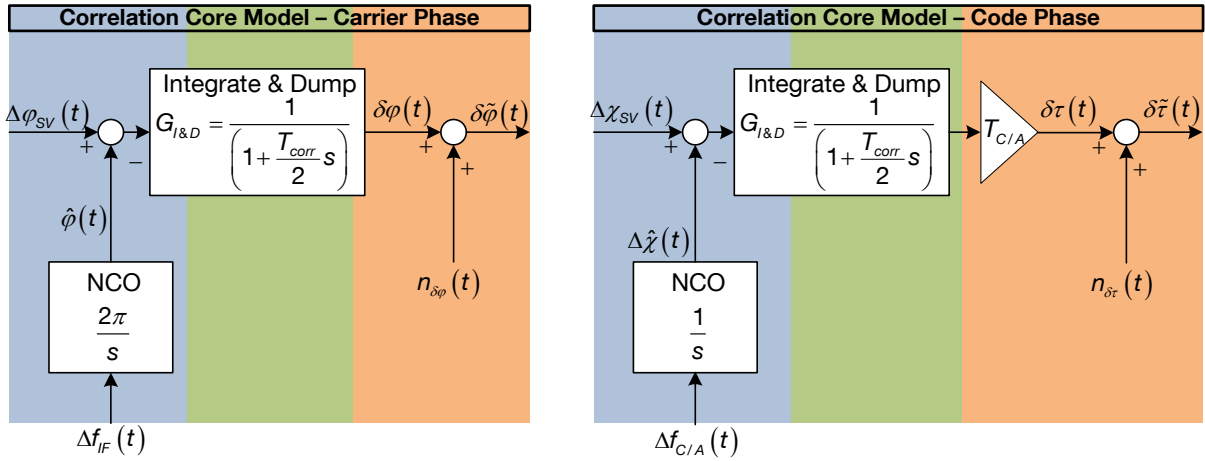
The integration can be expressed as a convolution with a rectangular impulse.

$$y(t) = \frac{1}{T_{corr}} \cdot \int_{<T_{corr}>} u(t) dt = u(t) * \frac{1}{T_{corr}} \cdot \text{rect}\left(\frac{t}{T_{corr}}\right) \quad (IV-31)$$

$$y(t) \circ - \bullet y(s) = u(s) \cdot \frac{1}{T_{corr}} \cdot \frac{1 - e^{-T_{corr} \cdot s}}{s}, \text{ using Pade 1st order: } e^{-T_{corr} \cdot s} \approx \frac{1 - \frac{T_{corr} \cdot s}{2}}{1 + \frac{T_{corr} \cdot s}{2}} \quad (IV-32)$$

$$G_{I\&D}(s) = \frac{1}{1 + \frac{T_{corr} \cdot s}{2}} \quad (IV-33)$$

The following two figures show the equivalent base band approach for the correlation core, including stimulation of the discriminator error.



IV-25 Correlation core in equivalent base band

In the next step, the correlation core will be developed as a state space model. The discriminator output is converted in units of seconds.

The differential equation system describing the code correlation core gets

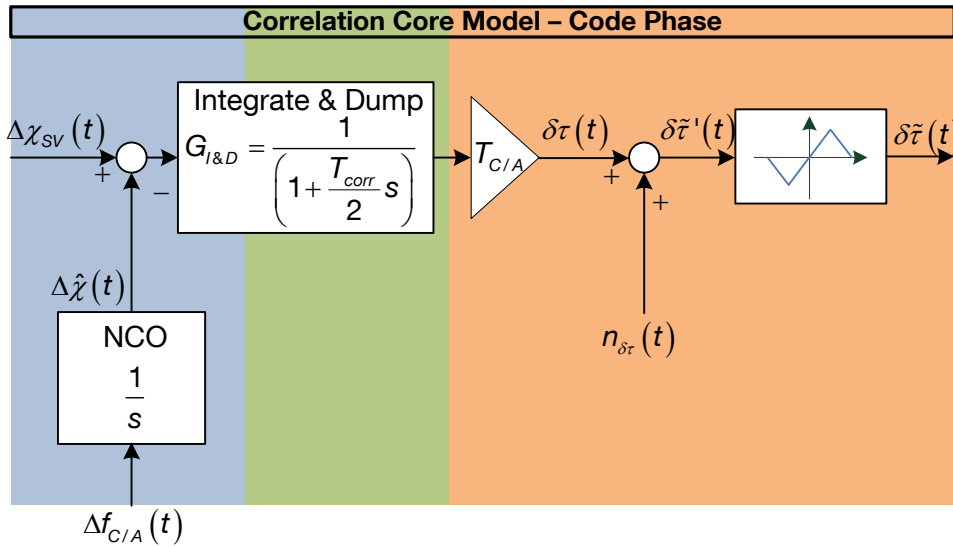
$$\begin{aligned}
 \dot{\mathbf{z}}_{DLLcorr} &= \mathbf{A}_{DLLcorr} \cdot \mathbf{z}_{DLLcorr} + \mathbf{B}_{DLLcorr} \cdot \mathbf{u}_{DLLcorr} \\
 y_{DLLcorr} &= \mathbf{c}_{DLLcorr}^T \cdot \mathbf{z}_{DLLcorr} + w_{DLLcorr} \\
 \begin{bmatrix} \delta\dot{\tau} \\ \Delta\dot{\hat{\chi}} \end{bmatrix} &= \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tau \\ \Delta\hat{\chi} \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta\chi_{SV} \\ \Delta\hat{f}_{C/A} \end{bmatrix} \\
 \delta\tilde{\tau} &= [1 \ 0] \cdot \begin{bmatrix} \delta\tau \\ \Delta\hat{\chi} \end{bmatrix} + n_{\delta\tau}
 \end{aligned} \tag{IV-34}$$

The differential equation system describing the carrier correlation core has a similar structure.

$$\begin{aligned}
 \dot{\mathbf{z}}_{PLLcorr} &= \mathbf{A}_{PLLcorr} \cdot \mathbf{z}_{PLLcorr} + \mathbf{B}_{PLLcorr} \cdot \mathbf{u}_{PLLcorr} \\
 y_{PLLcorr} &= \mathbf{c}_{PLLcorr}^T \cdot \mathbf{z}_{PLLcorr} + w_{PLLcorr} \\
 \begin{bmatrix} \delta\dot{\varphi} \\ \Delta\dot{\hat{\varphi}} \end{bmatrix} &= \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2}{T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\varphi \\ \Delta\hat{\varphi} \end{bmatrix} + \begin{bmatrix} \frac{2}{T_{corr}} & 0 \\ 0 & 2 \cdot \pi \end{bmatrix} \cdot \begin{bmatrix} \Delta\varphi_{SV} \\ \Delta\hat{f}_{IF} \end{bmatrix} \\
 \delta\tilde{\varphi} &= [1 \ 0] \cdot \begin{bmatrix} \delta\varphi \\ \Delta\hat{\varphi} \end{bmatrix} + n_{\delta\varphi}
 \end{aligned} \tag{IV-35}$$

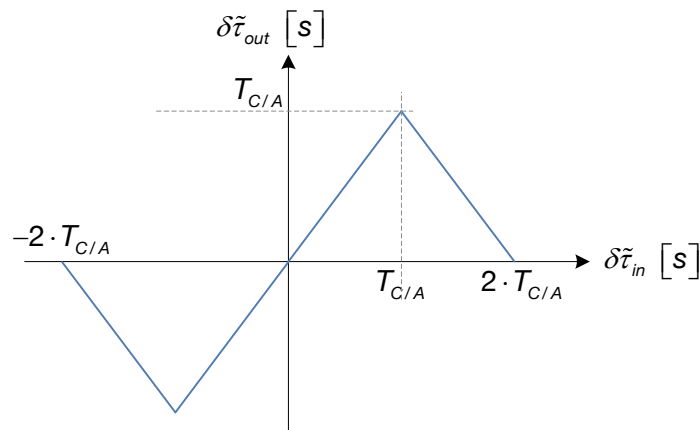
Nonlinear discriminator

In the previous section, the discriminator function was assumed to be linear. This is true, as long as the deviation between the received code phase and replicated code phase is smaller than one code phase chip (and if small discriminator spacing is used, even earlier). But beyond, the discriminator becomes nonlinear. In case of a small early-late spacing, the nonlinear region starts at even smaller phase deviations.



IV-26 Equivalent base band correlation core - nonlinear discriminator

Depending on the discriminator type, the nonlinearity looks different. In the following, an example of a nonlinear "early-late" discriminator is shown.



IV-27 Nonlinear discriminator function

In order to consider this nonlinear discriminator characteristic in the equivalent base band simulation, the output $\delta\tilde{\tau}$ is applied to the nonlinear discriminator function, which is realized by a section wise defined function.

IV - 2.10 Distributed state space representation

For equivalent base band tracking simulation, a state space representation of the whole tracking loop is developed.

The tracking loop can be separated in two components. The correlation core, which was developed in the previous section and the loop filter itself. Additionally, the coupling between the correlation core and the loop filter must be considered within the state space representation.

For each architecture, the specific distributed state space matrices are derived in chapter III.

Therefore, in this section, only the basic principles of the distributed state space realization will be developed, using one example tracking filter.

The state space representation of the correlation core is given by

$$\begin{aligned}\dot{\mathbf{z}}_{corr} &= \mathbf{A}_{corr} \cdot \mathbf{z}_{corr} + \mathbf{B}_{corr} \cdot \mathbf{u}_{corr} \\ \mathbf{y} &= \mathbf{c}_{corr}^T \cdot \mathbf{z}_{corr} + \mathbf{d}_{corr}^T \cdot \mathbf{u}_{corr}\end{aligned}\quad (IV-36)$$

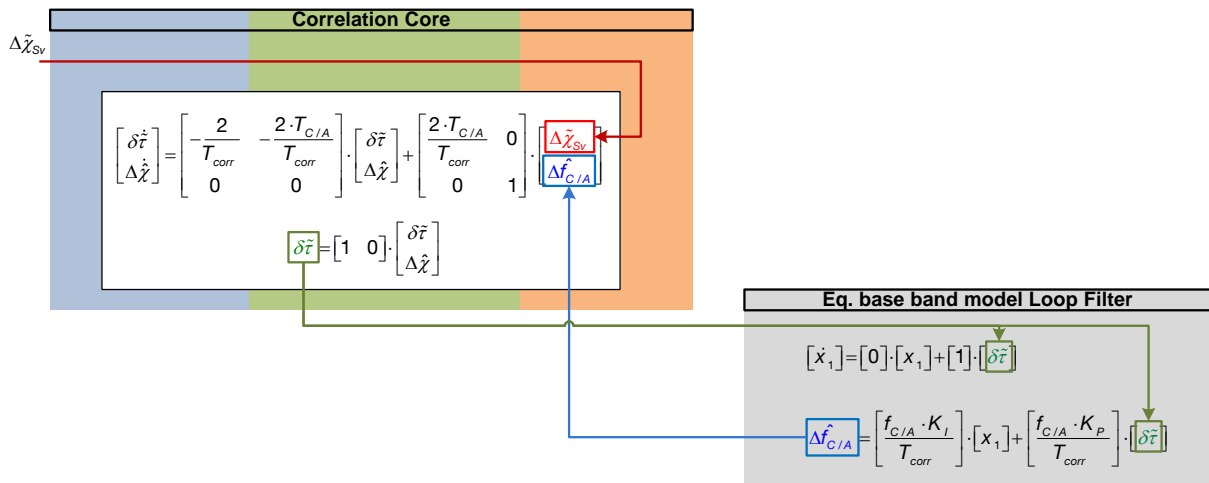
As example, the correlation core of code tracking loop is given below

$$\begin{bmatrix} \dot{\delta\tilde{\tau}} \\ \dot{\Delta\hat{\chi}} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A}}{T_{corr}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi} \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A}}{T_{corr}} & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta\tilde{\chi}_{SV} \\ \Delta\hat{f}_{C/A} \end{bmatrix}\quad (IV-37)$$

As the following architecture in figure IV-28 shows, $\Delta\tilde{\chi}_{SV}$ is the input variable and $\Delta\hat{f}_{C/A}$ is the feedback variable. This feedback variable is also the coupling between the equivalent base band representation of the loop filter and the state space form of the correlation core.

The discriminator measurement, which is used as input to the loop filter, is given in the next equation.

$$\delta\tilde{\tau} = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi} \end{bmatrix}\quad (IV-38)$$



IV-28 Distributed state space representation – scalar fixed gain DLL

The state space representation of the loop filter is written in general as

$$\begin{aligned}\dot{\mathbf{z}}_{LF} &= \mathbf{A}_{LF} \cdot \mathbf{z}_{LF} + \mathbf{B}_{LF} \cdot \mathbf{u}_{LF} \\ y &= \mathbf{c}_{LF}^T \cdot \mathbf{z}_{LF} + \mathbf{d}_{LF}^T \cdot \mathbf{u}_{LF}\end{aligned}\tag{IV-39}$$

As example, the state space representation of a scalar fixed gain loop filter is given below. The derivation of this and the detailed derivation of all other state space representations are given in chapter III.

$$\begin{aligned}\dot{x}_1 &= [0] \cdot [x_1] + [1] \cdot [\delta\tilde{r}] \\ \Delta\hat{f}_{C/A} &= \left[\frac{f_{C/A} \cdot K_I}{T_{corr}} \right] \cdot [x_1] + \left[\frac{f_{C/A} \cdot K_P}{T_{corr}} \right] \cdot [\delta\tilde{r}]\end{aligned}\tag{IV-40}$$

IV - 2.11 Centralized state space representation

Based on the developed distributed state space representation, a centralized state space representation can be developed by combining the differential equations of the correlation core and the tracking filter in one state space matrix respectively linearized 1st order differential equation system. The coupling between the correlation core and the tracking filter is realized within this matrix.

The centralized state space representation offers the advantage to evaluate the closed loop behavior between any input – output combination. The centralized state space representation is a MIMO system. It is possible, to calculate any possible SISO path, respectively transfer function of the system.

The centralized state space representation will be used for closed loop tracking bandwidth calculation, similar to [83], for eigenvalue calculation and for tracking stability verification

Like the distributed state space representation, also the centralized state space representation is developed in chapter III, in detail for every tracking architecture.

The equations below show an example centralized state space representation of a scalar fixed gain tracking loop, based on the previous distributed state space form from equations (IV-37) to (IV-40).

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{u}\tag{IV-41}$$

The matrices **A** , **B** can be time dependent if online gain scheduling is applied.

$$\begin{bmatrix} \delta \dot{\tilde{\tau}} \\ \Delta \dot{\hat{\chi}} \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2 \cdot T_{C/A0}}{T_{corr}} & 0 \\ \frac{f_{C/A0} \cdot K_p(t)}{T_{corr}} & 0 & \frac{f_{C/A0} \cdot K_i(t)}{T_{corr}} \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi} \\ x_1 \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot T_{C/A0}}{T_{corr}} \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi}_{SV} \quad (IV-42)$$

$$\Delta \tilde{\rho} = \mathbf{c}_{\Delta \rho}^T \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi} \\ x_1 \end{bmatrix} = \begin{bmatrix} 0 & \frac{c}{f_{C/A0}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{\tau} \\ \Delta \hat{\chi} \\ x_1 \end{bmatrix}$$

IV - 2.12 Closed loop tracking bandwidth

The closed loop tracking bandwidth can be derived by using the already introduced centralized state space representation. In [65], a parametric tracking formulation is used for bandwidth calculation, but without the consideration of the correlation core and only for a scalar Kalman filter based tracking loop.

The bandwidth calculation derived in this chapter considers the correlation core and is able to calculate the closed loop tracking bandwidth of all tracking channels in case of vector tracking due to a state space approach.

The centralized state space representation is a MIMO system. In order to get the closed loop tracking bandwidth for a single satellite channel, a SISO path through this MIMO system is needed. In this case, the transfer function from the received code phase $\Delta \tilde{\chi}_{SV}$ to the estimated pseudorange $\Delta \tilde{\rho}$ is of interest.

$$G(t): \Delta \tilde{\chi} \rightarrow \Delta \hat{\rho} \quad (IV-43)$$

The bandwidth is defined as the frequency, where the closed loop amplification is 3 dB less than at zero frequency. Bode diagrams are typically normed to have a gain of 0 dB at 0 Hz. This is realized by using the same physical unit for the input and output variable. As closed loop tracking transfer function therefore, the following equation is used

$$G(t): \Delta \tilde{\chi} \rightarrow \Delta \hat{\rho} \cdot \frac{1}{\lambda_{C/A}} = \Delta \tilde{\chi} \rightarrow \Delta \hat{\chi} \quad (IV-44)$$

The introduced matrices can be used according to [32, p. 235], to calculate the closed loop transfer function of the tracking loop.

$$\Delta\tilde{\chi} \rightarrow \Delta\hat{\rho}: \quad G = \mathbf{c}_{\Delta\rho}^T(t) \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A}(t))^{-1} \cdot \mathbf{b}_{\Delta\tilde{\chi}}(t) = \begin{bmatrix} 0 & \frac{c}{f_{C/A0}} & 0 \end{bmatrix} \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A}(t))^{-1} \cdot \begin{bmatrix} \frac{2 \cdot T_{C/A0}}{T_{corr}} \\ 0 \\ 0 \end{bmatrix} \quad (\text{IV-45})$$

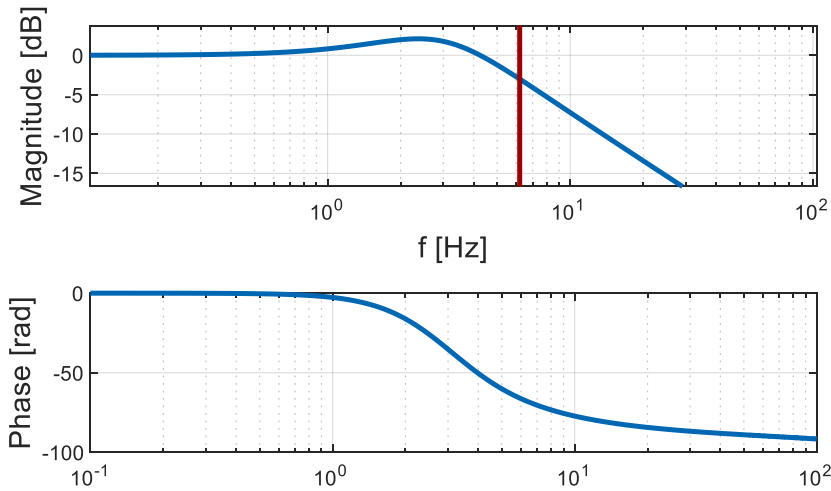
$$\Delta\tilde{\chi} \rightarrow \Delta\hat{\chi}: \quad G(t, s) = \frac{1}{\lambda_{C/A}} \cdot \mathbf{c}_{\Delta\rho}^T(t) \cdot (\mathbf{s} \cdot \mathbf{I} - \mathbf{A}(t))^{-1} \cdot \mathbf{b}_{\Delta\tilde{\chi}}(t) \quad (\text{IV-46})$$

The following equation shows the transfer function of the scalar fixed gain tracking loop.

$$G(s)_{\Delta\tilde{\chi}, \Delta\hat{\chi}, \text{ScalarFixedGain}} = \frac{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr}}{0.5 \cdot s^3 \cdot T_{corr} + s^2 \cdot T_{corr} + K_p \cdot s + K_i} \quad (\text{IV-47})$$

The controller gains are selected to get a closed loop tracking bandwidth of 6 Hz. In this example, $K_p = 0.0042$ and $K_i = 0.009$. A detailed derivation of the used gains is shown in detail in chapter III - 2.

The following figure shows the corresponding bode diagram of the closed loop transfer function.



IV-29 Bode diagram - 6 Hz tuning

For tracking bandwidth simulation along trajectory, the centralized state space representation respectively its system matrices are calculated at defined points along trajectory or at defined discrete times.

IV - 2.13 Eigenvalue based stability analysis

Besides the closed loop tracking bandwidth, also the poles of code and carrier tracking loops are very important regarding stability analysis.

One approach to get the poles of the closed loop tracking function, is to calculate the eigenvalues of the centralized state space matrix $\mathbf{A}(t)$.

Because the centralized state space system is a MIMO system, not all eigenvalues may be also poles of the corresponding SISO transfer function.

In the following derived eigenvalue calculation, the closed loop centralized state space representation of a **scalar optimal total state code tracking loop** (TSS) is used, which is derived in detail in chapter III - 3.5.3.

$$\begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi}_{NCO} \\ \Delta\hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} = \begin{bmatrix} \frac{-2}{T_{corr}} & \frac{-2}{f_{CA} \cdot T_{corr}} & -\frac{2}{c \cdot T_{corr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{k_{DLL,11}(t)}{T_{corr}} & 0 & 0 & 1 & 0 \\ \frac{k_{DLL,21}(t)}{T_{corr}} & 0 & 0 & 0 & 1 \\ \frac{k_{DLL,31}(t)}{T_{corr}} & 0 & 0 & 0 & -\frac{1}{\tau_{aLoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta\tilde{\tau} \\ \Delta\hat{\chi}_{NCO} \\ \Delta\hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} \frac{2}{f_{CA} \cdot T_{corr}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta\tilde{\chi} \quad (IV-48)$$

The matrix \mathbf{A} has full rank with 5 eigenvalues $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$. These eigenvalues are calculated, using Matlab.

Considering the poles in figure IV-30, the closed loop tracking transfer function

$$\mathbf{G} = \frac{\Delta\hat{\chi}}{\Delta\tilde{\chi}} \quad (IV-49)$$

has only three poles, whereas there are 5 eigenvalues. The reason is that with the chosen output $y = \Delta\hat{\rho}$ respectively $y = \Delta\hat{\chi}$, not all modes of the state space system can be observed. In order to identify the not observable modes, the time space solution of the differential equation given in (IV-48) is considered. It is important to evaluate even non observable eigen modes in order to identify possible instable internal states.

According to [84, p. 35], the general solution of the steady state closed loop state space differential system can be written as

$$\mathbf{y}(t) = \mathbf{C} \cdot e^{\mathbf{A}t} \cdot \hat{\mathbf{z}}_0 + \int_0^t \mathbf{C} \cdot e^{\mathbf{A}(t-\tau)} \cdot \mathbf{B} \cdot \mathbf{u}(\tau) d\tau + \mathbf{D} \cdot u(\tau) \quad (IV-50)$$

In this case, the initial condition is given as $\hat{\mathbf{z}}_0 = \mathbf{0}$. The solution is composed out of a free movement depending on the initial condition and a forced movement. The forced movement is calculated as a convolution of the impulse response $\mathbf{G}(t)$ and the input signal $\mathbf{u}(t)$.

$$\mathbf{y}(t) = \mathbf{G}(t) * \mathbf{u}(t) = \int_0^t \left(\mathbf{C} \cdot e^{\mathbf{A}(t-\tau)} \cdot \mathbf{B} + \mathbf{D} \cdot \delta(t-\tau) \right) \cdot \mathbf{u}(\tau) d\tau \quad (\text{IV-51})$$

In Laplace space, the output is calculated using the well-known transfer function

$$\mathbf{y}(s) = \mathbf{G}(s) \cdot \mathbf{u}(s) = \mathbf{C} \cdot (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} \quad (\text{IV-52})$$

The impulse response is the result, if the system is excited with an impulse.

The impulse response can be written in its eigenspace. In the eigenspace representation, the eigenmodes of the closed loop state space system are becoming apparent. The following equation shows the transformation of the transfer function matrix into eigen space representation.

$$\begin{aligned} \mathbf{G}(t) &= \mathbf{C} \cdot \mathbf{V} \cdot \mathbf{V}^{-1} \cdot e^{\mathbf{A}_{TDL} t} \cdot \mathbf{V} \cdot \mathbf{V}^{-1} \cdot \mathbf{B} + \mathbf{D} \cdot \delta(t-\tau) \\ &= \tilde{\mathbf{C}} \cdot \text{diag}(e^{\lambda_i t}) \cdot \tilde{\mathbf{B}} + \mathbf{D} \cdot \delta(t-\tau) \end{aligned} \quad (\text{IV-53})$$

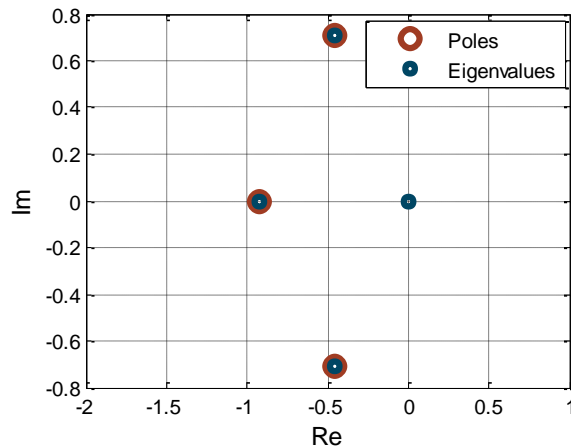
The SISO transfer function $g_{\Delta\hat{p},\Delta\tilde{z}}$ respectively $g_{\Delta\tilde{z},\Delta\tilde{z}}$ is given in eigenspace representation as:

$$g_{\Delta\tilde{z},\Delta\tilde{z}} = \tilde{\mathbf{C}}^T \cdot \text{diag}(e^{\lambda_i t}) \cdot \tilde{\mathbf{b}} = \sum_{l=1}^n \tilde{c}_{1,l} \cdot \tilde{b}_{l,1} \cdot e^{\lambda_l t} \quad (\text{IV-54})$$

Equation (IV-54) shows, that only such eigenvalues are observable at the output for which $\tilde{c}_{1,l} \cdot \tilde{b}_{l,1} \neq 0$ is valid.

$$g_{\Delta\tilde{z},\Delta\tilde{z}} = [0 \ 0 \ 1 \ 0 \ 0] \cdot \mathbf{V} \cdot \text{diag}(e^{\lambda_i t}) \cdot \mathbf{V}^{-1} \cdot \begin{bmatrix} 2 \\ T_{\text{corr}} \cdot f_{C/A} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{IV-55})$$

The following figure shows the poles and the eigenvalues for the code tracking loop.



IV-30 Poles and eigenvalues of optimal gain DLL

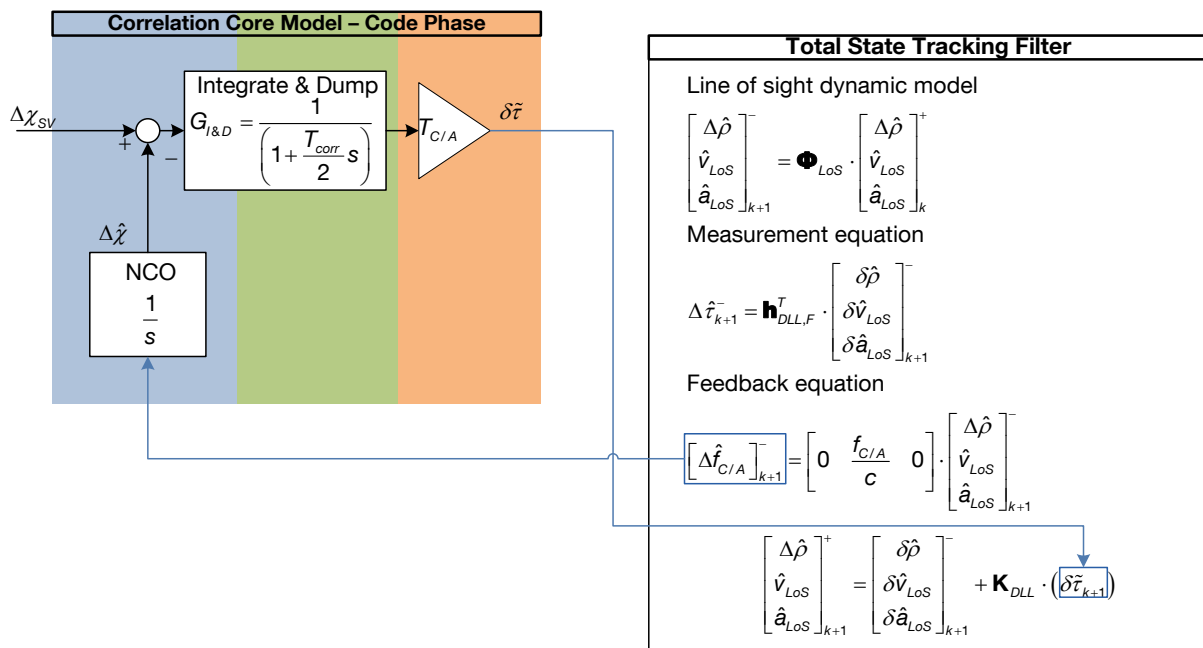
An additional pole and eigenvalue is located at -2000. But this pole respectively eigenvalue is neglected, due to its fast decaying influence.

Example of an instable tracking architecture

The previous optimal scalar tracking architecture uses the estimated pseudorange in the feedback path for NCO control.

All eigenvalues are in the left region – the tracking loop is stable.

Instead of using the estimated pseudorange, it would be also possible to use the estimated line of sight velocity \hat{v}_{LoS} in the feedback path to control the replica generation, as it is shown in the tracking architecture of figure IV-31.

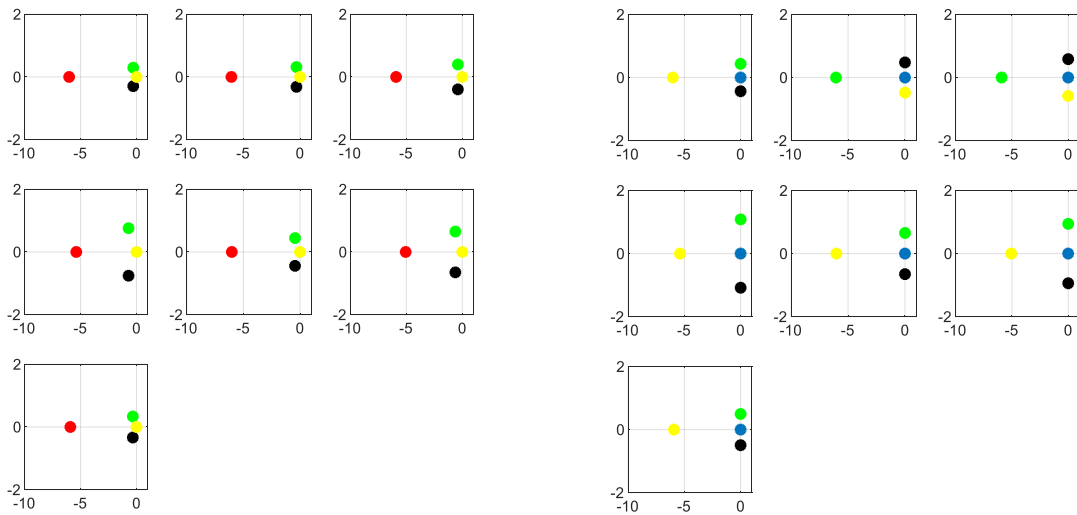


IV-31 Alternative feedback structure - Scalar optimal gain total state DLL

The corresponding closed loop state space representation is given below.

$$\begin{bmatrix} \delta \dot{t} \\ \Delta \dot{\hat{\chi}}_{NCO} \\ \Delta \dot{\hat{\rho}} \\ \dot{\hat{v}}_{LoS} \\ \dot{\hat{a}}_{LoS} \end{bmatrix} = \begin{bmatrix} -\frac{2}{T_{corr}} & -\frac{2}{T_{corr} \cdot f_{C/A}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{f_{C/A}}{c} & 0 \\ \frac{k_{DLL,11}}{T_{corr}} & 0 & 0 & 1 & 0 \\ \frac{k_{DLL,21}}{T_{corr}} & 0 & 0 & 0 & 1 \\ \frac{k_{DLL,31}}{T_{corr}} & 0 & 0 & 0 & -\frac{1}{\tau_{aLoS}} \end{bmatrix} \cdot \begin{bmatrix} \delta \tilde{t} \\ \Delta \hat{\chi}_{NCO} \\ \Delta \hat{\rho} \\ \hat{v}_{LoS} \\ \hat{a}_{LoS} \end{bmatrix} + \begin{bmatrix} \frac{2}{T_{corr} \cdot f_{C/A}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta \tilde{\chi}_{SV} \quad (IV-56)$$

The following figure shows the eigenvalues respectively poles of both feedback variants from (IV-48) and (IV-56).

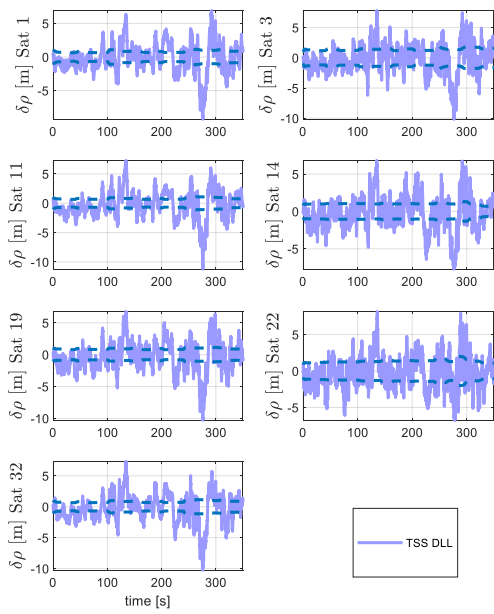


IV-32 Eigenvalues - rho feedback (for all 9 satellites being used in tracking)

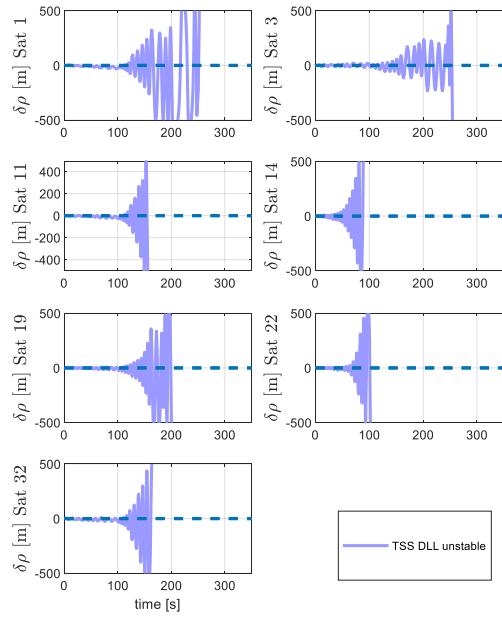
IV-33 Eigenvalues - vLoS feedback (for all 9 satellites being used in tracking)

Figure IV-33 shows that using the estimated line of sight velocity for NCO control, gives an instable behavior. There are poles on the imaginary axis.

The following time space simulation confirms this instable behavior.



IV-34 stable tracking behavior



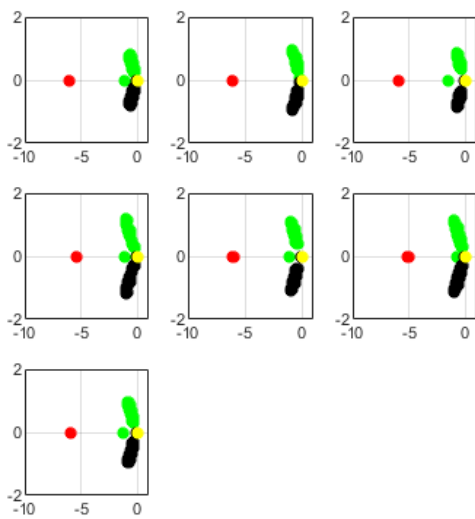
IV-35 instable tracking behavior

This example shows the importance of this approach, using a centralized closed loop state space representation of tracking loops for stability evaluation. Besides the basic tracking and feedback architecture, also different tuning settings can be evaluated regarding tracking stability.

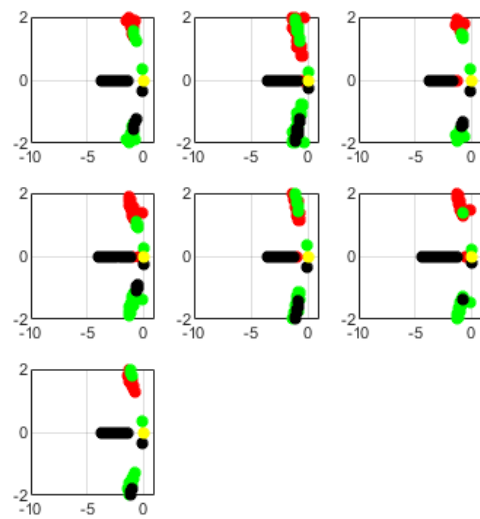
As already mentioned, not all eigenvalues may be at the same time be poles of the SISO transfer function $\Delta\tilde{x} \rightarrow \Delta\hat{x}$. But for stability verification it is useful, to consider all eigenvalues and not only the poles of the SISO transfer function, in order to be sure, that no internal instability would lead to badly scaled matrices, especially in case of used optimal filters or even vector tracking filters.

The introduced eigenvalue stability analysis can also be used for “along track pre mission stability evaluation”. For that, at defined points on trajectory or with a defined sampling rate, the eigenvalues are calculated, using the Kalman gains together with the centralized state space matrix, valid at the corresponding point.

The following figures show the pole movement along track, for the DA42 approach and the high dynamic trajectory, using as tracking architecture the scalar optimal filter-based approach (TSS).



IV-36 DA42 – TSS – Pole Movement



IV-37 High dynamic – TSS – Pole Movement

The pole movement is caused through changing Kalman gains along trajectory, due to changing dynamic parameters and changing signal to noise levels.

IV - 3 Software-based IF space segment simulator and IF GPS receiver

IV - 3.1 Introduction and motivation

In the forgoing section, an equivalent base band GPS receiver modeling and simulation concept was developed. There, only code and carrier phase dynamic together with satellite signal power, noise power and jammer power models were used for a realistic tracking simulation.

In this section, a software receiver and software based space segment simulator is introduced, in order to verify the phase dynamic simulation approach by real signal tracking.

The focus of this software based GPS receiver is to verify the tracking behavior and to verify pseudorange and positioning error covariance prediction, stated by the equivalent base band scheme. Therefore, an IF band approach is developed.

Within a software defined space segment simulator, the civil spreading code is generated together with a carrier at an IF frequency of almost 10 MHz.

For signal generation, at first the line of sight dynamic model, already used for phase dynamic calculation in the previous section, is also used here for code and carrier Doppler calculation. Also the already introduced satellite signal link budget model, introduced in the previous section is used to calculate the satellite signal power at an assumed antenna interface. The IF band space segment simulator generates the satellite signal as a GPS receiver would get at after the IF bandpass filter within the reception chain.

No navigation message is modulated upon the satellite signal, because the ephemeris data are already known and are used within the GPS receiver.

From the software defined GPS receiver only the correlation core and the tracking filter are implemented. The correlation core can directly use the generated satellite signal in IF band from the space segment simulator, because the signals equal the signals, a real receiver would get after down conversion, IF bandpass filtering and automatic gain control.

The correlation core and repique generation are partly based on [5], [85] and [86].

In order to facilitate an easy exchange of tracking filters, a novel architecture and integration scheme of tracking filter into the software receiver was developed. Moreover, a novel code phase pointer and carrier phase pointer control was developed, enabling a direct phase control interface for optimal filter based tracking.

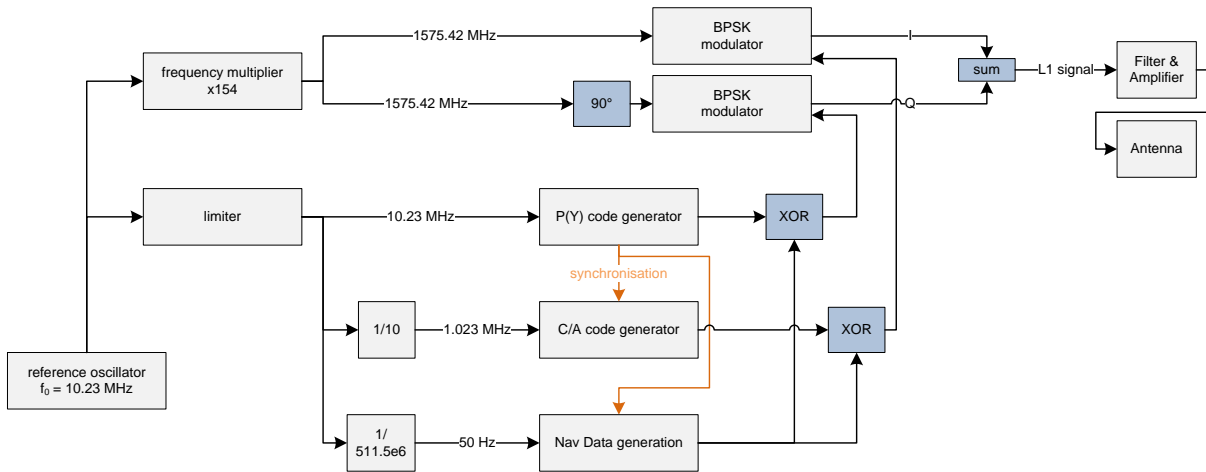
All tracking filters developed in chapter III can be integrated into the software receiver architecture.

No expensive high frequency equipment is necessary, while the possibilities regarding tracking behavior simulation are equal. Moreover, even multi frequency satellite simulations are possible and also multi jammer scenarios. Regarding jamming scenarios, any jamming signals can be stimulated.

IV - 3.2 IF band Space Segment Simulator

IV - 3.2.1 Structure of signal generation

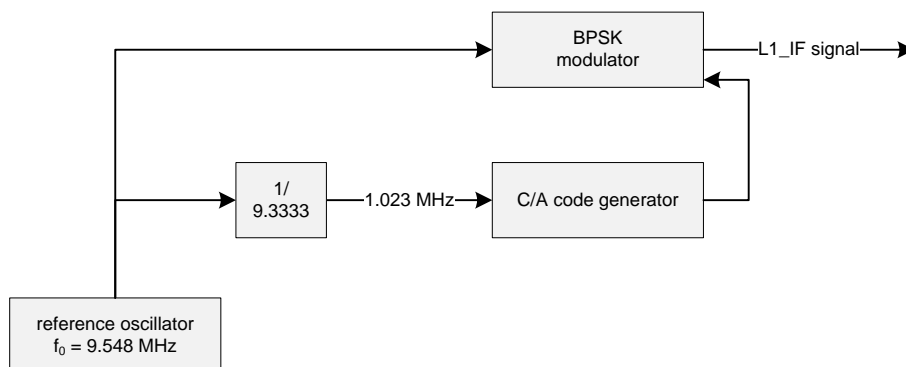
The following figure shows the satellite signal generation architecture, according to [5].



IV-38 Satellite signal generation - L1 signal

The given signal processing architecture only shows the L1 signal generation. The L1 contains in the in-phase component the C/A code and on the quadrature component the P(Y) signal.

For the IF band space segment simulator, the satellite signal processing part is given below.



IV-39 IF band satellite signal generation

The focus of this thesis is to analyze the C/A code tracking. Therefore, only the civil spreading code is generated. The IF band satellite signal is given as

$$s_{IF,sat}(t) = A_{sat} \cdot C(t) \cdot \sin(2\pi f_{IF} t) \quad (IV-57)$$

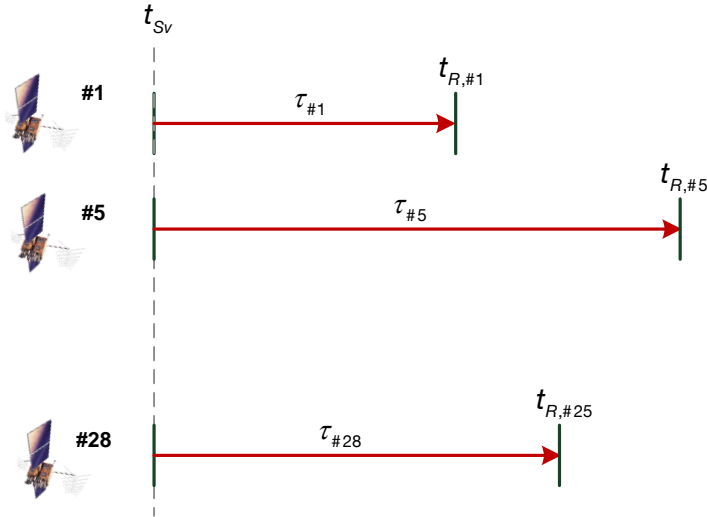
The given time t is the system reference time. The sampling frequency of the software space segment simulation is given as

$$f_s = 38.192 \text{ MHz} \quad (IV-58)$$

The sampling frequency was selected to be 4 times the IF frequency.

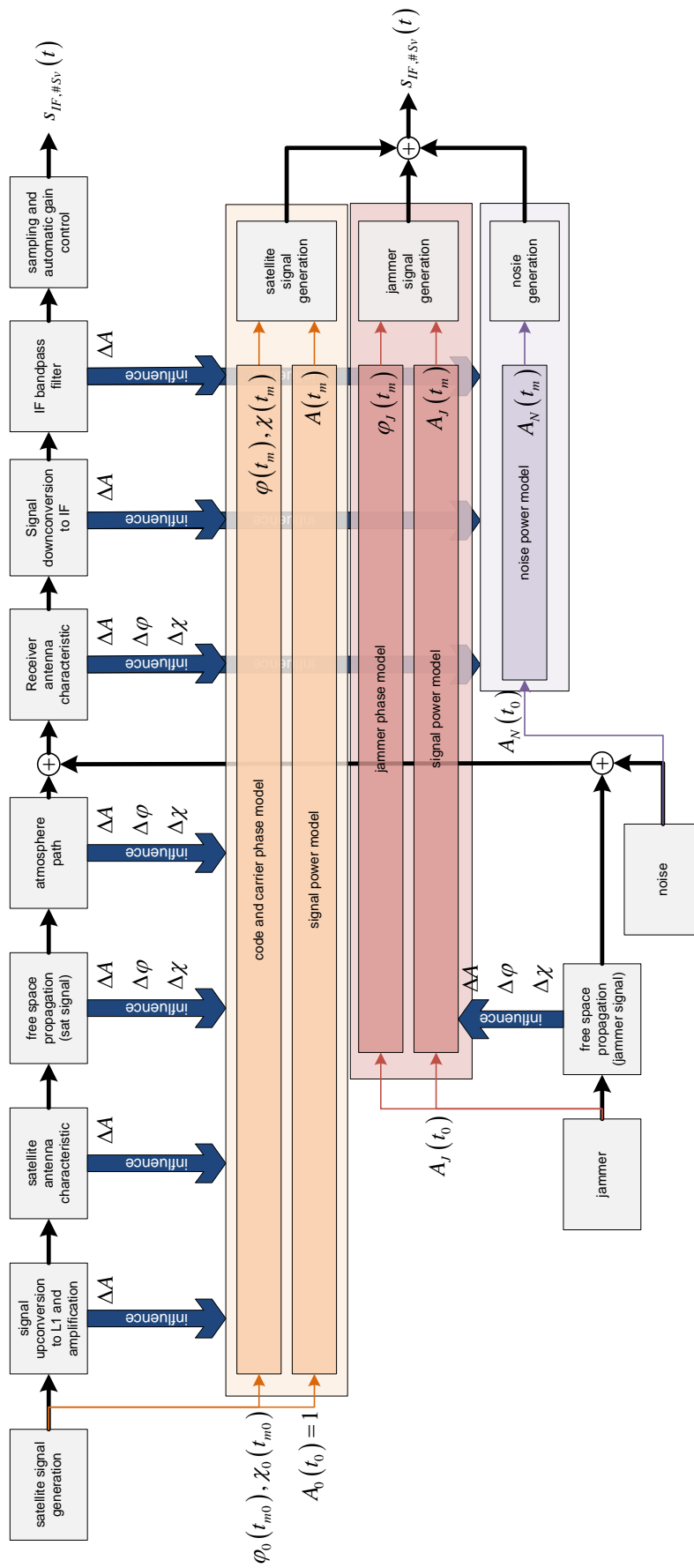
The system reference time t_s is at the same time for all satellites the signal transmission time t_{sv} . Without loss of generality, it is assumed that all satellites transmit the signals at the same common transmission time.

The reception time t_r at the GPS receiver differs due to the different distances between the GPS receiver and the corresponding satellites. The signal travel time is defined as τ .



IV-40 Transmission and reception time

The following figure compares the real signal transmission path with the space segment simulation. Given the space segment simulator, the signal is generated for the IF interface within the GPS receiver, whereby the signal power, code-phase and carrier-phase are calculated by a corresponding power transmission path model and phase transmission path model.

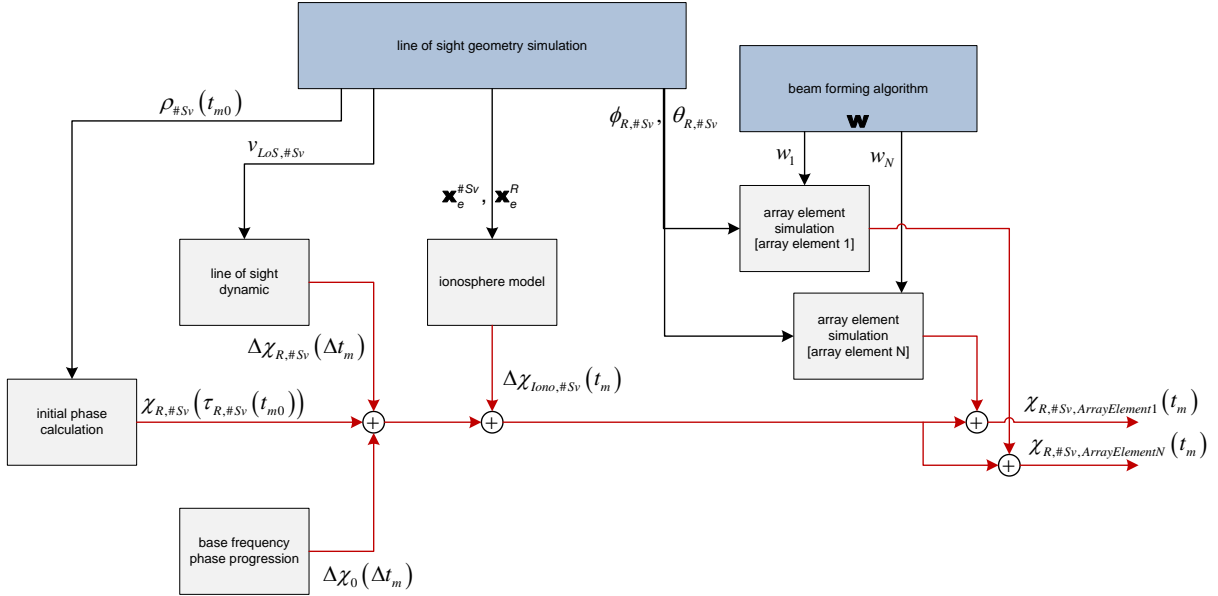


IV-41 Comparing real signal transmission and space segment simulation

The received signal at the receiver antenna interface can be written as

$$s_{R,\#Sv}(t) = A_{R,\#Sv} \cdot C_{R,\#Sv}(\chi_{R,\#Sv}(t)) \cdot \sin(\varphi_{R,\#Sv}(t)) \quad (IV-59)$$

For code phase $\chi_{R,\#Sv}(t)$ and carrier phase $\varphi_{R,\#Sv}(t)$ calculation, the actual range between the satellite and receiver at the simulation start time is necessary, together with the ongoing line of sight dynamic respectively line of sight velocity.



IV-42 Simulated code phase dynamic

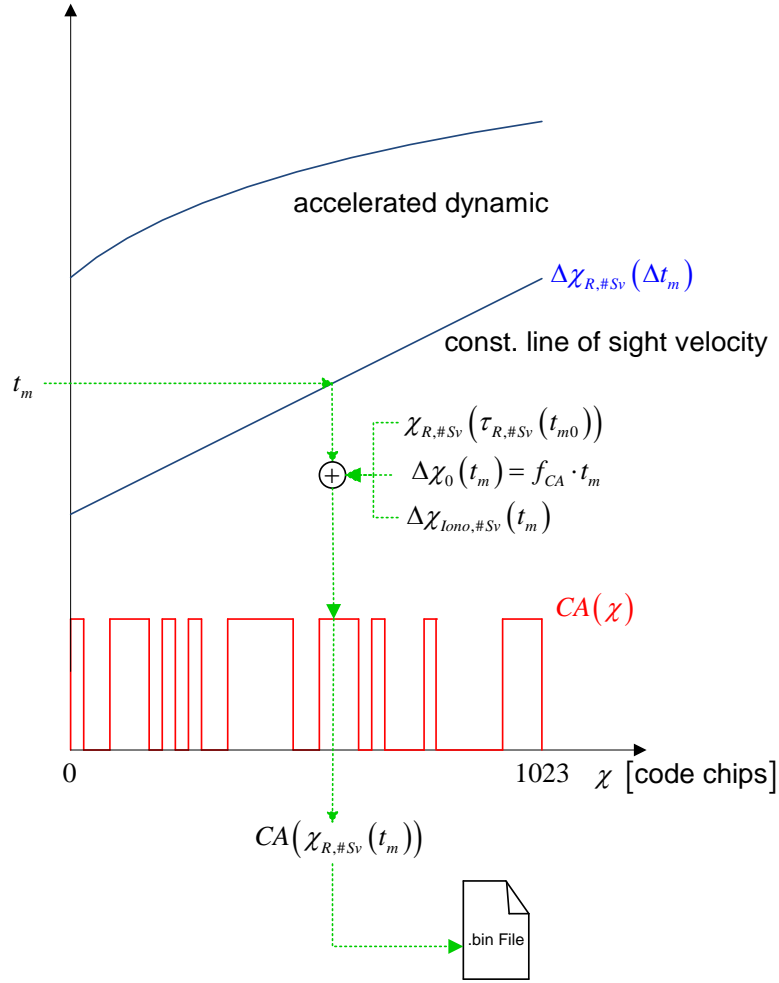
The mission start time is given as t_{m0} . At this time, the initial range is used to calculate the initial code phase $\chi_{R,\#Sv}(\tau_{R,\#Sv}(t_{m0}))$.

$$\chi_{R,\#Sv}(\tau_{R,\#Sv}(t_{m0})) = \frac{\rho_{\#Sv}(t_{m0})}{\lambda_{C/A}} = \frac{\rho_{\#Sv}(t_{m0}) \cdot f_{C/A}}{c} \quad (IV-60)$$

Along mission, the code phase changes due to the normal time progression $\Delta\chi_0(\Delta t_m)$ and due to Doppler frequency shift $\Delta\chi_{R,\#Sv}(\Delta t_m)$. To a smaller amount, the code phase changes also due to ionospheric caused frequency shifts $\Delta\chi_{Iono,\#Sv}(t_m)$.

IV - 3.2.2 Digital IF band signal synthetization

At first the CA code generation at base band will be discussed. The corresponding code-phase is derived according to the given phase transmission model.



IV-43 CA code generation

For each satellite, one CA code period is generated. One period consists out of 1023 chips. For each code chip, one value is generated, which gives in total 1023 values.

With the given sampling rate of $f_s = 38.192\text{MHz}$, the code phase is calculated as

$$\chi_{R,\#Sv}(t_m) = f_{CA} \cdot t_m + \frac{v_{LoS}(t_m)}{c} \cdot f_{CA} \cdot t_m + \Delta\chi_{Iono,\#Sv}(t_m) + \chi_{R,\#Sv}(\tau_{R,\#Sv}(t_{m0})) \quad (\text{IV-61})$$

$$\chi_{R,\#Sv}[n] = f_{CA} \cdot \frac{n}{f_s} + \frac{v_{LoS}\left(\frac{n}{f_s}\right)}{c} \cdot f_{CA} \cdot \frac{n}{f_s} + \Delta\chi_{Iono,\#Sv}\left(\frac{n}{f_s}\right) + \chi_{R,\#Sv}(\tau_{R,\#Sv}(0)) \quad (\text{IV-62})$$

The start mission time t_{m0} is assumed to be 0. The term $\chi_{R,\#Sv}(\tau_{R,\#Sv}(t_{m0}))$ describes the initial phase offset between transmitted phase and received phase based on the distance between satellite

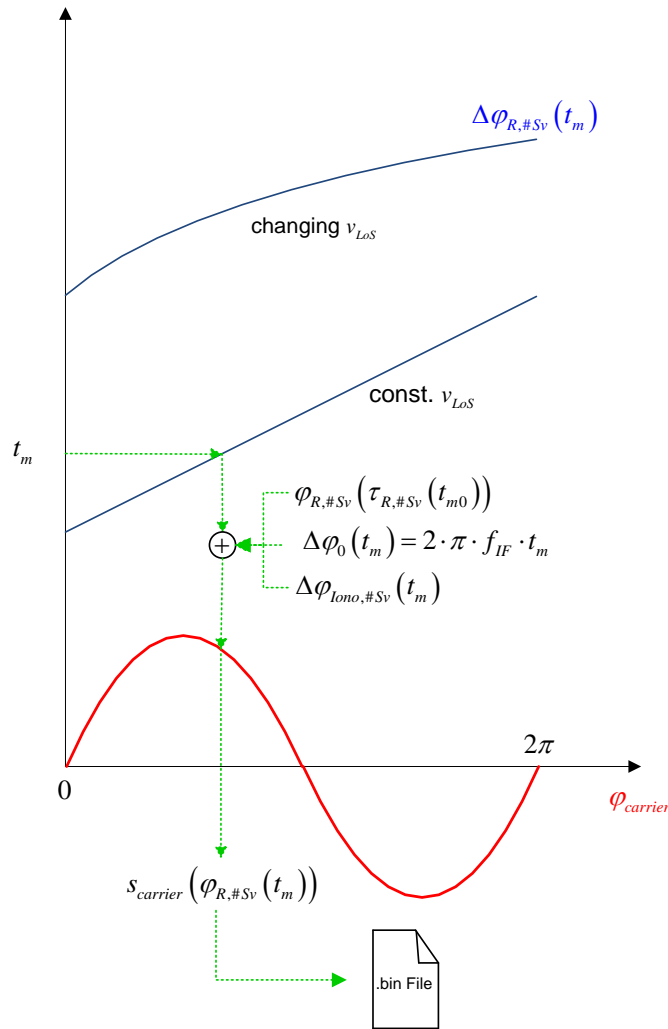
and GPS receiver.

As long as the actual code phase $\chi_{R,\#Sv}[n] < 1$, the value of the first CA code chip is stored in the file for each sample n . For $1 \leq \chi_{R,\#Sv}[n] < 2$, the value of the second CA code chip is stored in the file, also for each sample n .

After having generated the base band C/A code, it is upconverted to an intermediate frequency IF, using in this thesis an intermediate frequency of 9.5486 MHz.

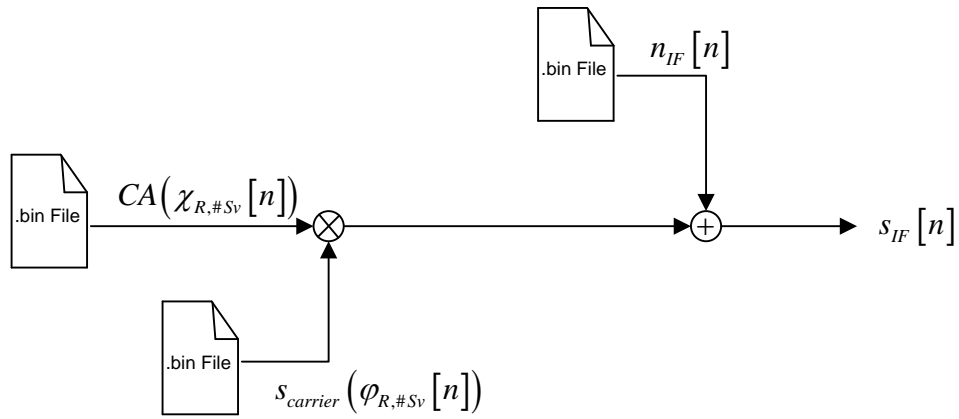
The carrier at IF frequency is generated in a similar way, also considering different types of phase influences.

$$\varphi_{R,\#Sv}(t_m) = 2 \cdot \pi \cdot f_{IF} \cdot t_m + \frac{v_{LoS}(t_m)}{c} \cdot 2 \cdot \pi \cdot f_{IF} \cdot t_m + \Delta\varphi_{Iono,\#Sv}(t_m) + \varphi_{R,\#Sv}(\tau_{R,\#Sv}(t_{m0})) \quad (IV-63)$$



IV-44 Carrier generation

The following figure shows the discrete up-conversion.



IV-45 Realization of discrete IF signal generation

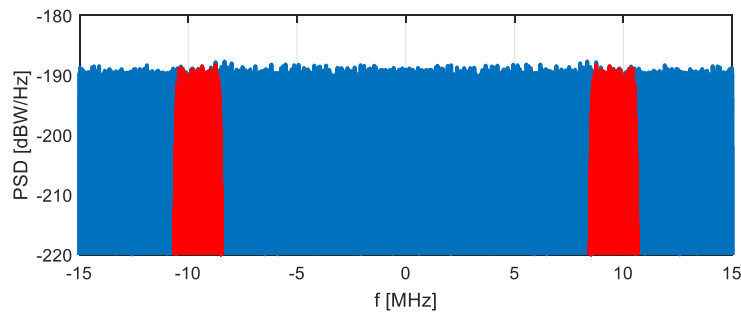
The added standard noise is already at IF interface and shaped according the IF bandwidth filter.

The standard noise generated as a normal distributed process $\mathcal{N}(n | \mu = 0, \sigma_n^2)$.

Given a wanted noise power spectral density N_0 , in discrete signal processing, the corresponding variance is calculated as

$$\sigma_n^2 = N_0 \cdot f_s \quad (\text{IV-64})$$

The variable f_s is the discrete sampling frequency, in this realization 38.192 MHz. The generated noise is filtered by a bandpass, similar to the IF bandpass filter.

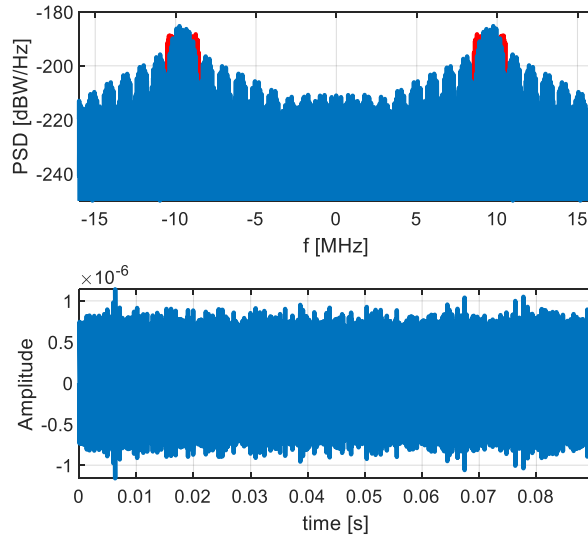


IV-46 noise power spectral density

The red colored noise is the filtered noise using an IF bandpass of 4 MHz, which corresponds 2 MHz in base band.

The following figure show's the spectral representation of the received satellite signal after IF filter.

The noise part in the spectral representation is highlighted in red.



IV-47 real IF signal

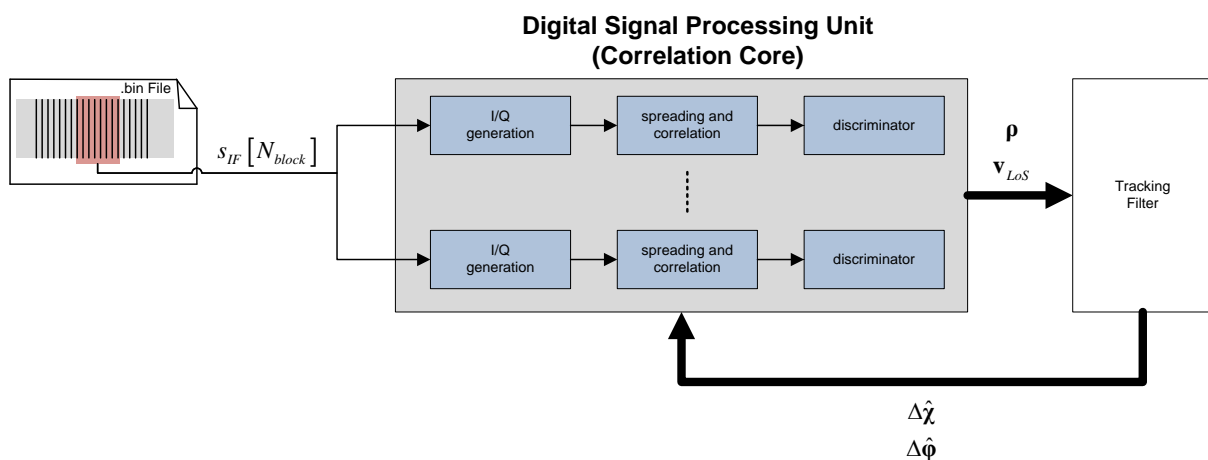
IV - 3.3 Software defined IF band GPS receiver with modular tracking kernels

IV - 3.3.1 Concept of IF band GPS software receiver

The IF band GPS software receiver takes sampled data at intermediate frequency for I/Q generation and correlation. For every satellite an own I/Q subsequent correlation unit is realized. The realized Matlab based software receiver subsequently processes one sample block of each satellite. A sample block consists out of all samples within one correlation period.

$$N_{block} = T_{corr} \cdot f_s \quad (IV-65)$$

In a FPGA realization, the I/Q and correlation units could be also realized in parallel for better performance.



IV-48 Basic structure Matlab based software receiver

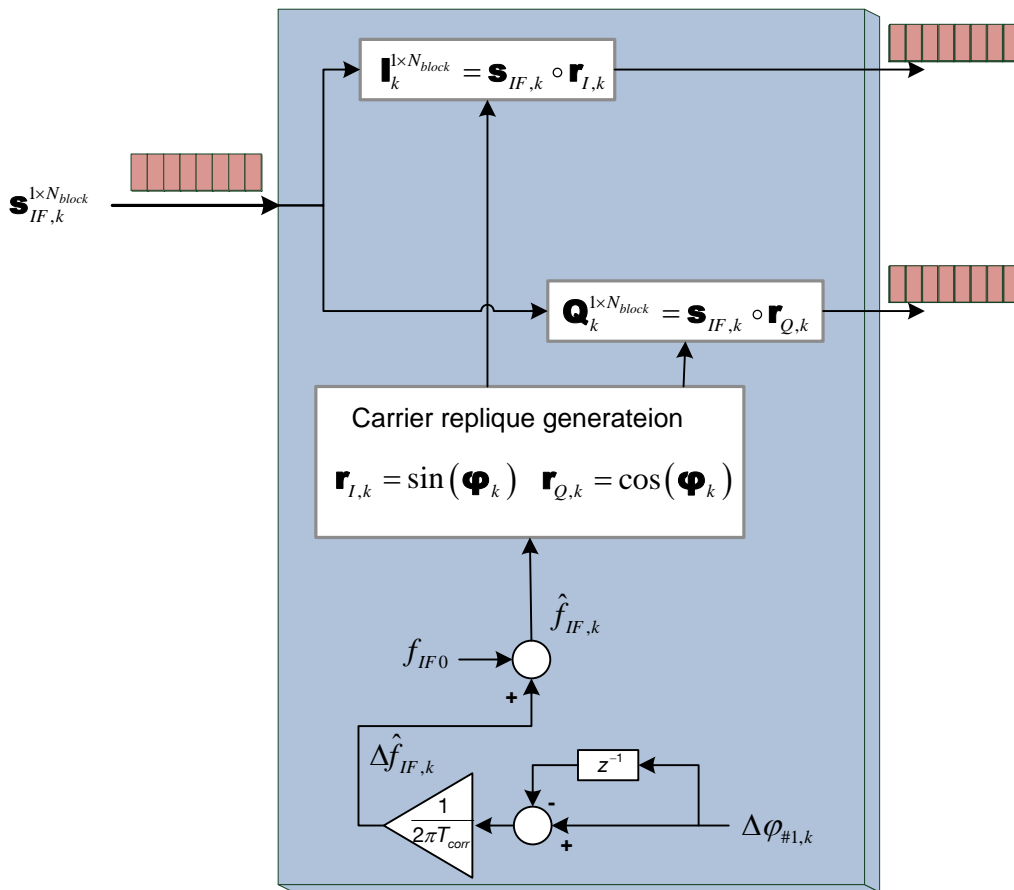
The actual part of the software receiver is the correlation core. This part realizes all digital signal processing tasks. The subsequent tracking filter is equivalent with the equivalent base band realization.

The focus of this thesis was to develop and analyze different tracking filter and to improve antijam and evaluate the jammer impact at different stages within the GPS receiver, only the tracking part of the software receiver was developed. The navigation and acquisition parts are not considered.

Because the developed software receiver is used together with the software based IF band space segment simulator, the initial code and carrier phase relations are well known, which make acquisition redundant. Also the satellite positions are well known, because they are already used in the software based space segment simulator for data generation. Therefore, also the navigation unit, extracting ephemeris data from the navigation message is not necessary.

IV - 3.3.2 I/Q generation and Correlation

The signal processing within the I/Q generation is also organized block-wise.



IV-49 Block-wise I/Q generation

The input signal $\mathbf{s}_{IF,k}^{1 \times N_{block}}$ is a row vector, having a length of N_{block} . The replicated carrier signals $\mathbf{r}_{I,k}^{1 \times N_{block}}$ and $\mathbf{r}_{Q,k}^{1 \times N_{block}}$, are also vector having the size of N_{block} .

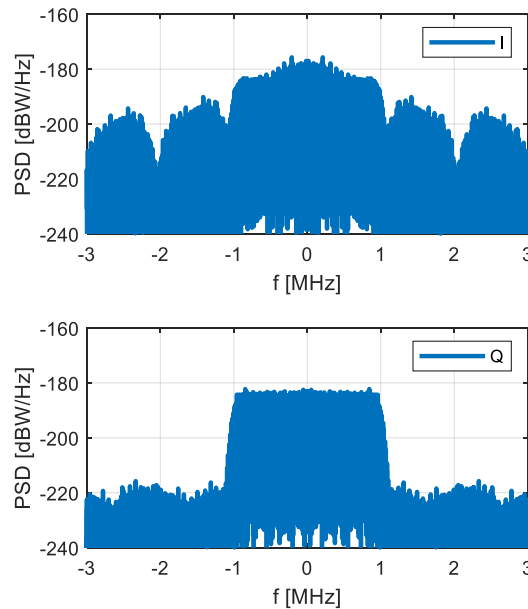
The operation $\mathbf{I}_k^{1 \times N_{block}} = \mathbf{s}_{IF,k} \circ \mathbf{r}_{I,k}$ and $\mathbf{Q}_k^{1 \times N_{block}} = \mathbf{s}_{IF,k} \circ \mathbf{r}_{Q,k}$ realizes a bit-wise multiplication within the given signal block k .

The phase $\boldsymbol{\varphi}_k$ is also a vector with N_{block} entries. This phase vector is generated as given next.

$$\begin{aligned} \boldsymbol{\varphi}_k &= \left(\mathbf{t} \cdot 2 \cdot \pi \cdot \hat{f}_{IF} \right) + \Delta \varphi_{k-1} + \varphi_0 \\ \Delta \varphi_k &= \text{mod}(\boldsymbol{\varphi}_k(N_{blk}), 2\pi) \\ \mathbf{t}^{1 \times N_{blk}} &= \begin{bmatrix} 0 & \frac{1}{f_s} & \dots & \frac{N_{blk} - 1}{f_s} \end{bmatrix} \end{aligned} \tag{IV-66}$$

As already mentioned, in case of synchronized carrier replique, only the I signal contains the C/A code, whereas the Q signal contains only noise.

The following example show's the power spectral density of the I and Q signal.



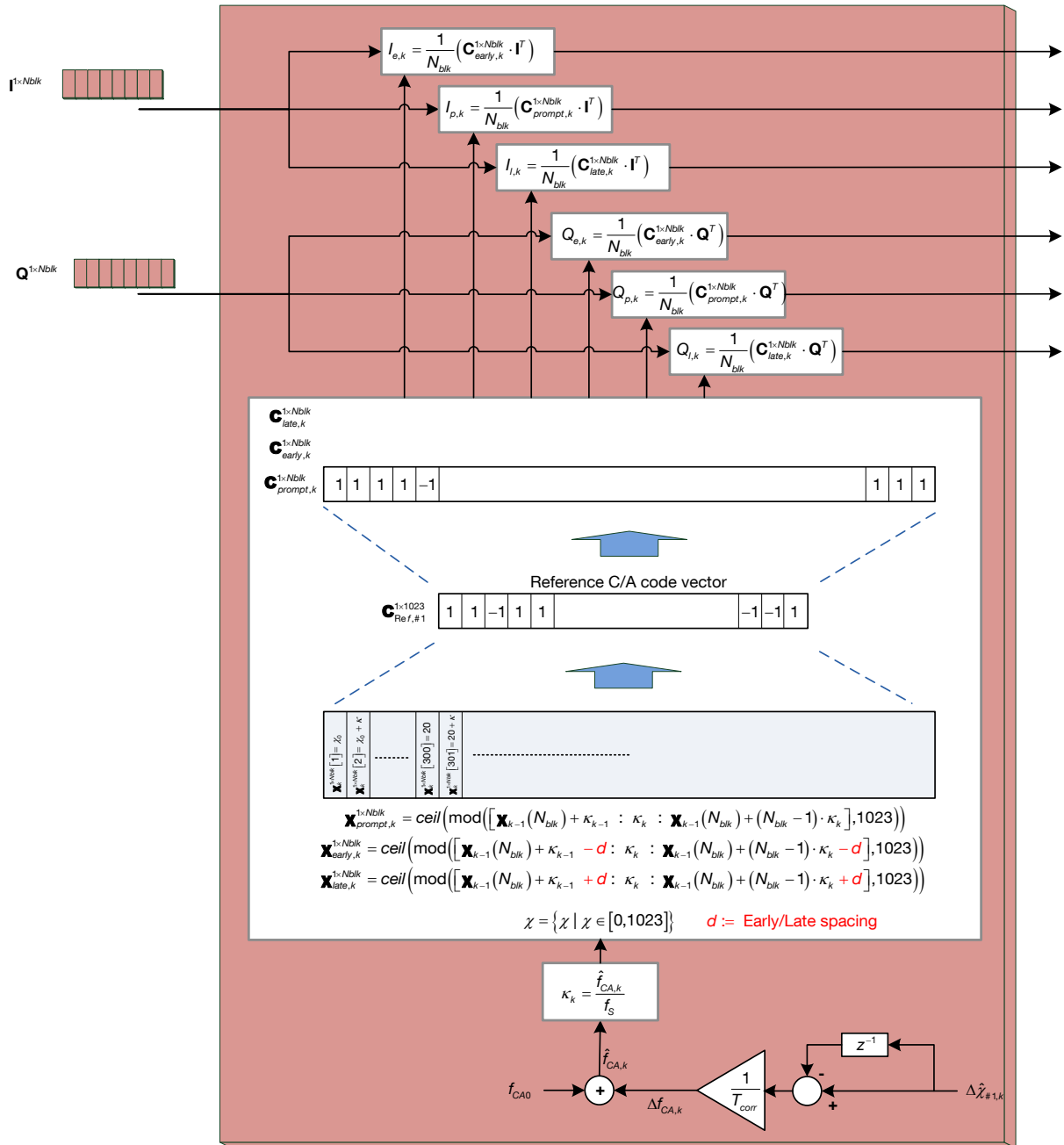
IV - 3.3.3 Spreading and Correlation

After I/Q generation, the so called “spreading operation” follows. There, the received signal is multiplied with a replicated version of the C/A code. The received C/A code part within the received signal is compressed to 0 Hz, whereas all other non-wanted signals are spread to a wider frequency spectrum.

After the “spreading operation”, the correlation follows. This correlation equals in frequency space a low pass filter with a cut-off frequency of $\frac{1}{T_{corr}}$. The compressed C/A code signal passes without any damping the low pass filter. Because all other unwanted signals are spread over a wider

frequency spectrum, the power spectral density in average is lowered. Therefore, within the passage area of the low pass filter, only a small part of the original unwanted signal power remains.

The following figure shows the software receiver realization of spreading and correlation.



IV-50 Block-wise spreading and correlation

The estimated code phase at time k is given as $\Delta \hat{\chi}_k$. For frequency correction, the code phase variation between time k and time $k-1$ is needed, being calculated by using a discrete delay element. The resulting frequency correction is given as

$$\Delta \hat{f}_{CA,k} = \frac{\Delta \hat{\chi}_k - \Delta \hat{\chi}_{k-1}}{T_{corr}} \quad (IV-67)$$

The actual C/A code frequency, needed for replica generation gets

$$f_{CA,k} = f_{CA} + \Delta \hat{f}_{CA,k} \quad (IV-68)$$

Instead of using a classic NCO, the phase progress per time sample is calculated as

$$\kappa_k = \frac{\hat{f}_{CA,k}}{f_s} \quad (IV-69)$$

Using κ_k , the actual code phase for replica generation is calculated according to next equation.

$$\hat{\chi}_{k+1} = \hat{\chi}_k + \kappa_k \quad (IV-70)$$

Due to the block wise processing, a vector containing the code phases for all block samples is generated.

$$\mathbf{X}_{prompt,k}^{1 \times N_{blk}} = \text{ceil} \left(\text{mod} \left(\left[\mathbf{X}_{k-1}(N_{blk}) + \kappa_{k-1} : \kappa_k : \mathbf{X}_{k-1}(N_{blk}) + (N_{blk} - 1) \cdot \kappa_k \right], 1023 \right) \right) \quad (IV-71)$$

The code phase $\mathbf{X}_{k-1}(N_{blk})$ is the last entry of the generated code phase vector from the previous block. Similar vectors are generated for the early and late code replica.

The calculated code phase vector $\mathbf{X}_{prompt,k}^{1 \times N_{blk}}$ has in the given realization $N_{blk} = 38192$ samples.

The time difference between two samples is in the given SW receiver, $T_s = 26.183\text{ns}$. For all samples $\hat{\chi}$ of the code phase vector the following expression is valid: $\{\hat{\chi} | 0 \leq \hat{\chi} \leq 1023\}$, with $\hat{\chi} \in \mathcal{Q}$.

In the next step, all elements $\hat{\chi}$ of the code phase vector are mapped onto the reference C/A code vector, which defines for each element $\chi_{CA} \in \mathcal{N}_0$ a binary value of +1 or -1. All samples between for example $100 \leq \hat{\chi}_k < 101$, are mapped onto the same C/A code reference sample.

The resulting C/A replica vectors are given as

$$\mathbf{C}_{early,k}^{1 \times N_{blk}}, \mathbf{C}_{prompt,k}^{1 \times N_{blk}}, \mathbf{C}_{late,k}^{1 \times N_{blk}} \quad (IV-72)$$

with each containing N_{blk} samples.

The correlation is realized by the dot product between the replicated C/A code vector and the corresponding I or Q signal vector generated in the previous stage.

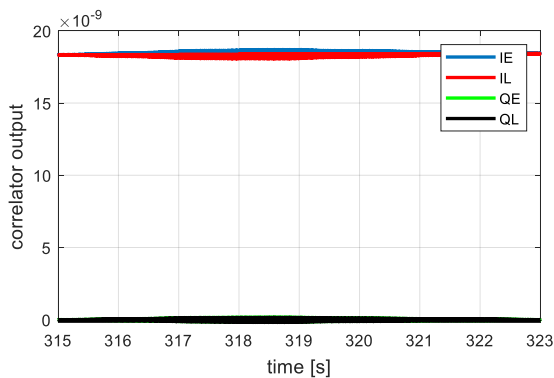
$$I_{p,k} = \frac{1}{N_{blk}} (\mathbf{c}_{prompt,k}^{1 \times N_{blk}} \cdot \mathbf{I}^T)$$

$$Q_{p,k} = \frac{1}{N_{blk}} (\mathbf{c}_{prompt,k}^{1 \times N_{blk}} \cdot \mathbf{Q}^T)$$
(IV-73)

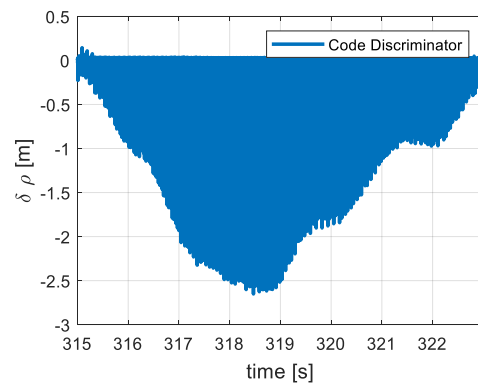
Similar dot products are also calculated for the early and late C/A replicas.

The following examples give an impression of the early and late signals in I and Q path after correlation and already the corresponding early-late discriminator output, normed to range.

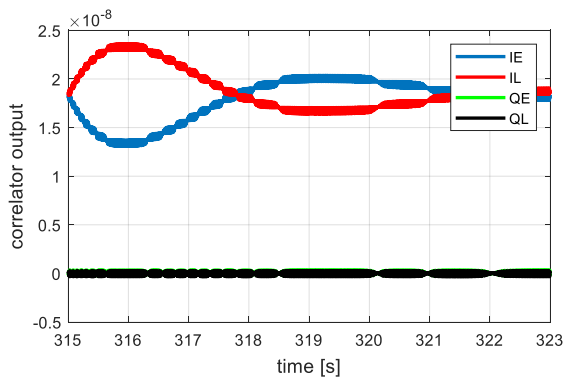
In order to show the pure dynamic behavior, tracking without noise was realized. For that, within the space segment simulator, a received GPS signal without noise was generated.



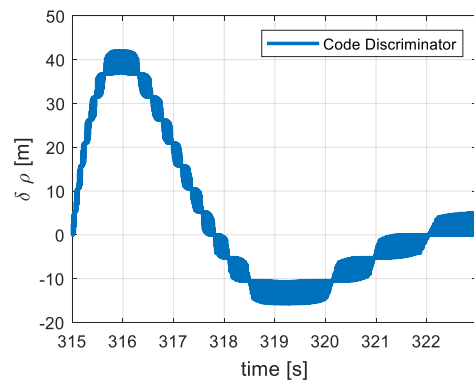
IV-51 Correlator output - no noise, perfect initialization



IV-52 Discriminator output - no noise, perfect initialization



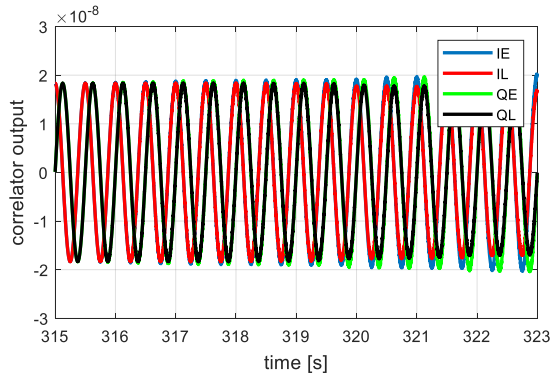
IV-53 Correlator output - no noise, false initialization



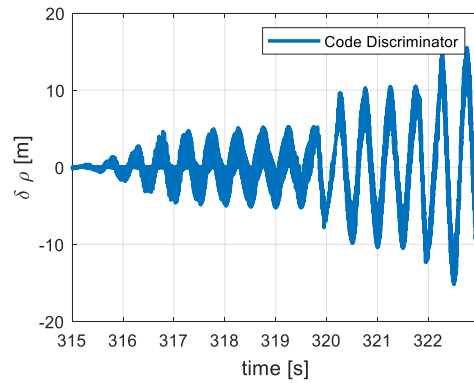
IV-54 Discriminator Output - no noise, false initialization

Given a perfect synchronized carrier tracking loop, all information is within the I path. The correlator outputs for the Q path remain zero.

The next simulation shows the correlator outputs, given a not synchronized carrier tracking loop. It gets obvious that the information oscillates between the I and Q path.



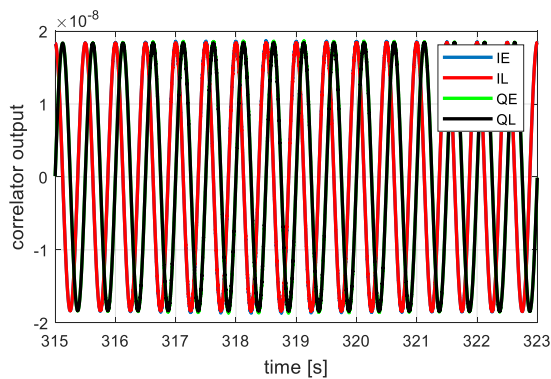
IV-55 Correlator output - no noise, unsynchronized PLL



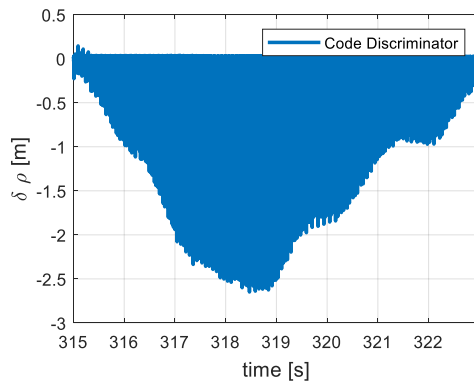
IV-56 Discriminator Output - no noise, unsynchronized PLL

This example provides some further interesting details. As discriminator, again the early-late discriminator was used. But in case of a not synchronized carrier tracking loop, early-late power discriminators must be used in order to compensate the carrier phase influence. The early-late discriminator gives wrong code phase error measurements, which lead to a general loop divergence. This can be seen by a divergent behavior between IE and IL and between QE and QL.

If an early-late power discriminator is used, the missing carrier synchronizing does not have any effect on the code phase measurement.

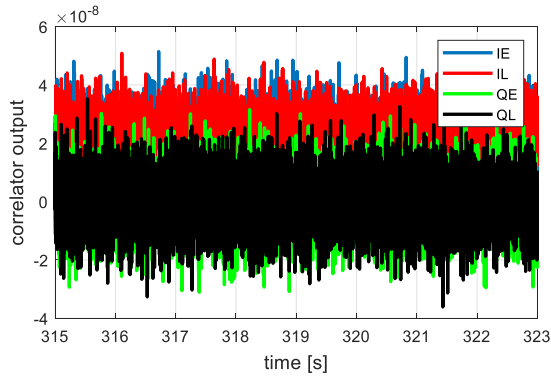


IV-57 Correlator output - no noise, unsynchronized PLL, early-late-power discriminator

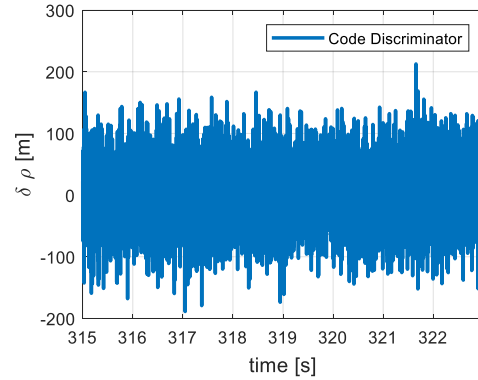


IV-58 Discriminator Output - no noise, unsynchronized PLL, early-late-power discriminator

The next example shows the scenario, given a synchronized carrier tracking loop and real received GPS signals in a standard noise environment.



IV-59 Correlator output – real noise

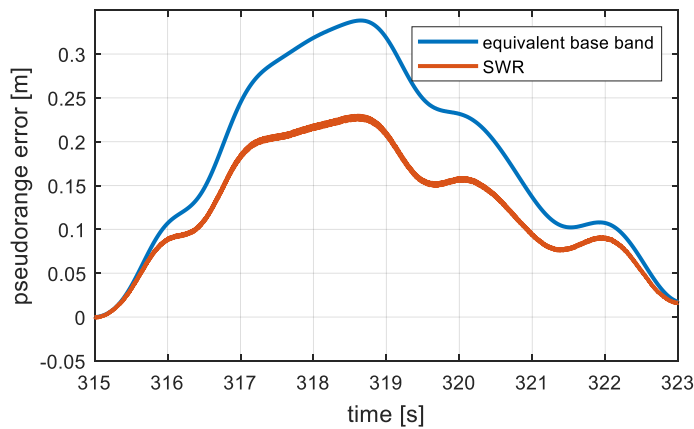


IV-60 Discriminator Output – real noise

IV - 3.3.4 Short example tracking

In this section, an 8 second piece of the DA42 trajectory will be considered. Real GPS signals are generated for this short mission excerpt. In order to emphasize the similarity between equivalent base band simulation and software receiver-based tracking, noise free GPS signals for IF interface are generated, using the IF space segment simulator.

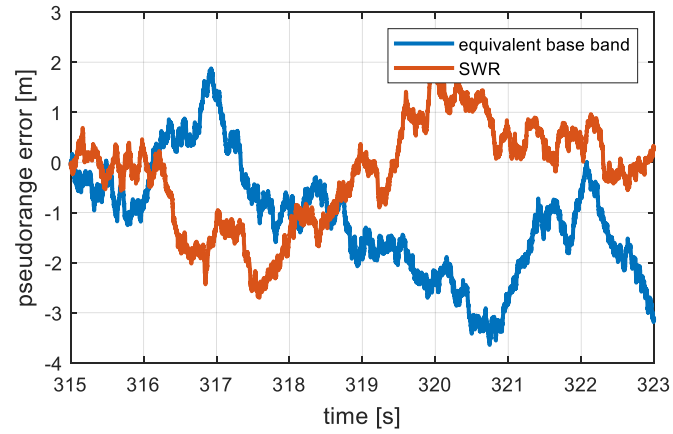
The following figure compares the pseudorange error for one satellite – in this example, satellite 19 – of equivalent base band tracking and software receiver-based tracking.



IV-61 Comparing software receiver - equivalent base band - No Noise

In the next figure, equivalent base band tracking is compared to software receiver-based tracking, considering a real noise scenario.

Based on a link budget model, the signal to noise ratio at antenna interface is calculated along trajectory. For the software receiver, real noise reception signals are generated by the space segment simulator, based on the calculated link budget model. Within the equivalent base band simulation, noise is stimulated, based on the link budget model and also based on the derived signal processing model for all components within the GPS receiver.



IV-62 Comparing software receiver - equivalent base band

As the simulation shows, the equivalent base band simulation and the software receiver tracking show similar noise characteristics.

V Reception Resilience

V - 1 Motivation

GPS - and all other GNSS systems - are typically well functioning, but at the same time, they are very vulnerable systems. The emitted power of a typical GPS satellite is round about 50W at an altitude of 22 000 km. The received signal power at a GPS receiver is $1 \cdot e^{-15}W$ and therefore below the thermal noise power generated within the GPS antenna. Due to this low signal power, GPS is very vulnerable to electromagnetic distortions like jamming.

Especially with the always increasing number of applications, using GPS navigation for safety critical applications like auto landing in aircrafts, autonomous cars, VTOL or multicopter based air taxis or even train protection systems, it is necessary to understand and analyze the effect of jamming on different tracking architectures as well as the possible antijam measures at the different stages within the reception chain.

There is already some literature on different jammer types and the impact of jamming within the receiver, like [87], [88], [89], [87], [90], [91], [92], [93], [94] and [95]. In [96] and [97] different methods for improving antijam are stated at a very general level. Nevertheless, some essential questions are not discussed in literature.

- What is the exact influence of reducing the IF bandwidth on different jammer types
- For simulation purpose, tuning and loss of lock forecast, the discriminator error variance must be calculated in advance. For that, the dependence of the discriminator error variance on different jammer signal types must be derived in detail. For example, a detailed derivation of the antijam, provided by spreading is necessary, given different jammer types or different design variations within the GPS signal processing chain.
- What is more beneficiary, reducing the IF bandwidth or using longer correlation times
- Beam forming is promoted to be the better antijam measure, but being more relevant for civil applications, is Nulling. In this section the question will be answered, under which conditions, Nulling can provide similar antijam as beam forming.

In order to answer this question, representative for all others, two jammer types are analyzed in detail at different stages within the reception chain. For that, the reception chain will be divided into different stages, whereby at each stage, different antijam measures are discussed and the remaining jamming power at the corresponding stage is derived.

Moreover, an interesting effect of PRN code jammers will be analyzed, exploring the dependence of jammer severity on the number of equal PRN code sequences.

Further main topics are a “design guide” for the analog reception chain, in order to get maximum antijam.

V - 2 Discriminator error variance in jammed environments

V - 2.1 Introduction

The correlation core, consisting out of the I/Q generation (carrier wipe off), the despreading, the correlation and the subsequent discriminator, is for calculating respectively measuring the misalignment between the received code phase and the replicated code phase (and carrier phase). This measured misalignment is the used by the tracking filter (which can be a scalar or vector, aided or unaided tracking filter) as measurement. The tracking filter generates the appropriate correction for the replicated code phase. Moreover, the tracking filter calculates the pseudorange and position together with the raw data and position error covariance information.

The measured misalignment at the output of the discriminator is a noisy measurement. In literature like [98], standard relations are given for error variance of this discriminator measurement, depending on the signal to noise ratio at IF interface, the early late spacing, the discriminator type itself and depending on the used correlation time.

For tracking filter tuning and also for simulation of tracking, especially with the purpose to determine the maximum possible jamming power before loss of lock occurs, a precise calculation of this discriminator error variance is vital.

This section will show, that using the standard formulas, in case of different jamming types, the derived discriminator error variance is estimated with too high values. Therefore, in this section, a modified discriminator error variance calculation is derived, providing more accurate results. This modified discriminator error variance calculation uses an effective post correlation signal to noise ratio, being derived in a similar way in [95].

The standard formulas are using the signal to noise ratio at IF frequency, this will also be assumed to be available in this modified approach.

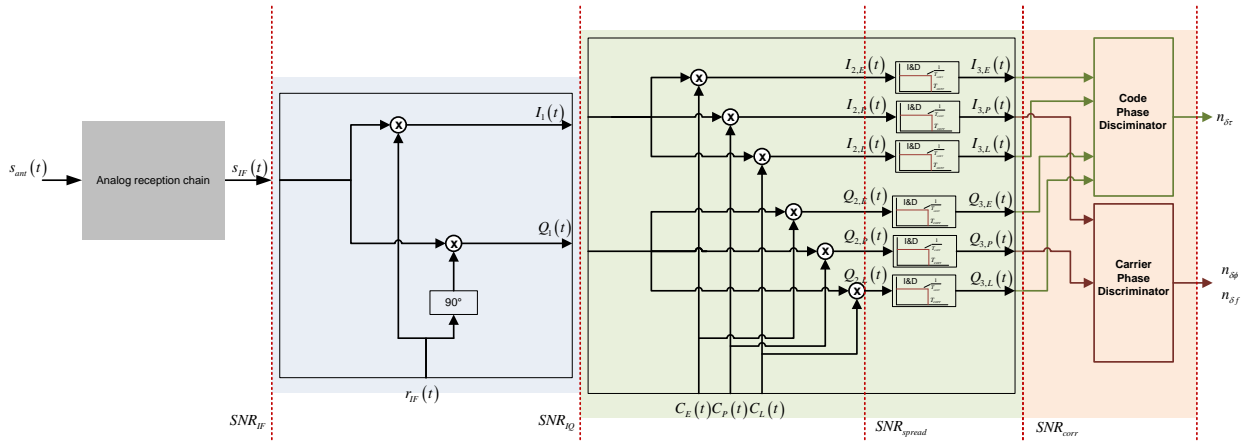
Estimating online the actual signal to noise ratio at the IF frequency, also in case of jamming, is an already well discussed problem in literature as for example in [95], [99], [100] and [101].

V - 2.2 Considered receiver stages

In this section, the remaining jamming power after correlation will be derived. The post correlation jamming power is a vital figure, determining the discriminator error variance and therefore also the pseudorange error variance.

For the holistic approach of pre mission equivalent base band simulation and tracking behavior prediction, this section develops a method, providing an approximation of the post correlation jamming power.

For calculating the post correlation jamming power, it is important to know the impact of different stages within the signal processing chain on the jamming signal and spectrum. The following figure shows these stages.



V-1 Correlation core - stages

V - 2.3 Jammer power at I and Q

In the first stage, the received GPS signal is down converted from an IF frequency to the base band, which is also called “carrier wipe off”. Additionally, the received signal is split into its I and Q parts.

The received signal at IF interface can be represented by the following equation.

$$s_{IF}(t) = A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \sin(\varphi_{IF}(t)) + J(t) + n(t)$$

A_{IF} : signal amplitude

D : navigation message

C : C/A code

J : jammer

χ : received code phase of CA-Code

φ_{IF} : received carrier phase

(V-1)

For down conversion, a replicated carrier, synchronized in frequency and phase, is multiplied with the received IF signal.

$$I(t) = s_{IF}(t) \cdot \sin(\hat{\varphi}_{IF}(t))$$

$$Q(t) = s_{IF}(t) \cdot \cos(\hat{\varphi}_{IF}(t))$$

(V-2)

$$I(t) = A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \frac{1}{2} \cdot (\cos(\varphi_{IF}(t) - \hat{\varphi}_{IF}(t)) + \cos(\varphi_{IF}(t) + \hat{\varphi}_{IF}(t))) + (J_{IF}(t) + n_{IF}(t)) \cdot \sin(\hat{\varphi}_{IF}(t)) \quad (V-3)$$

$$Q(t) = A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \frac{1}{2} \cdot (\sin(\varphi_{IF}(t) - \hat{\varphi}_{IF}(t)) + \sin(\varphi_{IF}(t) + \hat{\varphi}_{IF}(t))) + (J_{IF}(t) + n_{IF}(t)) \cdot \cos(\hat{\varphi}_{IF}(t)) \quad (V-4)$$

The terms $\varphi_{IF}(t) + \hat{\varphi}_{IF}(t)$ represent the high frequency conversion parts, which are filtered out by the subsequent correlation, acting as a low pass filter with an edge frequency of $\frac{1}{T_{corr}}$.

$$\begin{aligned} I(t) &= A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \frac{1}{2} \cdot \cos(\delta\varphi(t)) + (J_{IF}(t) + n_{IF}(t)) \cdot \sin(\hat{\varphi}_{IF}(t)) \\ Q(t) &= A_{IF}(t) \cdot D(t) \cdot C(\chi(t)) \cdot \frac{1}{2} \cdot \sin(\delta\varphi(t)) + (J_{IF}(t) + n_{IF}(t)) \cdot \cos(\hat{\varphi}_{IF}(t)) \end{aligned} \quad (V-5)$$

For post correlation power approximation, the spectral view is important.

The jammer spectra in frequency space at I and Q are given below.

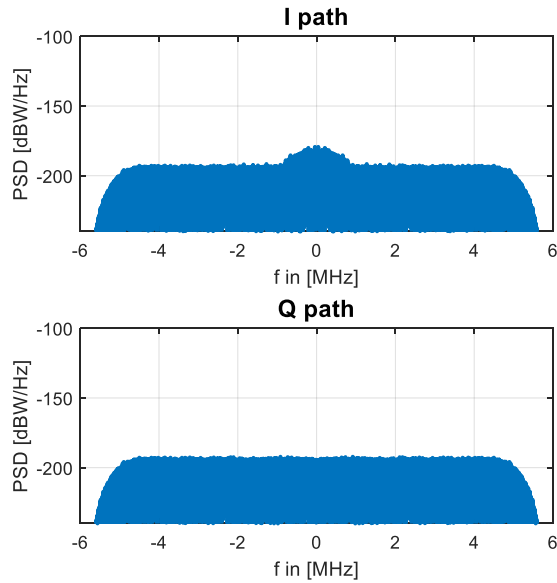
$$\begin{aligned} I_J(t) &= J_{IF}(t) \cdot \sin(\hat{\varphi}_{IF}(t)) \\ Q_J(t) &= J_{IF}(t) \cdot \cos(\hat{\varphi}_{IF}(t)) \\ &\rightarrow \text{Fourier Transformation} \rightarrow \\ \mathcal{I}_J(f) &= \mathcal{J}_{IF}(f) * \left(\frac{1}{j2} \cdot \delta(f + \hat{f}_{IF}) - \frac{1}{j2} \cdot \delta(f - \hat{f}_{IF}) \right) \\ \mathcal{Q}_J(f) &= \mathcal{J}_{IF}(f) * \left(\frac{1}{2} \cdot \delta(f + \hat{f}_{IF}) + \frac{1}{2} \cdot \delta(f - \hat{f}_{IF}) \right) \end{aligned} \quad (V-6)$$

The corresponding power spectral densities get

$$\begin{aligned} \phi_{I,I} &= |\mathcal{I}_J(f)|^2 \\ \phi_{Q,Q} &= |\mathcal{Q}_J(f)|^2 \end{aligned} \quad (V-7)$$

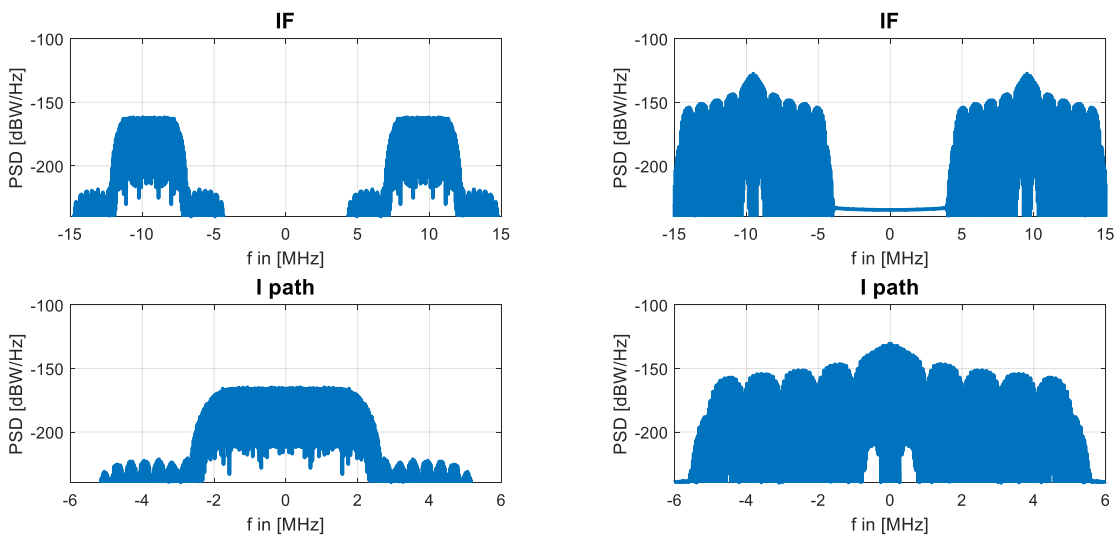
The spectra of different jammers at defined stages are shown at the end of the next section.

In case of perfect synchronization, $\delta\varphi(t) = 0$, only the in-phase component includes the satellite signal. The following figure show's this fact by considering an unjammed normal noise scenario. The example uses an IF bandwidth of 10 MHz.



V-2 IQ spectrum - unjammed scenario

In this thesis, two different jamming types will be considered. The following figures show these jammers at IF band, before IF bandpass filtering and in the I/Q channel. The jamming, as well as the signal and noise power levels are taken from the link budget model, derived in the previous chapter. The given values are from the high dynamic trajectory at time 50s.



V-3 WGN jammer spectrum at IF and I stage (IF BW = 10MHz)

V-4 PRN jammer spectrum at IF and I stage (IF BW = 10MHz)

V - 2.4 Jammer power after correlation

After carrier down conversion, the I and Q signals are multiplied with a replicated C/A code $C(\hat{\chi}(t))$. This multiplication in time domain equals a convolution in frequency domain. This

operation is in literature often described as spreading, [102]. Convoluting small-bandwidth signals, like small band jammers, with a large-bandwidth C/A code spectrum, spreads the jamming power, originally concentrated in a small band, over a much wider band. The overall signal energy remains constant, whereby the power spectral density is reduced. The following integration, which equals a low pass filter in frequency domain, is now able to cut off much of the jamming power, spread beyond the low pass filter bandwidth. For simplicity, the time dependency is dropped in the following equations.

$$\begin{aligned}
I_{2,E} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi} + \Delta\chi) \cdot \cos(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi} + \Delta\chi) \\
I_{2,P} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi}) \cdot \cos(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi}) \\
I_{2,L} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi} - \Delta\chi) \cdot \cos(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi} - \Delta\chi) \\
Q_{2,E} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi} + \Delta\chi) \cdot \sin(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \cos(\hat{\varphi}_{IF}) \cdot C(\hat{\chi} + \Delta\chi) \\
Q_{2,P} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi}) \cdot \sin(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \cos(\hat{\varphi}_{IF}) \cdot C(\hat{\chi}) \\
Q_{2,L} &= \frac{1}{2} \cdot A_{IF} \cdot D \cdot C(\chi) \cdot C(\hat{\chi} - \Delta\chi) \cdot \sin(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \cos(\hat{\varphi}_{IF}) \cdot C(\hat{\chi} - \Delta\chi)
\end{aligned} \tag{V-8}$$

Because the noise power after correlation is important for the following discriminator error variance prediction, in the following equations, the noise power will be derived. Representative of all I and Q signals, the correlation output is derived only for the prompt in phase signal in the following.

$$\begin{aligned}
I_{3,P} &= \int_{t-T_{corr}}^t \frac{A_{IF}}{2} \cdot D \cdot C(\chi) \cdot C(\hat{\chi}) \cdot \cos(\delta\varphi) + (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi}) dt \\
&= \frac{A_{IF}}{2} \cdot \int_{t-T_{corr}}^t D \cdot C(\chi) \cdot C(\hat{\chi}) \cdot \cos(\delta\varphi) dt + \int_{t-T_{corr}}^t (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi}) dt
\end{aligned} \tag{V-9}$$

Assuming a perfectly synchronized code tracking loop with $\hat{\chi} = \chi$ and a perfectly synchronized carrier tracking loop with $\delta\varphi = 0$, the correlation output gets

$$I_{3,P} = \frac{A_{IF}}{2} \cdot T_{corr} + \underbrace{\int_{t-T_{corr}}^t (J_{IF} + n_{IF}) \cdot \sin(\hat{\varphi}_{IF}) \cdot C(\hat{\chi}) dt}_{I_{3,P,n} + I_{3,P,J}} \tag{V-10}$$

It is further assumed, that the navigation message is known and cancelled before correlation.

The noise power after correlation is calculated, using the variance.

$$P_{3,P,J+n} = \mathcal{E}\{I_{3,P,J+n}^2\} = \mathcal{E}\left\{ \int_{t-T_{corr}}^t (J_{IF} + n_{IF}) \cdot \sin(\hat{\phi}_{IF}) \cdot C(\hat{\chi}) dt_1 \cdot \int_{t-T_{corr}}^t (J_{IF} + n_{IF}) \cdot \sin(\hat{\phi}_{IF}) \cdot C(\hat{\chi}) dt_2 \right\} \quad (V-11)$$

The equation above can be split into a noise power part and a jamming power part, assuming that there is no correlation between jammer and noise. At first, the noise power part will be considered.

$$P_{3,P,n} = \mathcal{E}\left\{ \int_{t-T_{corr}}^t \int_{t-T_{corr}}^t n_{IF}(t_1) \cdot n_{IF}(t_2) \cdot \delta(t_1 - t_2) \cdot \sin(\hat{\phi}_{IF}(t_1)) \cdot \sin(\hat{\phi}_{IF}(t_2)) \cdot C(\hat{\chi}(t_1)) \cdot C(\hat{\chi}(t_2)) dt_1 dt_2 \right\} \quad (V-12)$$

Because n_{IF} being white gaussian noise, it is uncorrelated if not $t_1 - t_2 = 0$. Therefore, the noise power after correlation gets

$$\begin{aligned} P_{3,P,n} &= \mathcal{E}\left\{ \frac{1}{T_{corr}} \cdot \int_{t-T_{corr}}^t n_{IF}^2 \cdot \sin^2(\hat{\phi}_{IF}) \cdot \underbrace{C^2(\hat{\chi})}_1 dt \right\} = \frac{1}{T_{corr}} \cdot \int_{t-T_{corr}}^t \mathcal{E}\{n_{IF}^2\} \cdot \mathcal{E}\{\sin^2(\hat{\phi}_{IF})\} dt \\ &= \frac{1}{T_{corr}} \cdot \int_{t-T_{corr}}^t \frac{N_{0,IF}}{2 \cdot T_{corr}} \cdot \frac{1}{2} dt = \frac{N_{0,IF}}{4 \cdot T_{corr}} \cdot \frac{1}{T_{corr}} \cdot \int_{t-T_{corr}}^t 1 dt = \frac{N_{0,IF}}{4 \cdot T_{corr}} \cdot \frac{1}{T_{corr}} \cdot T_{corr} = \frac{N_{0,IF}}{4 \cdot T_{corr}} \end{aligned} \quad (V-13)$$

The approximation $\mathcal{E}\{\sin^2(\hat{\phi}_{IF})\} \approx \frac{1}{2}$ is valid, because the IF carrier frequency is much higher compared to the correlation time.

The remaining jamming power after correlation can be calculated like the noise power approach derived before.

$$P_{3,P,J} = \mathcal{E}\left\{ \int_{t-T_{corr}}^t \int_{t-T_{corr}}^t J_{IF}(t_1) \cdot J_{IF}(t_2) \cdot \sin(\hat{\phi}_{IF}(t_1)) \cdot \sin(\hat{\phi}_{IF}(t_2)) \cdot C(\hat{\chi}(t_1)) \cdot C(\hat{\chi}(t_2)) dt_1 dt_2 \right\} \quad (V-14)$$

If the jamming signal can be approximated as white gaussian noise after spreading, then the post correlation noise power caused by jamming, can be written as

$$P_{3,P,J} = \frac{N_{J,IF}}{4 \cdot T_{corr}} \quad (V-15)$$

If the white noise approach is not valid, there is in most cases no analytical solution.

In such scenarios, a different approach can help to get a rough estimation of the jammer induced noise power after correlation.

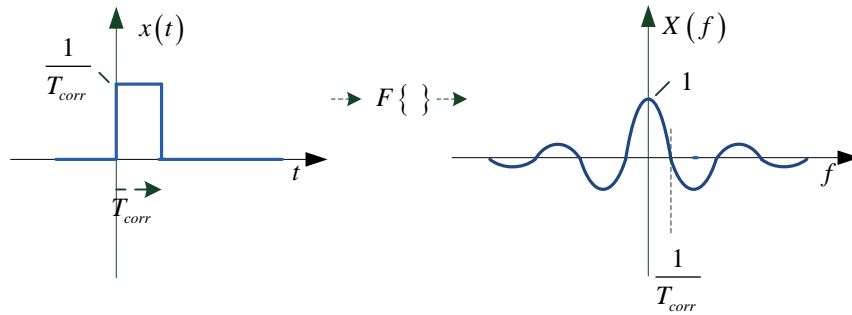
At first, the jammer spectrum after spreading is considered. For that, it is assumed that the navigation message is known and cancelled before spreading and the carrier tracking loop is locked.

$$\mathcal{I}_{2,P,J}(f) = \mathcal{J}_{IF}(f) * \left(\frac{1}{j2} \cdot \delta(f + \hat{f}_{IF}) - \frac{1}{j2} \cdot \delta(f - \hat{f}_{IF}) \right) * \hat{\mathcal{C}}(f) \quad (\text{V-16})$$

The subsequent integration can be considered as the convolution with a rectangular function.

$$I_{3,P,J}(t) = \frac{1}{T_{corr}} \cdot \int_{t-T_{corr}}^t I_{2,P,J}(t) dt = I_{2,P,J}(t) * \frac{1}{T_{corr}} \cdot \text{rect}\left(\frac{t}{T_{corr}}\right) \quad (\text{V-17})$$

In frequency space, the convolution with a rectangular function of duration T_{corr} , equals the multiplication with a si function. The si function shapes the noise spectrum in frequency space.



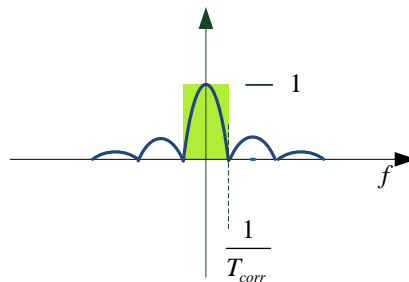
V-5 Fourier transformation of rectangular function

$$\mathcal{I}_{3,P,J}(f) = \mathcal{I}_{2,P,J}(f) \cdot \text{si}\left(\frac{2\pi}{T_{corr}}\right) \quad (\text{V-18})$$

The noise power after correlation gets

$$\begin{aligned} P_{3,P,J} &= \int_{-\infty}^{\infty} \mathcal{I}_{2,P,J}^2(f) \cdot \text{si}^2\left(\frac{2\pi}{T_{corr}}\right) df = \int_{-\infty}^{\infty} \phi_{2,P,J} \cdot \text{si}^2\left(\frac{2\pi}{T_{corr}}\right) df \\ &= \int_{-\infty}^{\infty} \left[\mathcal{J}_{IF}(f) * \left(\frac{1}{j2} \cdot \delta(f + \hat{f}_{IF}) - \frac{1}{j2} \cdot \delta(f - \hat{f}_{IF}) \right) * \hat{\mathcal{C}}(f) \right]^2 \cdot \text{si}^2\left(\frac{2\pi}{T_{corr}}\right) df \end{aligned} \quad (\text{V-19})$$

The squared si function is approximated by a rectangular function in frequency space, as the following figure shows.

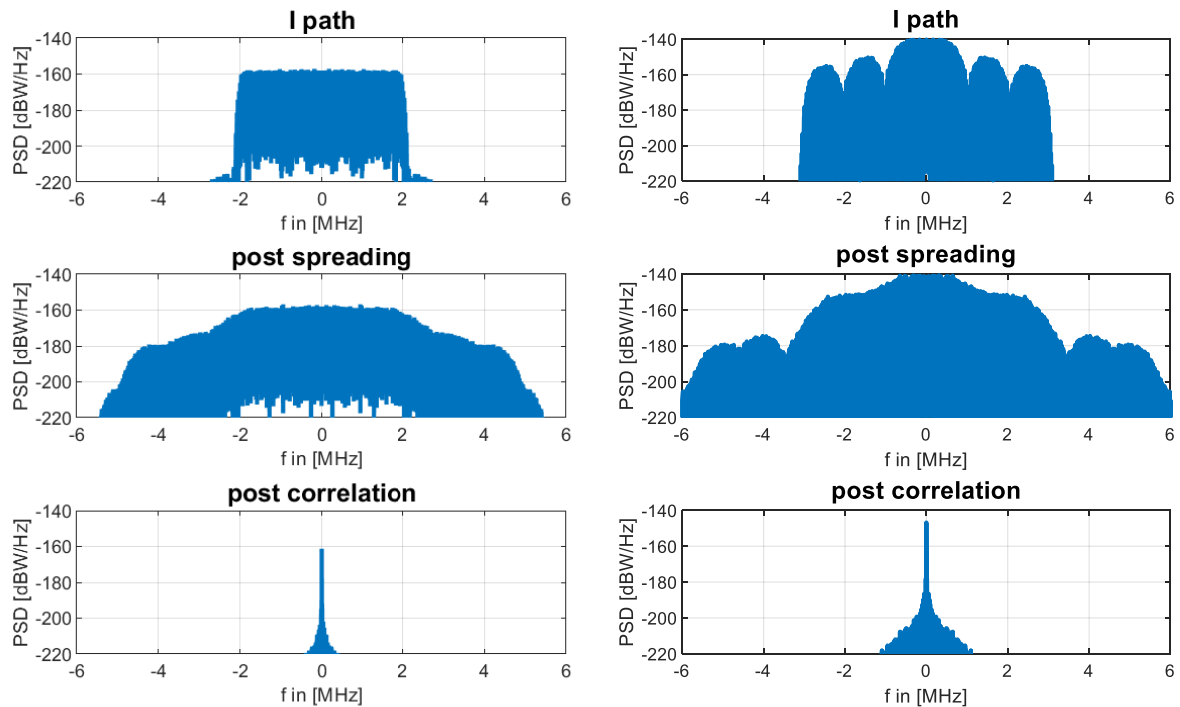


V-6 rectangular approximation of sinc function

If the power spectral density $\phi_{2,P,J} = |\mathcal{I}_{2,P,J}(f)|^2$ of the jammer can be approximated as constant within the small bandwidth of $\left(-\frac{1}{T_{corr}}, \frac{1}{T_{corr}}\right)$, the post correlation jamming power can be approximated as

$$P_{3,P,J} = \frac{\phi_{2,P,J}}{T_{corr}} \quad (V-20)$$

The following figures show the jammer spectra for two different jammer types at the different stages within the GPS receiver. As IF bandwidth, 6 MHz is used. For the WGN jammer, a one side jamming bandwidth of 2 MHz is used.



V-7 WGN jammer spectrum (IF BW = 10MHz)

V-8 PRN jammer spectrum (IF BW = 10MHz)

The figures above show the spreading effect. The jammer power within the I channel is spread about a greater bandwidth, by multiplication with the replicated C/A code. What matters is only the remaining jammer power within the small bandwidth, due to the subsequent correlation. The correlation can be approximated in frequency space as a low pass filter, having a single side bandwidth of $\frac{1}{T_{corr}}$.

The two examples above show already, that spreading works better in case of smaller spectra. The jammer power spectral density of the white noise jammer after correlation is almost equal, whereby the jammer power spectral density of the PRN jammer after correlation is reduced.

An evaluation of the influence of the IF bandwidth on post correlation jamming power for two

different jammer types will be done in section V - 3 and V - 4

The following tables compare at the example of a white gaussian noise jammer, the real measured post correlation jamming power with the approximation, introduced above.

IF BW = 10 MHz	WGN jammer
real power	-147.4373 dBW
Approximated power	-147.6784 dBW

Table 1 Real vs. approximated post correlation jamming power (IF BW 10MHz)

IF BW = 1 MHz	WGN jammer
real power	-144.8505 dBW
Approximated power	-145.0638 dBW

Table 2 Real vs. approximated post correlation jamming power (IF BW 1MHz)

The values above show, that the jamming power approximation is valid in case of white noise jamming signals.

In case of non-white jammers, which is the case if the IF bandwidth is very low, the approximation becomes less accurate.

V - 2.5 Discriminator error variance with spreading effect consideration

The discriminator output is a measurement of the phase deviation between the replicated code or carrier phase and the received code or carrier phase. Its error variance has a high impact on the pseudorange error variance and causes a loss of lock in case of high values. There are many types of different discriminator. In the following, two mostly used discriminator types are analyzed.

In literature [103], [81, p. 502], the following expressions for the discriminator error variances are given. In [104], an analytical derivation of the mean time to loss of lock, based on discriminator error variance is calculated. But the given approach is only valid for a distinct tracking architecture and distinct jammer type.

Early-Late code phase discriminator

$$\sigma_{n,\delta\tau}^2 = \frac{d \cdot T_{CAchip}^2}{4 \cdot T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}} \right)} \quad (V-21)$$

Early-Late-Power code phase discriminator

$$\sigma_{n,\delta\tau}^2 = \frac{T_{CAchip}^2}{4 \cdot T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}}\right)} \cdot \left(1 + \frac{2}{T_{corr} \cdot \left(\frac{C_{IF}}{N_{IF}}\right)} \right), \quad d = 1 \quad (V-22)$$

As can be seen, there are different expressions, depending on the discriminator type.

The Early-Late discriminator is only possible in case of a locked carrier tracking loop. Otherwise, the Early-Late-Power discriminator must be used, having a higher discriminator error variance.

The two expressions above use the signal to noise power at IF interface, to calculate the discriminator error variance.

But this assumption is valid only for white antenna noise or white jamming signals, together with a large IF bandwidth. In case of non-white jammers or a small IF bandwidth, the calculated discriminator error variances give to high values.

The noise power spectral density which matters for the discriminator error variance is the residual noise power spectral density at the output of the correlator.

The used noise power spectral density at IF stage, before I/Q generation, N_{IF} , is higher than the noise power spectral density after correlation, because spreading is not considered. Especially in case of a small IF bandwidth or low-bandwidth jamming signals, spreading decreases the post correlation noise power spectral density significantly.

As was shown in in section V - 3 the noise power spectral density after correlation, respectively jamming power density after correlation $|\mathcal{J}_{3,E}(f)|^2$, given some white noise jammer in combination with small IF bandwidth, is lower than the corresponding noise power spectral density at IF interface. Especially within the small bandwidth region $f < \frac{1}{T_{corr}}$, being relevant for discriminator error variance.

This effect, which is caused by spreading, is not considered in expressions (V-21) and (V-22) on discriminator error variance.

Instead using N_{IF} in equations (V-21) or (V-22), the post correlation jamming power spectral density $|\mathcal{J}_{3,E}(f)|^2$ will be used.

In the following, there will be a comparison of the following discriminator error variances respectively standard deviations.

(1) Real discriminator error variance measurement

(2) Analytic discriminator error variance calculation using $\left(\frac{C_{IF}}{N_{IF}}\right)$

(3) Analytic discriminator error variance calculation using $\left(\frac{C_{IF}}{|\mathcal{J}_{3,E}(f)|^2}\right)$

EL discriminator	IF BW 0.5 MHz	IF BW 1 MHz	IF BW 2 MHz
Real Measurement	2.1309 us	2.2024 us	2.1663 us
$\left(\frac{C_{IF}}{ \mathcal{J}_{3,E}(f) ^2}\right)$	2.337 us	2.4255 us	2.4247 us
$\left(\frac{C_{IF}}{N_{IF}}\right)$	4.4339 us	3.6616 us	3.5319 us

Table 3 EL discriminator – comparison of error standard deviation

The table above shows that calculating $\sigma_{n,\delta\tau}$ by using $\left(\frac{C_{IF}}{|\mathcal{J}_{3,E}(f)|^2}\right)$ gives more accurate results, compared to the calculation, which uses $\left(\frac{C_{IF}}{N_{IF}}\right)$. The last one estimates a too high discriminator error variance, because by using N_{IF} , the jammer density peak power reduction by spreading is not considered and N_{IF} is higher compared to $|\mathcal{J}_{3,E}(f)|^2$.

V - 3 Optimal IF bandwidth – given WGN jammer

This section is about the optimal IF bandwidth selection for maximum antijam.

The smaller the IF bandwidth, the less jamming power enters the subsequent signal processing stages.

The previous section on post correlation jamming power shows that only a very small part of the IF jamming spectrum enters the post correlation stage.

$$\mathcal{J}_{3,E}(f) \approx \begin{cases} \mathcal{J}_{2,E}(f) & 0 < f < \frac{1}{T_{corr}} \\ 0 & f > \frac{1}{T_{corr}} \end{cases} \quad (V-23)$$

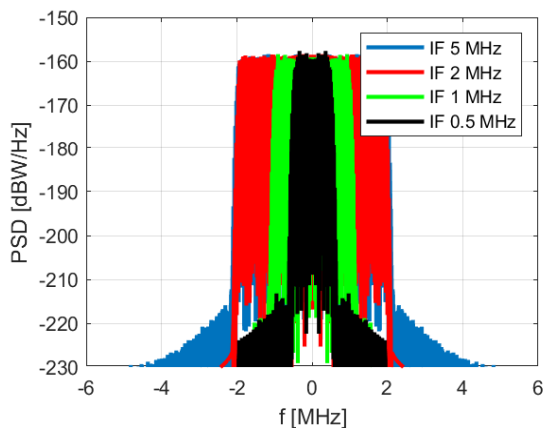
The condition is valid also for prompt and late components in I and Q path.

Without the spreading effect, the noise power spectral density after correlation $|\mathcal{J}_{3,E}(f)|^2$, within the small spectra range $0 < f < \frac{1}{T_{corr}}$, would be equal to the jamming power spectral density after IF filtering, $|\mathcal{J}_{IF}(f)|^2$.

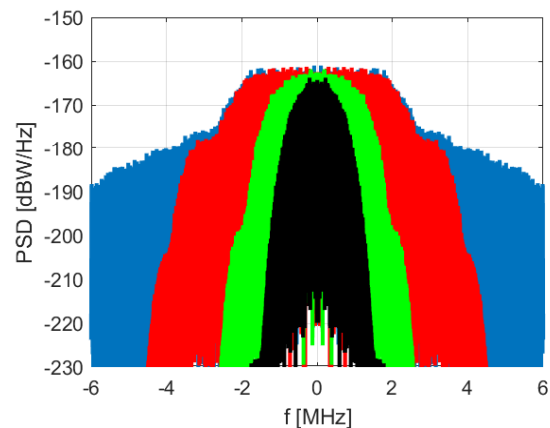
Spreading starts to add some antijam respectively lower the post correlation jammer power spectral density, as soon as the jammer bandwidth before spreading is almost equal or lower as the bandwidth of the replicated C/A code, which has a 1.023 MHz one-side bandwidth of the main lobe.

Given white noise jammers, the IF bandwidth must be smaller than ~ 2 MHz, in order to enable a post correlation jamming power reduction by spreading.

The following figures visualize this effect.



V-9 WGN jamming spectra - after IF filtering

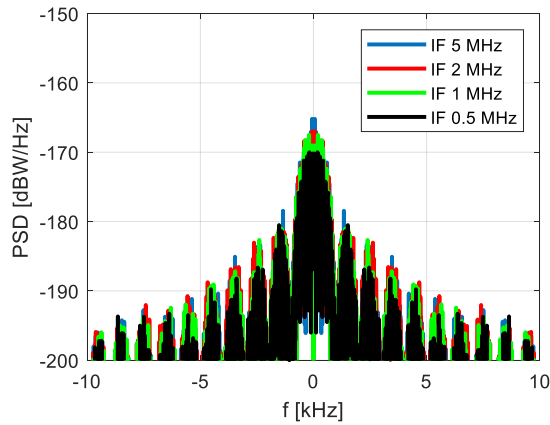


V-10 WGN jamming spectra - after spreading

An IF bandwidth of 0.5 MHz shows the greatest spreading effect.

Relevant for tracking stability and discriminator error variance is only the jamming power within the small frequency range of $0 < f < \frac{1}{T_{corr}}$. The examples above show that an IF bandwidth of 1 MHz and 0.5 MHz reduce the power spectral density within the relevant spectral region.

The following figure shows the post correlation spectra, given different IF bandwidth's and a correlation time of 1ms.

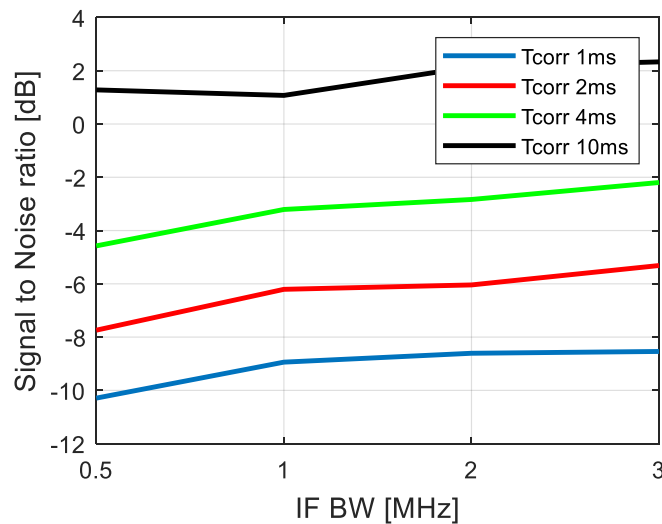


V-11 Post correlation spectra - WGN jammer - different IF BW

The downside of reducing the IF bandwidth is the reduced signal power, which comes with it.

Besides the IF bandwidth, also the correlation time has influence on the post correlation jamming power and post correlation signal power.

For evaluation of the optimal IF bandwidth together with the corresponding optimal correlation time, the following figure shows the post correlation signal to noise ratios for different correlation times and IF bandwidths.



V-12 Comparing C/N post correlation given different IF bandwidths

Given white noise jammer, the simulations show that reducing the IF bandwidth provides no additional antijam. On the contrary, the signal to noise ratio gets even worse, because the correlation gain is lowered to higher extent than the jamming power is reduced.

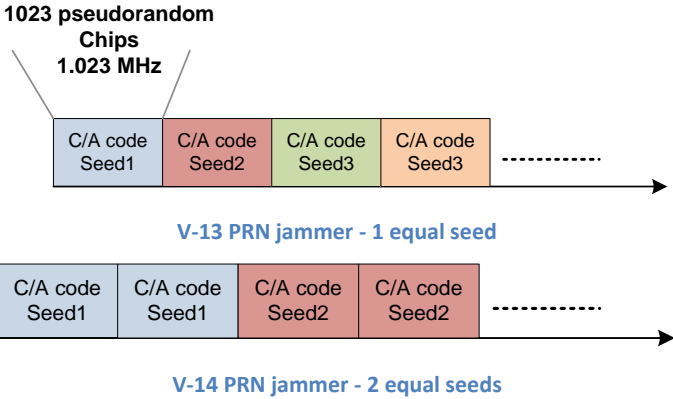
Using longer correlation times provides far more antijam than reducing the IF bandwidth. But, selecting a correlation time of 10 ms without the application of external aiding, the tracking loop is not able to follow high dynamic trajectories.

V - 4 Optimal IF bandwidth – given different PRN jammer realizations

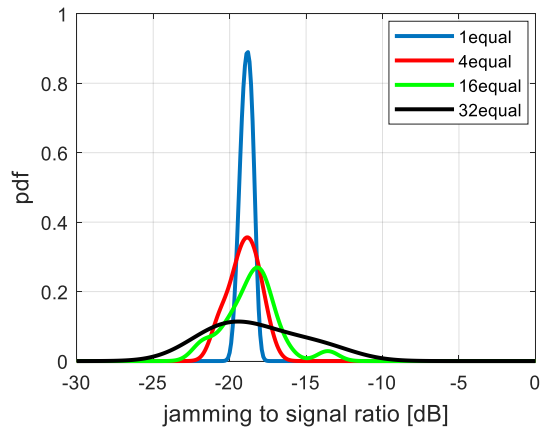
A special jammer type are pseudorandom code jammers. Their signal structure is similar to the real C/A code. The jammer signals, even if they are of the same type as C/A code, are neither orthogonal related to the different C/A signals, nor show any correlation with the real C/A code. Nevertheless, this jammer type worsens the post correlation signal to noise level to a huge extent.

This section evaluates the post correlation, jammed signal to noise ratio, depending on the IF bandwidth and the correlation time for pseudorandom code jammers.

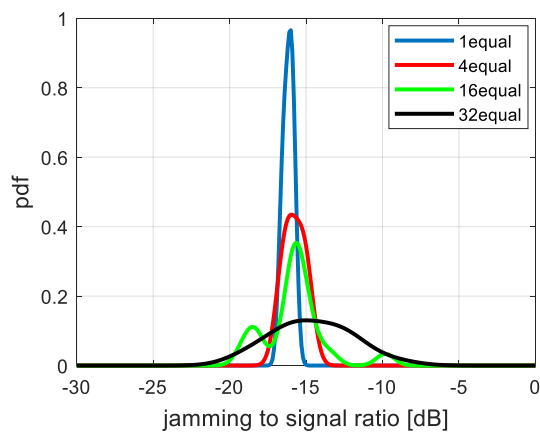
Moreover, an interesting effect of pseudorandom code jammer will be analyzed in detail. The post correlation signal to noise ratio shows a normal distribution, whereby the mean value and its variance depend on the pseudorandom code structure, respectively the number of equal seeds and the chosen IF bandwidth.



The following figures show the distribution of signal to noise ratios, given different IF bandwidths.



V-15 PRN jammer - IF BW 0.5 MHz - Tcorr 1ms



V-16 PRN jammer - IF BW 3 MHz - Tcorr 1ms

PRN jammers, using a new seed after 1023 code chips, having a sharp probability density function. The mean value depends slightly on the selected IF bandwidth. A wider IF bandwidth shifts the jamming to signal ratio about 3 dB to higher levels.

The more C/A code periods the seed is kept equal, the flatter the pdf becomes. This makes the jammer more dangerous, because with a small probability, there are jamming to signal ratios, post spreading, above -10 dB possible.

V - 5 Overall noise figure minimization

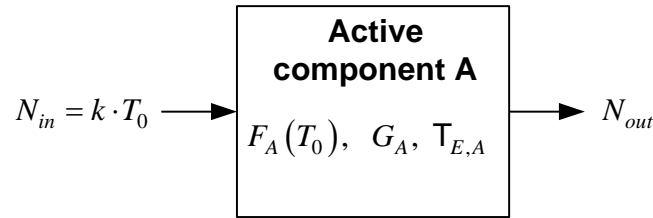
The first signal processing stage within the GPS receiver, after the antenna, is the analog unit. Within the analog unit, the received signal at L1 band is down converted and amplified by several stages, to an intermediate frequency.

The analog unit does not provide antijam by itself, but it is vital to sustain the antijam provided by other components within the signal processing chain. Every dB of additional noise, added by the analog unit, would reduce the overall antijam. Therefore, it is the most important design criteria, keeping the additional noise as small as possible.

In the following, a short introduction to noise figure calculations will be given. By using these equations, design guidelines for the analog unit will be derived.

Basically, four different tools are necessary for calculating the noise figure of cascaded systems, like the analog unit.

Noise figure of an active component



At the input of the active component, the following noise power density is assumed

$$N_{in} = k \cdot T_0 \quad (V-24)$$

The active component amplifies the input noise with the gain G_A and is adding additional noise. The output noise power density gets

$$N_{out} = k \cdot T_0 \cdot G_A + N_A \quad (V-25)$$

The output noise power can also be written, using a specific component temperature.

$$N_{out} = k \cdot G_A \cdot (T_0 + T_{E,A}) \quad (V-26)$$

The active component is classified by three specific parameters. The one parameter is specific temperature $T_{E,A}$ and the gain G_A . The third parameter is the noise figure.

$$F_A(T_0) = \frac{\left(\frac{C_{in}}{N_{in}}\right)}{\left(\frac{C_{out}}{N_{out}}\right)} = \frac{\left(\frac{C_{in}}{k \cdot T_0}\right)}{\left(\frac{C_{in} \cdot G_A}{k \cdot G_A \cdot (T_0 + T_{E,A})}\right)} = 1 + \frac{T_{E,A}}{T_0} \quad (V-27)$$

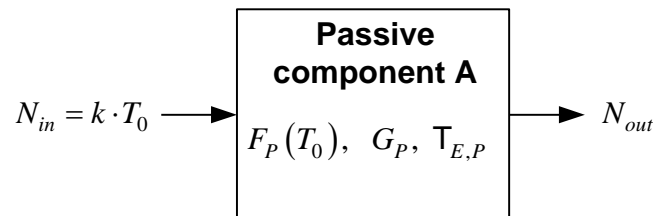
The noise figure of an active component is related to a reference temperature T_0 . If the noise figure is used for calculations in cascaded systems, this reference temperature may be adapted, because all components must have the same reference temperature.

Adapting noise figure to different reference temperatures

The specific temperature of a component is independent of the noise figure reference temperature. This fact is used for changing the noise figures reference temperature.

$$\begin{aligned}
 T_E &= (F(T_0) - 1) \cdot T_0 \\
 T_E &= (F(T_1) - 1) \cdot T_1 \\
 \Rightarrow F(T_1) &= (F(T_0) - 1) \cdot \frac{T_0}{T_1} + 1
 \end{aligned}
 \tag{V-28}$$

Noise figure of a passive component



The following model helps, understanding the noise figure derivation of a passive component.

The input noise power density is generated for example by some copper line, according to its temperature. Assuming, the passive component is also a copper line, connected to the first one and having the same temperature. In this case, there is no additional noise, because both connected lines can be considered as one line

The passive component is adding no additional noise, if its temperature equals the input noise temperature. The gain of the passive component is smaller than one.

The noise figure of the passive component gets

$$F_P(T_0) = \frac{\left(\frac{C_{in}}{N_{in}} \right)}{\left(\frac{C_{out}}{N_{out}} \right)} = \frac{\left(\frac{C_{in}}{k \cdot T_0} \right)}{\left(\frac{C_{in} \cdot G_P}{k \cdot T_0} \right)} = \frac{1}{G_P}
 \tag{V-29}$$

The input noise to a passive component is not attenuated, because the component generates exactly the same noise power density as given at the input. A passive component only attenuates the received signal power.

Using the following logical experiment, the result above gets obvious.

Assuming a one-meter piece of copper wire, this wire could be divided into an infinity number of sections. Every section could now be considered as an own passive component. If every

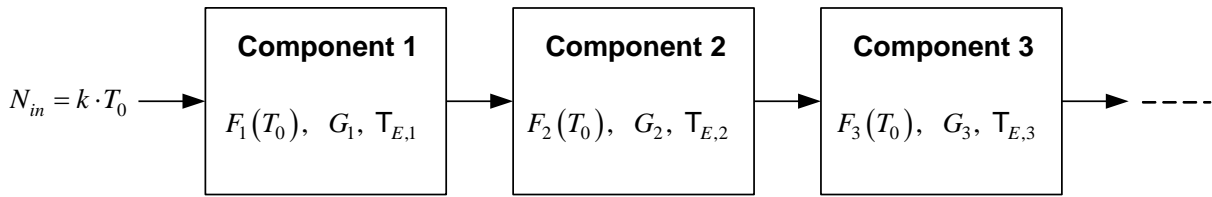
component would add the same amount of noise power density, at the end, an infinite amount of noise power would be available.

Also, for passive components, an equivalent noise temperature can be calculated.

$$T_{E,P} = (F(T_0) - 1) \cdot T_0 = \left(\frac{1 - G_P}{G_P} \right) \cdot T_0 \quad (V-30)$$

Noise figure of cascaded systems

Up to now, only noise figures of single components were considered. For calculating the noise figure of a cascaded system, consisting out of more components like the analog unit of the GPS receiver, the “Friis Formula” can be used. A detailed derivation of this equation is given in [105].



V-17 General example of a cascaded system

The output noise power density can be written, using on overall effective temperature $T_{E,R}$.

$$N_{out} = k \cdot G_1 \cdot G_2 \cdot G_3 \cdot (T_0 + T_{E,R})$$

$$T_{E,R} = T_{E,1} + \frac{T_{E,2}}{G_1} + \frac{T_{E,2}}{G_1 \cdot G_2} \quad (V-31)$$

The cascaded noise figure gets

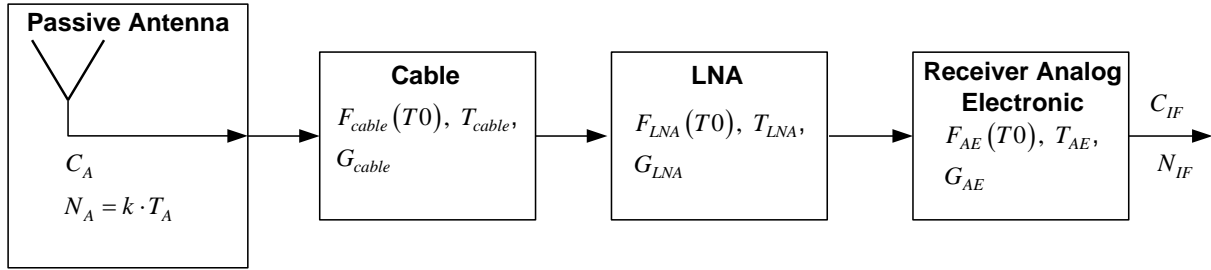
$$F_R(T_0) = F_1(T_0) + \frac{F_2(T_0) - 1}{G_1} + \frac{F_3(T_0) - 1}{G_1 \cdot G_2} \quad (V-32)$$

A detailed explanation of noise figure calculations can be found in [105] and [106].

Using the relations introduced so far, the cascaded noise figure of a typical analog unit can be calculated.

Overall noise figure of using a passive antenna

The following figure shows the aggregation of the analog reception chain, using a passive antenna.



V-18 Analog receiver reception chain - passive antenna

The reference temperature for the components noise figure is T_0 , which is different from the antenna temperature, being the first noise generator.

For calculating the cascaded noise figure, at first all component noise figures must be recalculated to the noise temperature of the antenna, using equation (V-28).

$$\begin{aligned}
 F_{cable}(T_A) &= \frac{T_0}{T_A} \cdot (F_{cable}(T_0) - 1) + 1 \\
 F_{LNA}(T_A) &= \frac{T_0}{T_A} \cdot (F_{LNA}(T_0) - 1) + 1 \\
 F_{AE}(T_A) &= \frac{T_0}{T_A} \cdot (F_{AE}(T_0) - 1) + 1
 \end{aligned}
 \tag{V-33}$$

If for each component the effective noise temperature is available, equation (V-31) should be used for calculating the overall effective noise temperature. The effective noise temperature is independent of the reference temperature. In the following, it is assumed that only the noise figures are available, that's why the classical Friis formula is used. Using this formula, all noise figures must be transformed to a similar reference temperature.

The cascaded noise figure of the reception chain above gets

$$F_{HF}(T_A) = F_{cable}(T_A) + \frac{F_{LNA}(T_A) - 1}{G_{cable}} + \frac{F_{AE}(T_A) - 1}{G_{LNA} \cdot G_{cable}}
 \tag{V-34}$$

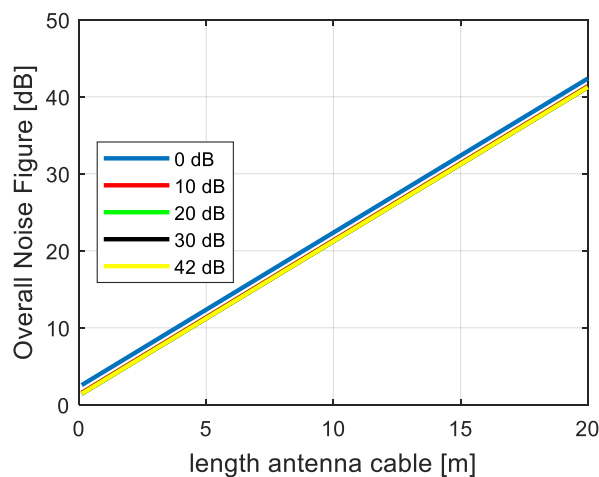
The following table gives the parameters of the components

Reference Temperature	$T_0 = 290K$
Cable	$G_{cable} = -2 \frac{dB}{m} \cdot l$ $F_{cable} = \frac{1}{G_{cable}}$
LNA	$G_{cable} = 29dB$ $F_{cable}(T_0) = 2dB$
Analog Electronic	$G_{cable} = 0dB$ $F_{cable}(T_0) = 3.2dB$

Antenna	$N_A = -200 \frac{dBW}{Hz} = 10^{\left(\frac{-200}{10}\right)} \frac{W}{Hz} = k \cdot T_A = 1.38064852 \cdot 10^{-23} \cdot T_A$ $T_A = 724K$
---------	---

Table 4 Parameters reception chain

As the cascaded noise figure shows, the longer the cable between antenna and LNA, the more impact has the noise of the LNA. If a passive antenna is used, the cabling between antenna and LNA must be as short as possible. The following figure shows the relation between cable length and overall noise figure, using different LNA gains.

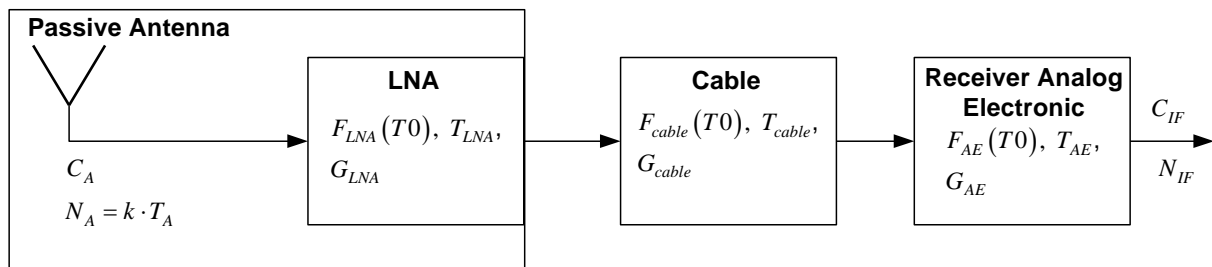


V-19 noise figure dependence on antenna cable length

The analysis shows that the cable length has tremendous influence on the overall noise figure. Whereby the LNA gain has almost no influence on the overall noise figure.

Overall noise figure of using an active antenna

For a minimum noise figure, amplification within the antenna itself is the best solution, as will be shown in the following section. This architecture is realized using a so-called active antenna. The following figure shows the corresponding reception chain.

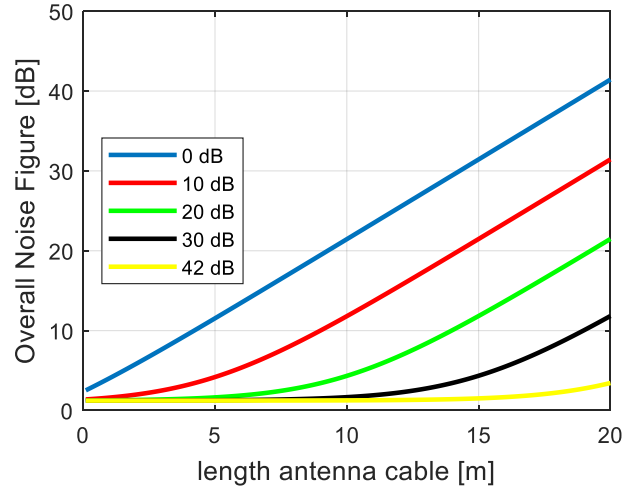


V-20 reception chain - active antenna

The overall noise figure gets

$$F_{HF}(T_A) = F_{LNA}(T_A) + \frac{F_{cable}(T_A) - 1}{G_{LNA}} + \frac{F_{AE}(T_A) - 1}{G_{LNA} \cdot G_{cable}} \quad (V-35)$$

The following figure shows the overall noise figure, depending on cable length and LNA gain.



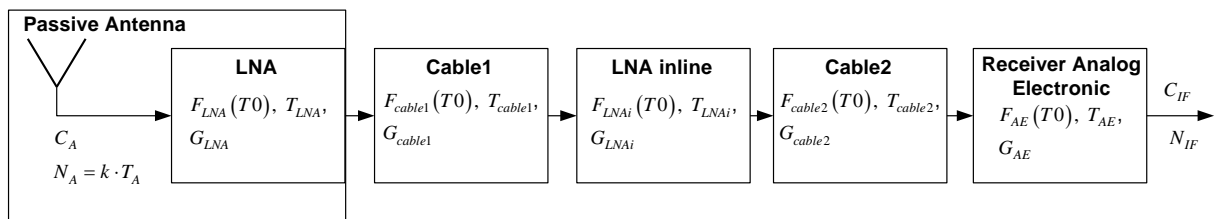
The figure shows the advantage of an active antenna. The influence of the antenna cable length on the overall noise figure becomes less, the higher the gain within the active antenna.

For optimal tracking robustness it's therefore a mandatory architecture recommendation to use an active antenna.

It is desirable, choosing an antenna integrated LNA with high gain. But in the overall design, also the maximum allowed signal amplification at the input of the receiver analog unit must be considered. Otherwise there will be a signal override, which causes even more noise.

Noise figure of an active antenna with additional inline amplifier

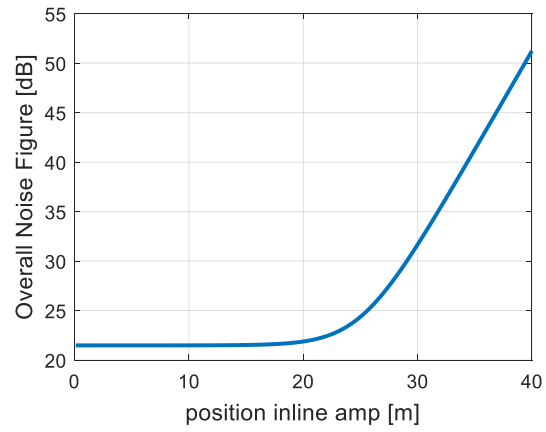
If it is necessary to position the GPS antenna far from the receiver, even in case of active antennas, an additional inline amplifier may be necessary. The following equation gives the overall noise figure, considering also this inline amplifier.



V-21 reception chain - active antenna - inline amplifier

$$F_{HF}(T_A) = F_{LNA}(T_A) + \frac{F_{cable1}(T_A) - 1}{G_{LNA}} + \frac{F_{LNAi}(T_A) - 1}{G_{LNA} \cdot G_{cable1}} + \frac{F_{cable2}(T_A) - 1}{G_{LNA} \cdot G_{cable1} \cdot G_{LNAi}} + \frac{F_{AE}(T_A) - 1}{G_{LNA} \cdot G_{cable1} \cdot G_{LNAi} \cdot G_{cable2}} \quad (V-36)$$

The following figure considers an example design, with a total cable length of 40m. The figure shows the optimal position of the inline amplifier. As gain, 30dB for each LNA's is used.



The optimal position of the inline amplifier is within the first third of the total cable length. The inline amplifier should not be positioned directly after the LNA in the active antenna, in order to prevent a signal override in the inline amplifier.

V - 6 Jammer mitigation by beam forming and nulling

Reception pattern control is one of the most effective measures to increase antijam of a GNSS receiver.

There are already many published approaches for reception pattern control. A large collection of algorithms can be found in [107] and [108]. A detailed mathematical introduction to phased arrays is given in [109], [110] and [111].

Nevertheless, there are some vital topics regarding reception pattern forming, which are not, or only superficially discussed in literature. This chapter discusses exactly this very important topics.

After a short discussion of signal processing principles of reception pattern forming, this section introduces one promising beam forming approach.

Beamforming needs a special GPS receiver with special interfaces and a special internal signal processing architecture.

Another approach is nulling, which can be realized within a CRPA and thus can be used in combination with a standard receiver. The literature provides only few publications regarding nulling. This section introduces some nulling algorithms and suggests also a novel nulling approach.

One goal of this section is to identify the possible antijam which can be achieved by nulling in contrast to beam forming.

Additionally, this section identifies the effects, having major impact on beam forming and nulling performance.

A further important design criterion is the array geometry, which is also discussed very little in literature. In this section, different array geometries will be introduced and their impact on antijam in case of beam forming and nulling is analyzed.

Most discussions regarding beam forming and nulling in literature assume a uniform spherical reception pattern for each single patch. This section evaluates the effect of a non-uniform spherical patch reception pattern, being more realistic.

The theoretical antijam of different nulling algorithms and beam forming algorithms is very high. This theoretical high antijam is achieved by assuming perfect array geometries without any errors and also no attitude errors regarding the jammer and beam forming direction.

Array geometry errors and attitude errors worsen the resulting antijam of both, beam forming and also nulling, tremendously. This section evaluates the resulting degradation and evaluates also the differences in case of such errors between nulling and beam forming.

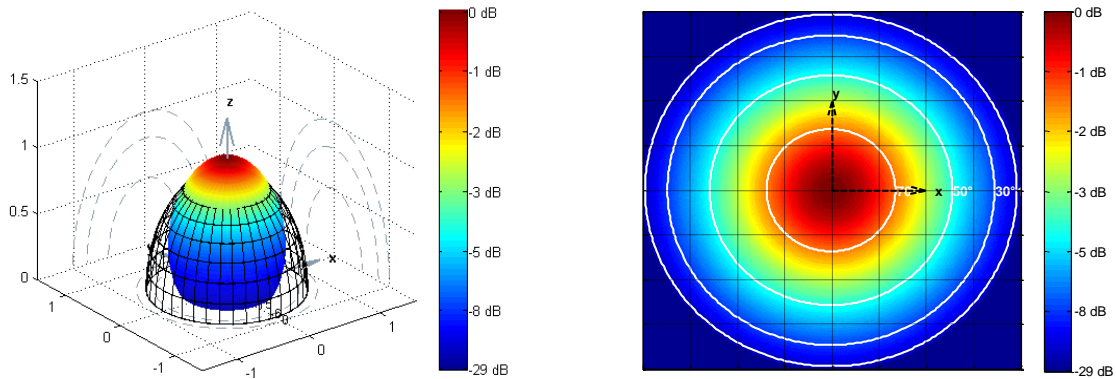
Finally, both approaches will be compared in case of more than one jammer.

V - 6.1 Introduction to reception pattern shaping

This section introduces general reception pattern control. It starts with introduction of GPS antenna reception pattern and single patch reception pattern definition. Afterwards, the underlying signal processing for reception pattern control will be introduced.

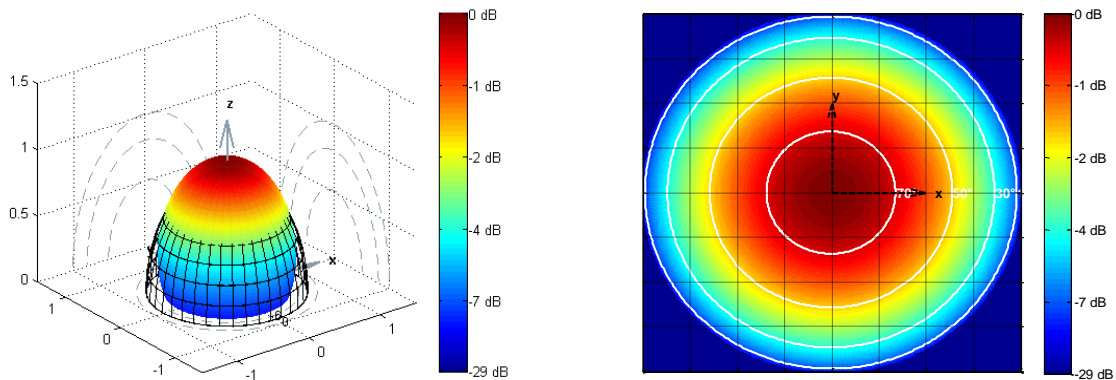
In the following, the gain pattern of a typical GPS antenna is shown. In this example, the GPS-704x antenna from Novatel is used. The reception pattern is scaled to 0dB.

CRPA's consist out of a set of microstrip patch antennas. For that purpose, in the following, also an example reception pattern of such a single patch is shown. In the example, the patch MPA 254 is used. This reception pattern is also scaled to 0dB.



V-22 reception pattern GPS-704X (scaled to 0dB)

V-23reception pattern GPS-704X (scaled to 0dB)



V-24 reception pattern MPA 254 (scaled to 0dB)

V-25reception pattern MPA 254 (scaled to 0dB)

This single antenna reception pattern has its maximum gain in zenith of the antenna hemisphere, which is bad, if for example the satellite is at a low elevation angle or if in case of high bank angles, a jammer points directly into the antenna zenith.

For this purpose, it is important to shape the reception pattern to get a maximum gain towards the wanted satellite and a minimum gain towards an identified jammer.

For reception pattern shaping, an array antenna, consisting out of more single antenna patches, is necessary.

There are basically three methods of reception pattern shaping.

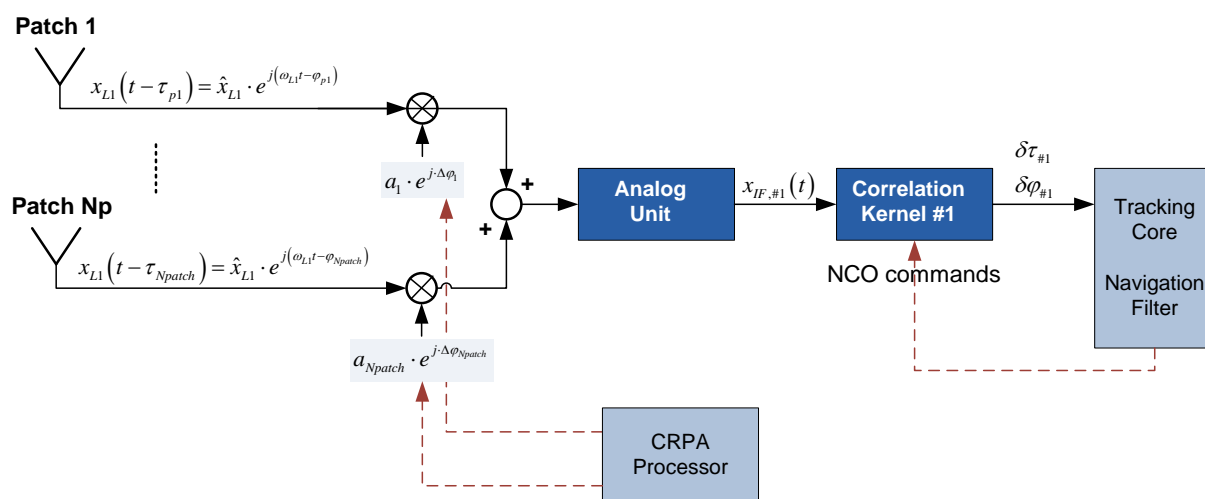
- I) Beam steering: The main lobe of the reception pattern is steered towards the satellite direction. Side lobes occur at arbitrary attitudes and may point directly towards a jammer direction

- II) Nulling: Towards one or more identified jammer directions, the reception pattern gain is minimized. Because there are no gain constraints towards the satellite direction, the antenna gain in satellite direction can be very low.
- III) Beamforming: The main lobe of the reception pattern is turned towards the satellite whereby at the same time, the reception pattern is minimized towards one or more jammer directions.

V - 6.1.1 Signal processing architecture and array geometry

In this first section on controlled radiation pattern antennas, the general signal processing architecture for beam steering is introduced. The following figures show the used array geometries in this and the following sections.

For reception pattern shaping, the received signal from each array element is modified in amplitude and phase. Afterwards all array elements are superimposed.



V-26 signal processing structure – reception pattern shaping

The signal processing architecture given in figure V-26, shows the typical approach for one tracking channel. In the given architecture, the reception pattern shaping is realized in L1 band. Another possibility is to realize the reception pattern shaping after the analog unit, at an intermediate frequency. (In this case, each antenna element would need its own analog unit). The complex weighting factors $a_n \cdot e^{j\Delta\phi_n}$ are calculated by a reception pattern processing unit, which is called “CRPA Processor” in the figure above.

In case of Nulling, this unit does not need any attitude information from the platform and can therefore be implemented in a standalone antenna, without any additional adaptations in the GPS receiver itself.

In case of beam forming, attitude and position information from the platform are needed, to calculate the complex weights. This is a disadvantage of beam forming, because therefore it’s only possible, if the GPS receiver provides a special interface. Another possibility is to realize beam

forming within the receiver. For that, the GPS receiver would need as many antenna interfaces as array elements.

V - 6.1.2 Uniform weighted unsteered reception pattern at L1 frequency

Before starting with beam forming and Nulling, this section introduces the basics of array signal processing and shows the natural unweighted reception pattern for the given antenna arrays.

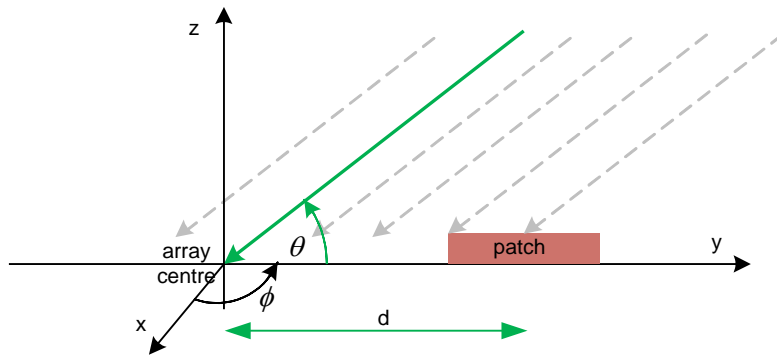
This section considers only single frequencies or small bandwidth signals, for which the single frequency assumption of the reception pattern is valid. Wide band jamming is considered in [112].

A very detailed derivation of the theory behind reception pattern control is given in [107].

The signals, received at the different antenna patches, are concentrated in one vector.

$$\mathbf{f}(t) = \begin{bmatrix} x_1(t - \tau_1) \\ \vdots \\ x_{N_p}(t - \tau_{N_p}) \end{bmatrix} \quad (V-37)$$

For the array center, the signal delay, respectively signal phase, is defined as zero, without any restrictions. The signal reception times at all other antenna patches are related to the array center. Depending on the distances of each patch from the array center and the direction of signal arrival, there will be some phase difference or delay τ between the signal received at the array center and the signal received at the corresponding patch.



V-27 Plan wave signal arrival

As input signal, a plane wave is assumed. The signal delay at the patch, related to the array center, is given as

$$\tau = \frac{d \cdot \cos(\theta)}{c} \quad (V-38)$$

Using the spatial signal arrival direction vector,

$$\mathbf{a} = \begin{bmatrix} -\cos(\theta) \cdot \cos(\phi) \\ -\cos(\theta) \cdot \sin(\phi) \\ -\sin(\theta) \end{bmatrix} \quad (V-39)$$

the delay at the corresponding antenna element gets

$$\tau_p = \frac{\mathbf{a}^T \cdot \mathbf{p}}{c} \quad (\text{V-40})$$

with \mathbf{p} being the position vector of the array element within the antenna coordinate system.

Instead of using time space, it is better to do reception pattern control in frequency space.

$$\mathcal{X}_p(\omega) = \int x(t - \tau_p) \cdot e^{-j\omega t} dt = e^{-j\omega\tau_p} \cdot \mathcal{X}(\omega) \quad (\text{V-41})$$

The received amplitude spectrum is equal at every antenna element. The only difference in frequency space is the phase.

Using equation (V-40),

$$\omega \cdot \tau_p = \omega \cdot \frac{\mathbf{a}^T \cdot \mathbf{p}}{c} \quad (\text{V-42})$$

and defining the wave number \mathbf{k} as

$$\mathbf{k} = \frac{\omega \cdot \mathbf{a}}{c} = \frac{2 \cdot \pi \cdot \mathbf{a}}{\lambda} \quad (\text{V-43})$$

the signal in frequency domain can be written as

$$\mathcal{X}_p(\omega) = e^{-j\mathbf{k}^T \cdot \mathbf{p}} \cdot \mathcal{X}(\omega) \quad (\text{V-44})$$

Similar to (V-37), all patch signals can be combined into one vector in frequency space.

$$\mathbf{F}(\omega) = \begin{bmatrix} e^{-j\mathbf{k}^T \cdot \mathbf{p}_1} \\ \vdots \\ e^{-j\mathbf{k}^T \cdot \mathbf{p}_{N_p}} \end{bmatrix} \cdot \mathcal{X}(\omega) = \mathbf{v}_k(\mathbf{k}) \cdot \mathcal{X}(\omega) \quad (\text{V-45})$$

The vector $\mathbf{v}_k(\mathbf{k})$ is defined as array manifold vector.

In general, the reception pattern is calculated according to the following equation, whereby \mathbf{w} is called the weight vector being calculated by the CRPA processor.

$$B(\omega, \theta, \phi) = \mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k}) \quad (\text{V-46})$$

In order to get the unsteered and uniformly weighted reception pattern, the weight vector is defined as

$$\mathbf{w} = \mathbf{1}^{N_p \times 1} \quad (\text{V-47})$$

The reception gain pattern is typically given as the power reception pattern which is calculated as

$$|B(\omega, \theta, \phi)|^2 = (\mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k}))^T \cdot (\mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k})) = \mathbf{v}_k^T(\mathbf{k}) \cdot \mathbf{w} \cdot \mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k}) \quad (\text{V-48})$$

Reception pattern forming causes a phase shift, with relation to the unmodified received signal. These phase shift can be calculated according to the next equation.

$$\varphi(\omega, \theta, \phi) = a \tan \left(\frac{\text{Im}\{B(\omega, \theta, \phi)\}}{\text{Re}\{B(\omega, \theta, \phi)\}} \right) = a \tan \left(\frac{\text{Im}\{\mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k})\}}{\text{Re}\{\mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k})\}} \right) \quad (\text{V-49})$$

The reception pattern according to (V-48) assumes a uniform element reception pattern.

In case of high platform dynamic and therefore high dynamic in changing reception pattern weights, the resulting phase shift may cause some tracking errors, because the tracking filter cannot differ if the phase shift is due to line of sight dynamic or reception pattern control. An analysis to which extent this effect worsens tracking is given in a later section.

The following equation shows a more realistic approach by considering for each array element its real pattern. The gain $a_1(\theta, \phi)$ of a single element is not uniform across the hemisphere, it is dependent on the elevation and if it is not a circular patch, also dependent on the azimuth.

$$B(\omega, \theta, \phi) = \mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k}, \theta, \phi) = \mathbf{w}^T \cdot \begin{bmatrix} a_1(\theta, \phi) \cdot e^{-j\mathbf{k}^T \cdot \mathbf{p}_1} \\ \vdots \\ a_{N_p}(\theta, \phi) \cdot e^{-j\mathbf{k}^T \cdot \mathbf{p}_{N_p}} \end{bmatrix} \quad (\text{V-50})$$

The figures in the following section show the natural unweighted reception patterns of the array geometries, which will be considered in this thesis. For each architecture, the natural reception pattern with a uniform element gain $a_k(\theta, \phi) = 1$ and also with the real element gain is shown.

V - 6.2 Benefits of reception pattern control in unjammed scenarios

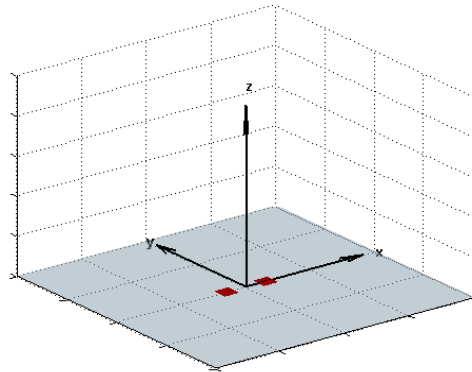
In this section, the unweighted reception pattern of different antenna array geometries will be considered. It will be analyzed, which percentage of the hemisphere and therefore which percentage the satellite positions are attenuated more than a defined value.

The question will be answered, if without any jammer, the single patch antenna reception pattern provides a better hemisphere coverage than different uniformly weighted array reception pattern.

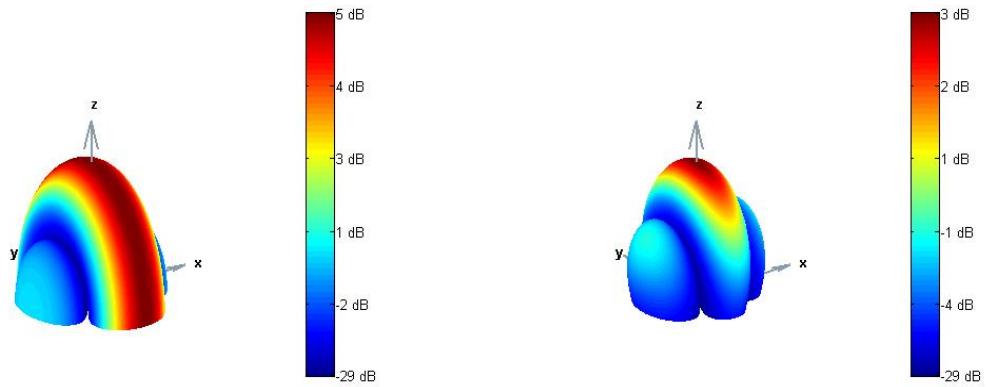
The following figures show the uniformly weighted reception pattern in a polar form, for all array geometries considered in this thesis.

The figures compare the reception pattern, if perfect, uniform element reception pattern are used and in the other case, if real non uniform element reception pattern are used, as given in figure V-24.

Array geometry: linear, 2 elements, radius $\frac{\lambda}{2}$ (lin2_051)

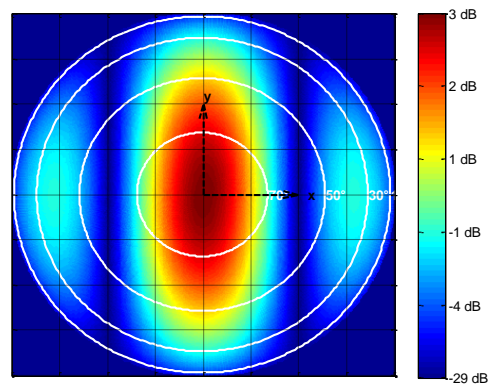
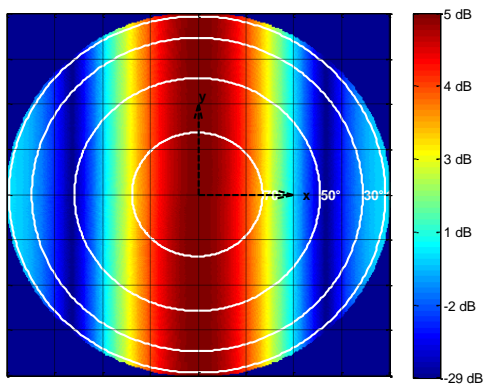


V-28 array geometry



V-29 natural array pattern –uniform element gain

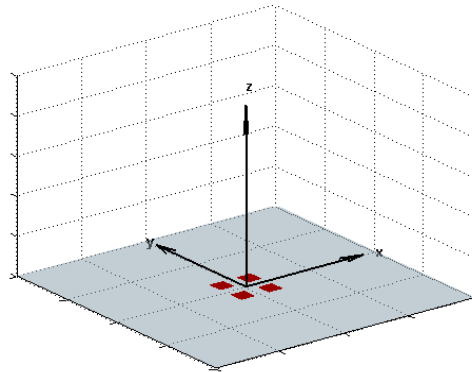
V-30 natural array pattern –real element gain



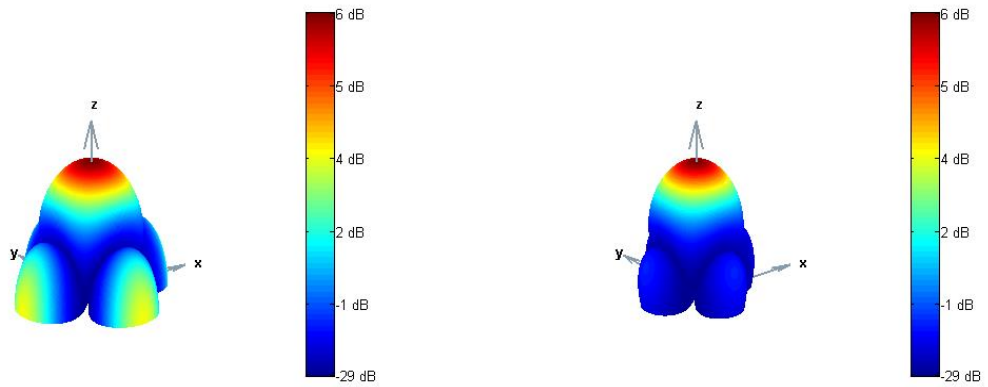
V-31 natural array pattern –uniform element gain

V-32 natural array pattern –real element gain

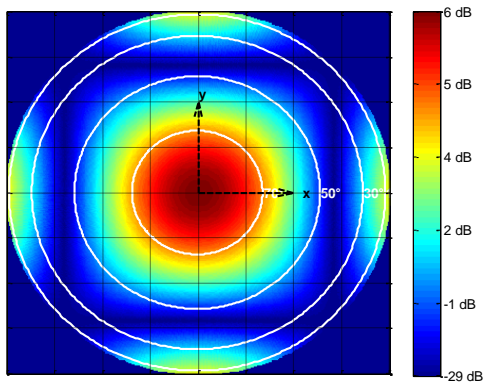
Array geometry: rectangular, 4 elements, radius $\frac{1}{\sqrt{2}} \cdot \lambda$ (rect4_051)



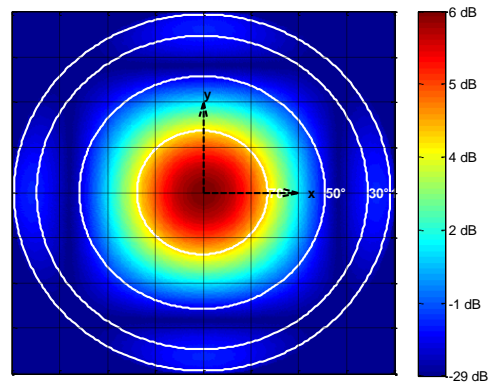
V-33 array geometry



V-34 natural array pattern –uniform element gain



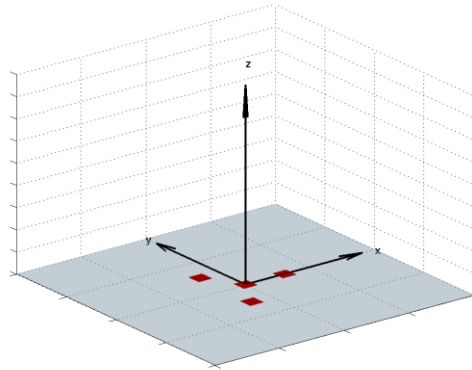
V-35 natural array pattern –real element gain



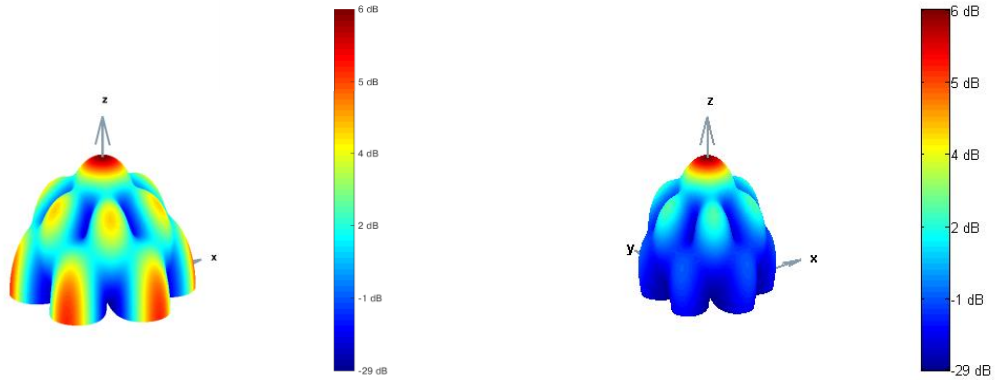
V-36 natural array pattern –uniform element gain

V-37 natural array pattern –real element gain

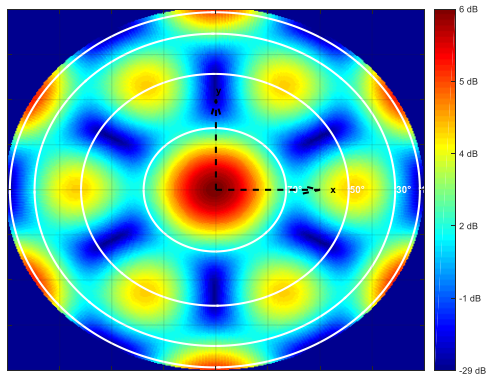
Array geometry: y-shape, 4 elements, radius $\frac{\lambda}{2}$ (y3m1_051)



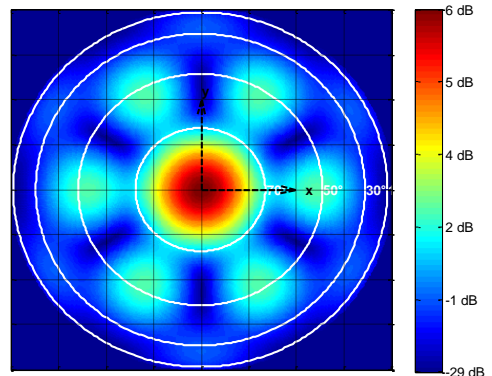
V-38 array geometry



V-39 natural array pattern –uniform element gain



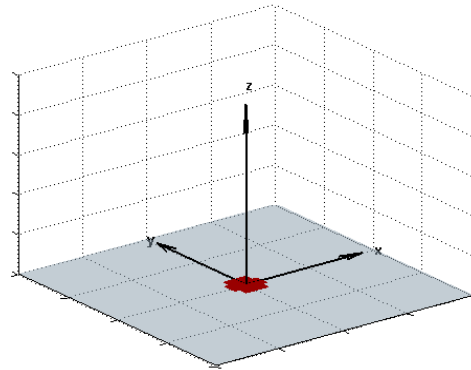
V-40 natural array pattern –real element gain



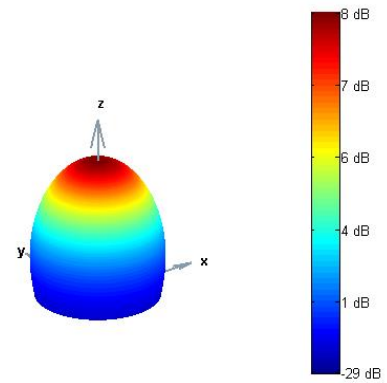
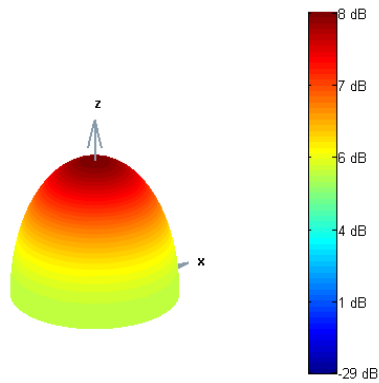
V-41 natural array pattern –uniform element gain

V-42 natural array pattern –real element gain

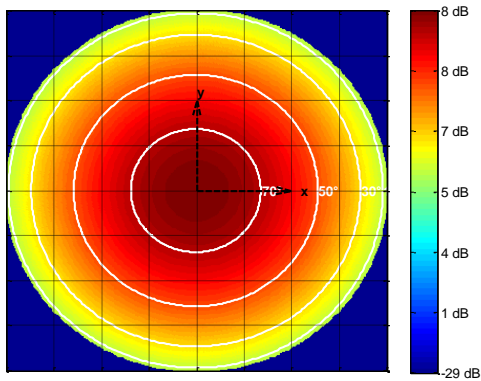
Array geometry: hexagon with middle element, radius $\frac{\lambda}{4}$ (c6m1_0251)



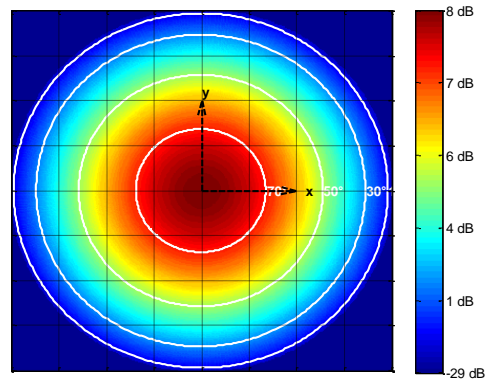
V-43 array geometry



V-44 natural array pattern –uniform element gain



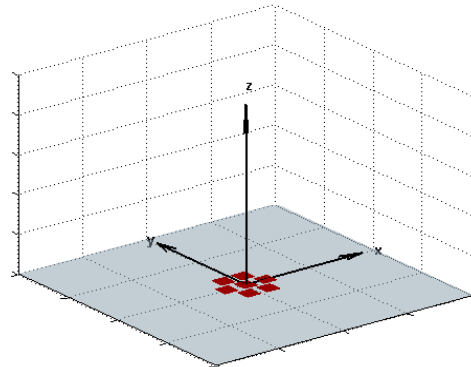
V-45 natural array pattern –real element gain



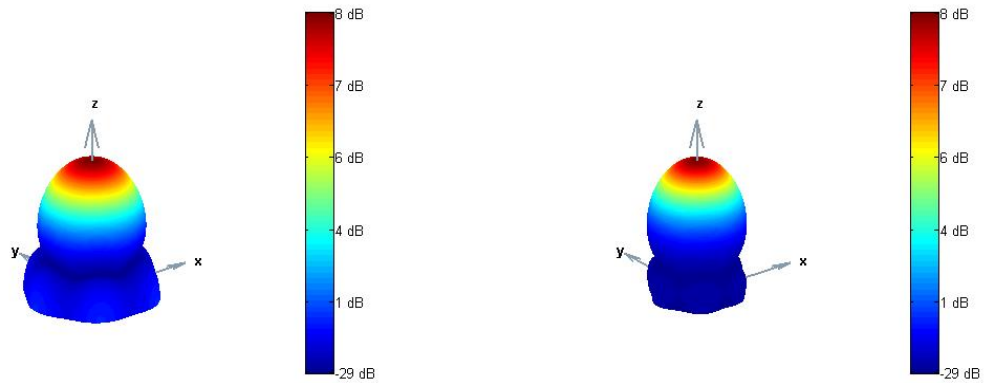
V-46 natural array pattern –uniform element gain

V-47 natural array pattern –real element gain

Array geometry: hexagon with middle element, radius $\frac{\lambda}{2}$ (c6m1_051)

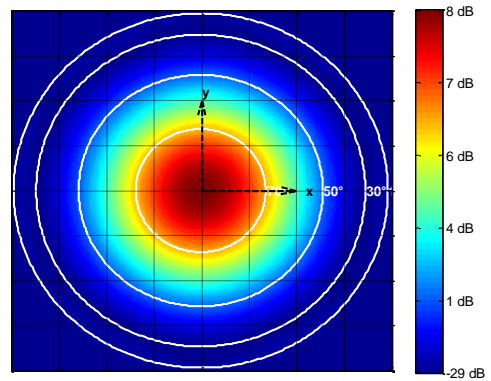
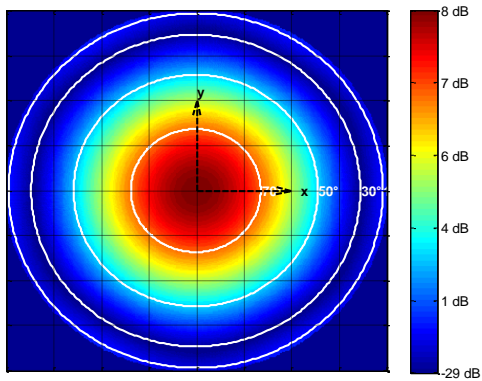


V-48 array geometry



V-49 natural array pattern –uniform element gain

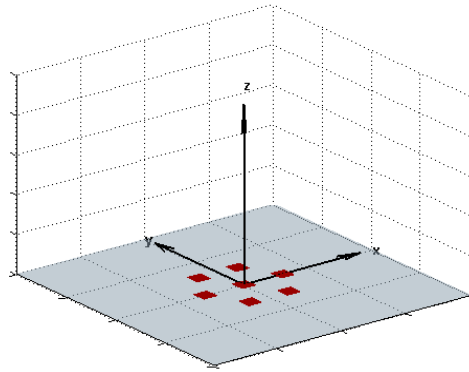
V-50 natural array pattern –real element gain



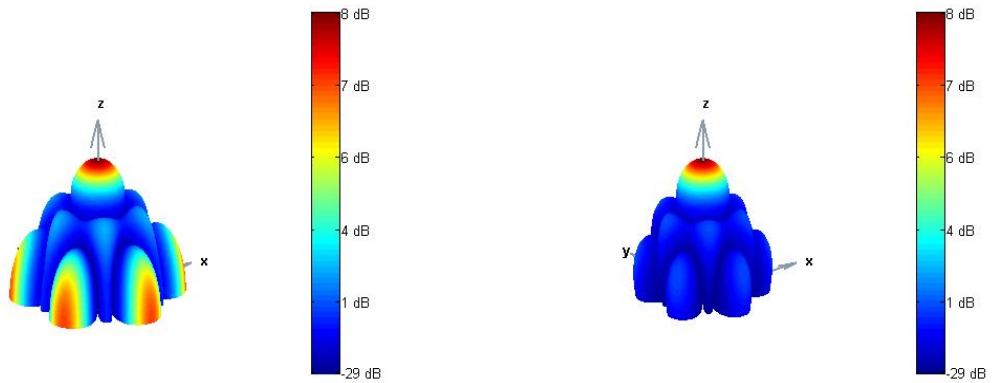
V-51 natural array pattern –uniform element gain

V-52 natural array pattern –real element gain

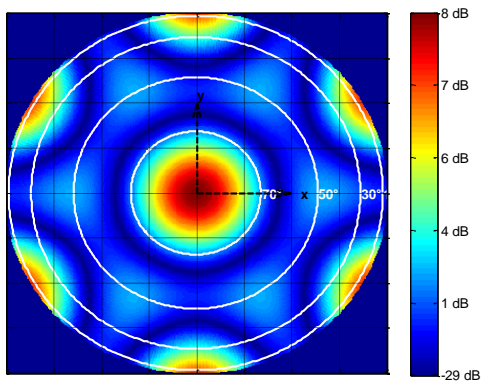
Array geometry: hexagon with middle element, radius λ (c6m1_11)



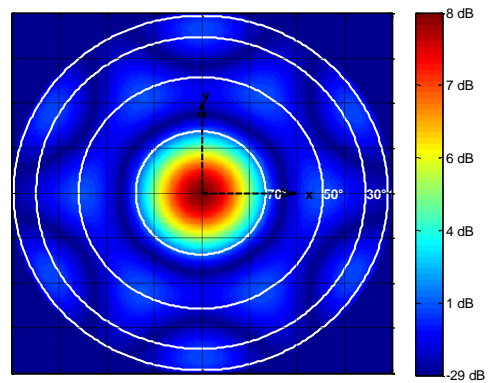
V-53 array geometry



V-54 natural array pattern –uniform element gain



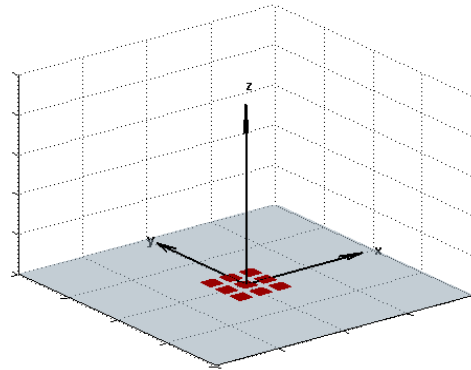
V-55 natural array pattern –real element gain



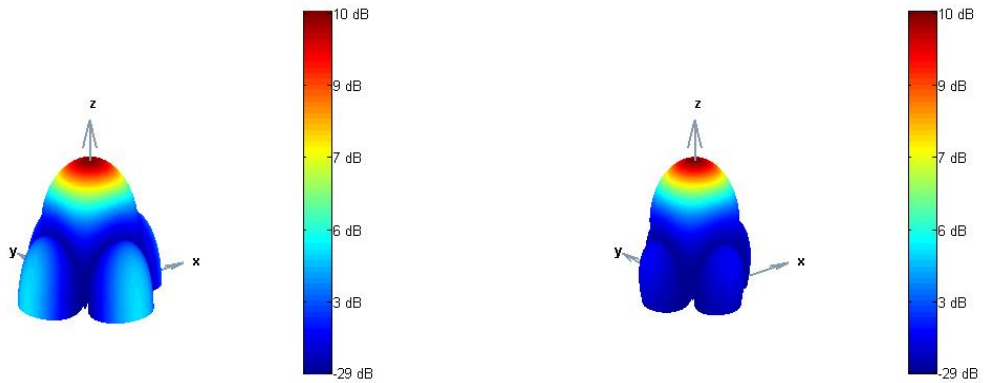
V-56 natural array pattern –uniform element gain

V-57 natural array pattern –real element gain

Array geometry: rectangle field, spacing $\frac{\lambda}{2}$ (rect9_051)

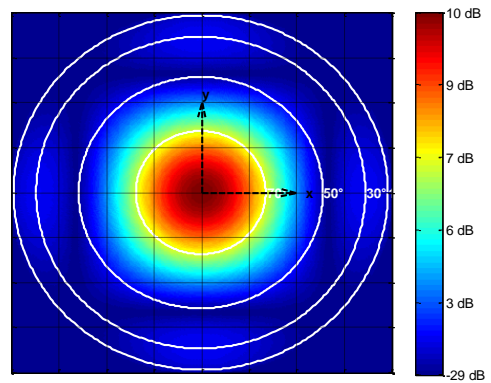
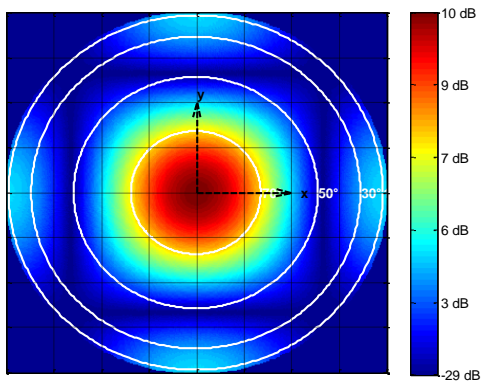


V-58 array geometry



V-59 natural array pattern –uniform element gain

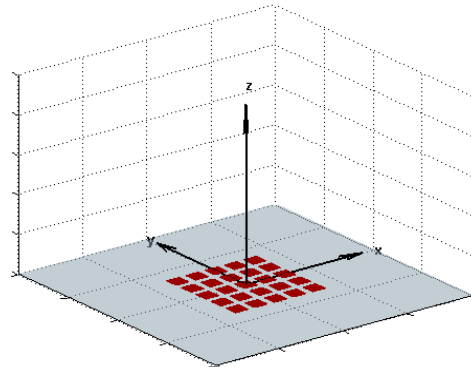
V-60 natural array pattern –real element gain



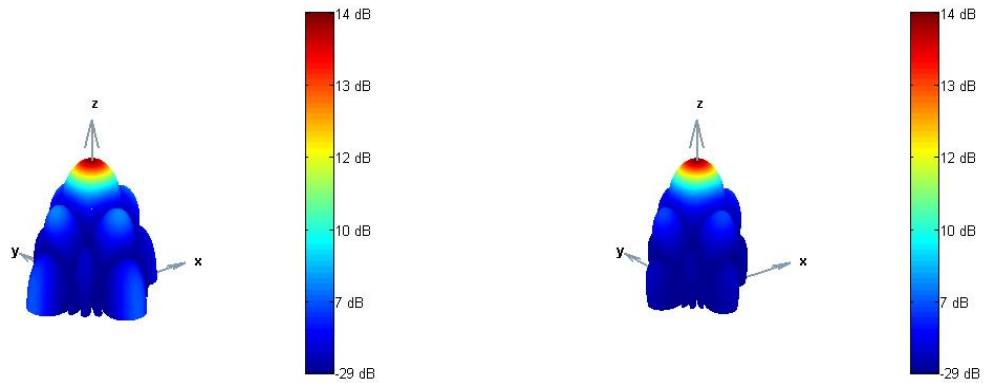
V-61 natural array pattern –uniform element gain

V-62 natural array pattern –real element gain

Array geometry: rectangle field, spacing $\frac{\lambda}{2}$ (rect25_051)

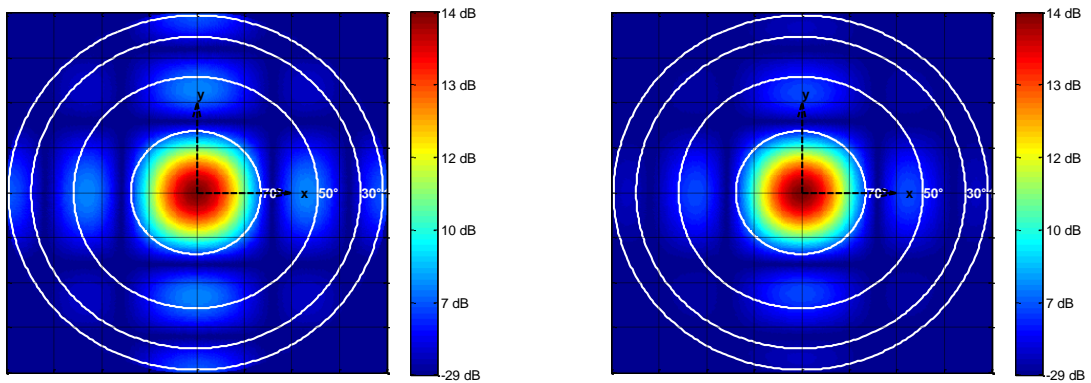


V-63 array geometry



V-64 natural array pattern –uniform element gain

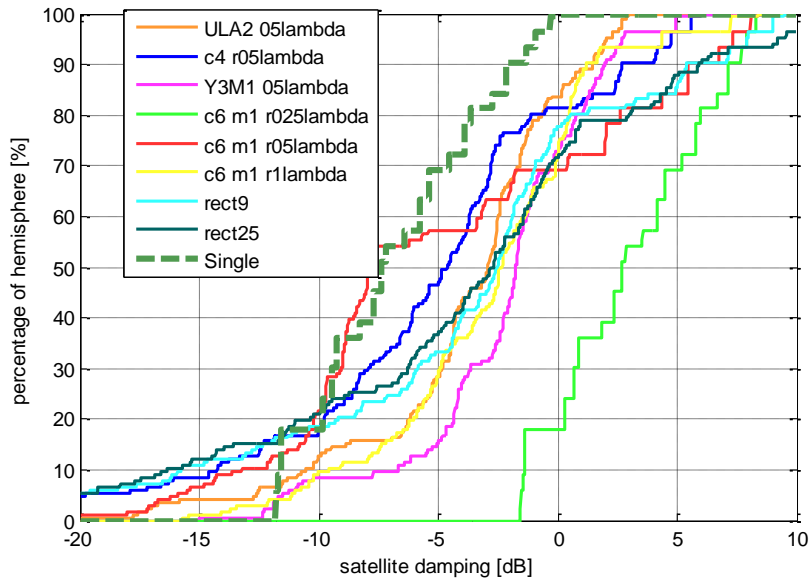
V-65 natural array pattern –real element gain



V-66 natural array pattern –uniform element gain

V-67 natural array pattern –real element gain

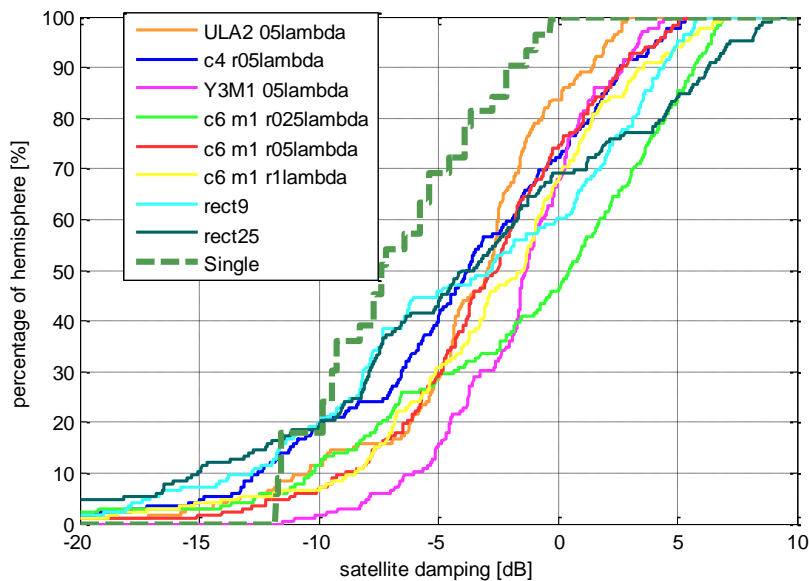
The following two figures show the cumulative percentage of the hemisphere, having a gain less than the given value on the abscissa.



V-68 Satellite damping – unsteered array

In figure V-68, the unsteered, uniformly weighted array reception pattern is compared to a normal GPS antenna reception pattern. The figure shows, which percentage of hemisphere has a damping of more than the corresponding value of abscises.

Besides the very small array “c6m1_r025lambda”, all arrays show a damping of more than -10dB on a significant part of the hemisphere. That means that for more than 10% of the satellite positions on hemisphere, the signal to noise ratio is reduced by more than 10 dB, which increases the risk of loss of lock tremendously. (The signal to noise ratio decreases due to satellite damping). The array geometry “c6m1_r025lambda” shows a very small satellite damping, but this geometry shows a bad performance in case of nulling, which will be shown in a following section.



V-69 Satellite damping – array steering to elevation 5°

In figure V-69, the array is steered towards an elevation of 5° and again compared to the normal GPS antenna reception pattern.

All array geometries show a significant satellite damping on more than 10% of the hemisphere.

Up to a satellite damping of 12 dB, the array reception pattern shows less satellite damping than the single element reception pattern.

But the single element reception pattern has no satellite damping beyond 12dB. That's the real advantage of the single element, because a satellite damping of more than 10dB poses already a high risk for loss of lock in bad signal to noise environments.

All in all, if no jammer is present and the satellite position is not known within the local antenna coordinate system, it's better to switch to a single patch reception antenna.

Important to notice is, that if the satellite position is known, steering the reception pattern towards the satellite, gives a higher gain as the normal GPS antenna reception pattern! This should only be used if there is no jammer, because beam steering causes high-gain side lobes which could be directed towards a potential jammer.

V - 6.3 Optimal array geometry with real gain patch reception pattern

V - 6.3.1 Introduction to beam forming

Beam forming describes the most advantageous, but also complex method of reception pattern shaping for array antennas. The goal of beam forming is to direct the main lobe of the antenna towards a wanted satellite direction. At the same time, beam forming can minimize the reception pattern gain respectively maximize damping towards $N_{patch} - N_{MainLobe} - 1$ jammer directions.

The literature provides different approaches for beam forming. In [108], [113], [114], [115], [116], [117] and [107] a comprehensive summary of many beam forming approaches together with a comparison can be found.

The focus of this section is to evaluate the influence of the array geometry on beam forming performance and to identify the effect of non-uniform, unknown element reception pattern. The focus is not the comparing of different beam forming algorithm. Therefore, for further considerations, a promising beam forming algorithm, the minimum variance distortion less beam forming algorithm, will be used.

V - 6.3.2 Minimum variance distortion less response beam forming (MVDR)

This chapter derives the minimum variance distortion less response beam forming algorithm, which can be also found in [107].

The received signal from one patch in frequency space can be written as

$$\mathcal{X}(\omega) = \mathcal{X}_s(\omega) + \mathcal{X}_{noise}(\omega) \tag{V-51}$$

The received signals from all antenna elements of the array antenna can be written by the following vector representation.

$$\mathbf{x}(\omega) = \mathbf{x}_s(\omega) + \mathbf{x}_{noise}(\omega) \quad (V-52)$$

The vector $\mathbf{x}_s(\omega)$ contains the received satellite signals from all array elements. It can be split into the absolute value of power spectral density $\mathcal{F}(\omega)$ and the array manifold vector $\mathbf{v}(\mathbf{k}_s, \mathbf{p})$, containing the phase relationship.

$$\mathbf{x}_s(\omega) = \mathcal{F}(\omega) \cdot \mathbf{v}(\mathbf{k}_s, \mathbf{p}) \quad (V-53)$$

The received power spectral density is the same at all array elements. The only difference is the phase relationship.

$$\mathbf{v}(\mathbf{k}_s, \mathbf{p}) = \begin{bmatrix} e^{-j\mathbf{k}_s^T \cdot \mathbf{p}_1} \\ \vdots \\ e^{-j\mathbf{k}_s^T \cdot \mathbf{p}_{Npatch}} \end{bmatrix} \quad (V-54)$$

The vector \mathbf{p}_n is the vector, pointing from the array center to the corresponding position of the patch element in the antenna array.

The noise consists out of the natural antenna noise, which power spectral density is typically $-200 \frac{dBW}{Hz}$. The noise signals from all array elements can also be combined in one vector.

$$\mathbf{x}_{noise}(\omega) = \mathcal{N}(\omega) \cdot \mathbf{1} + \begin{bmatrix} \mathcal{J}_1(\omega) \\ \vdots \\ \mathcal{J}_{Npatch}(\omega) \end{bmatrix} = \mathcal{N}(\omega) \cdot \mathbf{1} + \mathcal{J}(\omega) \cdot \begin{bmatrix} e^{-j\mathbf{k}_j^T \cdot \mathbf{p}_1} \\ \vdots \\ e^{-j\mathbf{k}_j^T \cdot \mathbf{p}_{Npatch}} \end{bmatrix} \quad (V-55)$$

The received jammer signal in frequency space is, except the phase relationship, also equal at every antenna element.

The MVDR beam forming algorithm needs the noise covariance matrix, which can be written in frequency space according to the next equation.

$$\begin{aligned} \mathbf{S}_{noise} &= \mathcal{F} \left\{ \mathcal{E} \left\{ \mathbf{x}_{noise}(t) \cdot \mathbf{x}_{noise}^T(t) \right\} \right\} = \mathcal{F} \left\{ \mathcal{E} \left\{ \mathbf{n}(t) \cdot \mathbf{n}^T(t) \right\} \right\} + \mathcal{F} \left\{ \mathbf{J}(t) \cdot \mathbf{J}^T(t) \right\} \\ &= \sigma_{noise}^2 \cdot \mathbf{I}^{Npatch \times Npatch} + |\mathcal{J}(\omega)|^2 \cdot \begin{bmatrix} 1 & e^{j\mathbf{k}_j^T \cdot (\mathbf{p}_2 - \mathbf{p}_1)} & \dots & e^{j\mathbf{k}_j^T \cdot (\mathbf{p}_{Npatch} - \mathbf{p}_1)} \\ e^{j\mathbf{k}_j^T \cdot (\mathbf{p}_1 - \mathbf{p}_2)} & 1 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\mathbf{k}_j^T \cdot (\mathbf{p}_1 - \mathbf{p}_{Npatch})} & e^{j\mathbf{k}_j^T \cdot (\mathbf{p}_2 - \mathbf{p}_{Npatch})} & \dots & 1 \end{bmatrix} \end{aligned} \quad (V-56)$$

The noise of two different array elements can assumed to be uncorrelated.

In the receiver, this noise covariance matrix is calculated using the fast Fourier transformation. For that, the signal from each antenna element is sampled. By using a moving time window, the samples within this window are used to calculate the frequency representation.

$$\mathbf{X}_{noise,record}(N_{window}, \omega) = \mathcal{N}(N_{window}, \omega) \cdot \mathbf{1} + \begin{bmatrix} \mathcal{J}_1(N_{window}, \omega) \\ \vdots \\ \mathcal{J}_{Npatch}(N_{window}, \omega) \end{bmatrix} \quad (V-57)$$

The windowed records in frequency space are multiplied afterwards according to the next equation, which gives the noise covariance matrix.

$$\mathbf{S}_n = \mathbf{X}_{noise,record} \cdot \mathbf{X}_{noise,record}^T \quad (V-58)$$

The line of sight direction to the jammer within the antenna coordinate system is included in the noise covariance matrix. The more powerful the received jamming signal is, the better works the direction estimation. Different methods for jammer attitude estimation is given in [118].

This noise covariance matrix is one advantage of this beam forming approach, because no separate jammer direction of arrival estimation is necessary.

The goal of the minim variance distortion loss response beam former is, to find an antenna weighting vector \mathbf{w} , reducing the variance of the following expression.

$$\mathbf{Y}(\omega) = \mathbf{w}^H \cdot \mathbf{X}(\omega) \quad (V-59)$$

In order to prevent the trivial solution and also to force the solution to build a main lobe towards the satellite direction, the following side condition for the weighting vector is necessary.

$$\mathbf{w}^H \cdot \mathbf{v}(\mathbf{k}_S, \mathbf{p}) = 1 \quad (V-60)$$

Important to notice is, if the amplitude of the satellite directed main lobe is forced to be 1, according to equation (V-60), the advantage of the natural array again is destroyed. Therefore, depending on the array elements used for superposition, the following side condition should be used.

$$\mathbf{w}^H \cdot \mathbf{v}(\mathbf{k}_S, \mathbf{p}) = G_{Array} = 10 \cdot \log_{10}(N_{elements}) \quad (V-61)$$

This gain is called “natural gain”, because no amplification is necessary. This gain is realized by a phase synchronous superposition of the incoming signals from the different array elements. The phase synchronous superposition increases the effective signal to noise ratio about the same amount, because the noise of the different array elements is assumed to be uncorrelated and do not add up.

A further amplification $G_{Array} > 10 \cdot \log_{10}(N_{elements})$ brings no benefit regarding the signal to noise

ratio, because if a synthetic amplification factor is used, the noise is amplified about the same amount and the signal to noise ratio remains unchanged.

The weighting vector is calculated by solving a Lagrange optimization problem according to the next equation.

$$\begin{aligned} L &= \mathcal{E} \left\{ (\mathbf{w}^H \cdot \mathbf{X}(\omega)) \cdot (\mathbf{w}^H \cdot \mathbf{X}(\omega))^H \right\} + \lambda \cdot (\mathbf{w}^H \cdot \mathbf{v}(\mathbf{k}_s, \mathbf{p}) - 1) + \lambda^* \cdot (\mathbf{v}^H(\mathbf{k}_s, \mathbf{p}) \cdot \mathbf{w} - 1) \\ &= \mathbf{w}^H \cdot \mathbf{S}_n(\omega) \cdot \mathbf{w} + \lambda \cdot (\mathbf{w}^H \cdot \mathbf{v}(\mathbf{k}_s, \mathbf{p}) - 1) + \lambda^* \cdot (\mathbf{v}^H(\mathbf{k}_s, \mathbf{p}) \cdot \mathbf{w} - 1) \end{aligned} \quad (\text{V-62})$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w}^H \cdot \mathbf{S}_n + \lambda^* \cdot \mathbf{v}^H = 0 \quad (\text{V-63})$$

$$\frac{\partial L}{\partial \lambda} = \mathbf{w}^H \cdot \mathbf{v} - 1 = 0 \quad (\text{V-64})$$

The gradient with respect to \mathbf{w} and λ gives two equations which need to be solved. From (V-63), \mathbf{w} is calculated and used in (V-64) to calculate λ .

$$\lambda = -(\mathbf{v}^H \cdot \mathbf{S}_n^{-1} \cdot \mathbf{v})^{-1} \quad (\text{V-65})$$

With (V-65), the weight vector gets

$$\mathbf{w} = (\mathbf{v}^H \cdot \mathbf{S}_n^{-1} \cdot \mathbf{v})^{-1} \cdot \mathbf{v}^H \cdot \mathbf{S}_n^{-1} \quad (\text{V-66})$$

In order to evaluate the beam forming and nulling performance and to compare different array geometries, the jammer is permuted over the hemisphere using a geodetic grid. For every jammer position, also the satellite is permuted over the whole hemisphere, whereby the gain towards the satellite, as well as the damping towards the jammer, is evaluated.

Generally, antijam is defined according to the next equation,

$$\text{Antijam}(\theta_J, \phi_J, \theta_S, \phi_S) = G(\theta_J, \phi_J) - G(\theta_S, \phi_S) \quad (\text{V-67})$$

with $G(\theta_J, \phi_J)$ being the reception pattern gain respectively damping in [dB] towards the actual jammer direction and $G(\theta_S, \phi_S)$ being the reception pattern gain towards the actual satellite direction.

V - 6.3.3 Optimal array geometry for MVDR beam forming

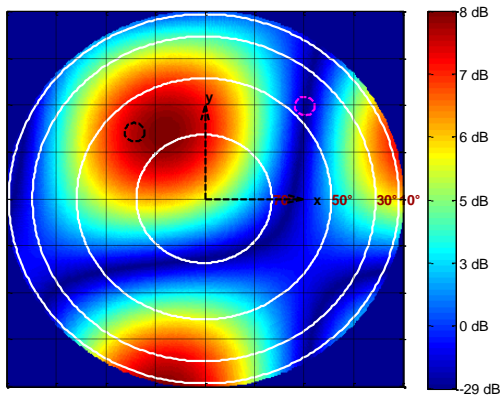
In this section, the optimal array geometry for MVDR beam forming will be identified. Additionally, the effect of using non uniform and unknown real element reception pattern will be analyzed.

The introduced minimum variance distortion less beam forming approach from the previous section, assumes for each single array element a uniform gain reception pattern with a gain of 0 dB

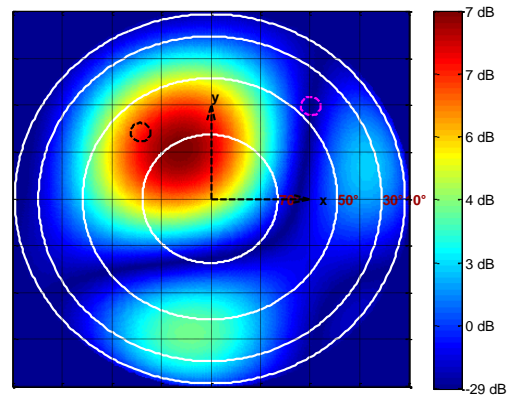
on the hemisphere. The real reception of each single array element is non uniform as given in figure V-24.

In order to get an impression of different reception pattern, given uniform 0dB real element reception pattern, the following figures show for a defined satellite – jammer position, the corresponding MVDR reception pattern.

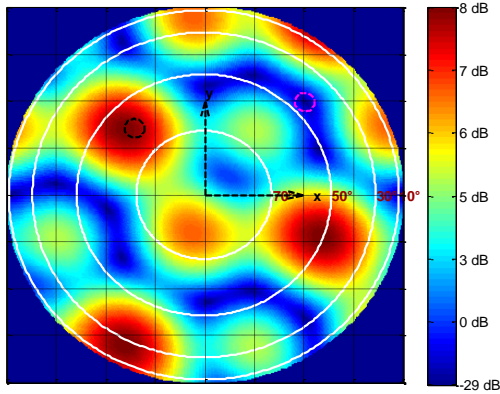
$$\mathbf{\theta}_S = \begin{bmatrix} \phi_S = 135^\circ \\ \theta_S = 60^\circ \end{bmatrix}, \quad \mathbf{\theta}_J = \begin{bmatrix} \phi_J = 45^\circ \\ \theta_J = 45^\circ \end{bmatrix} \quad (V-68)$$



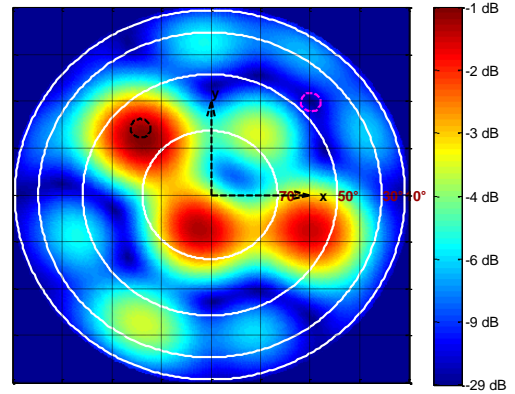
V-70 Uniform Patch Gain – rect4_05I



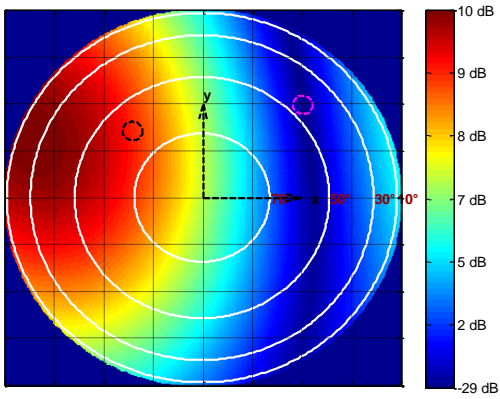
V-71 Real Patch Gain – rect4_05I



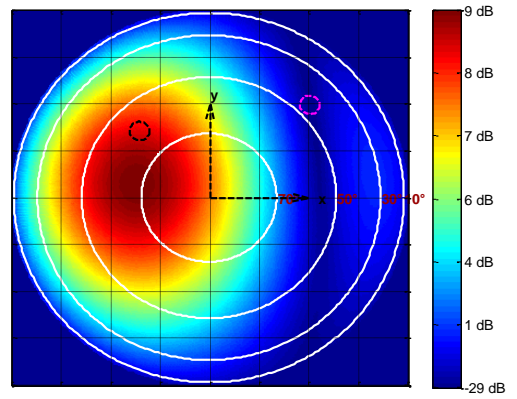
V-72 Uniform Patch Gain – y3m1_05I



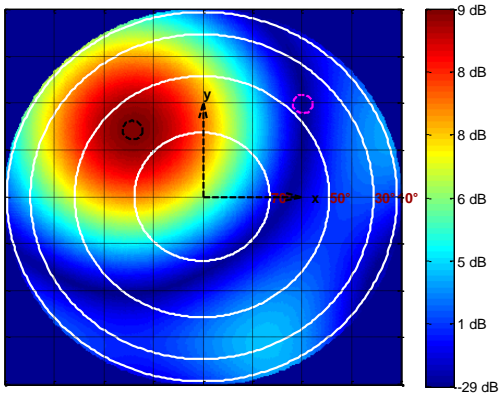
V-73 Real Patch Gain – y3m1_05I



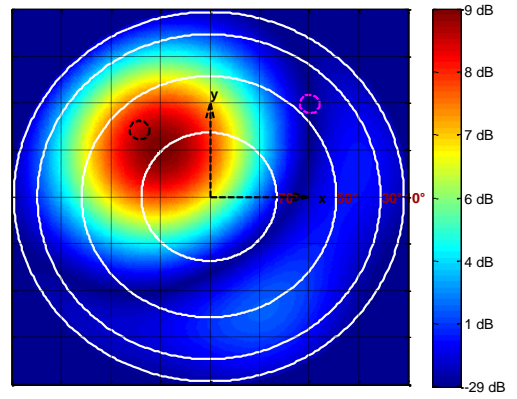
V-74 Uniform Patch Gain – c6m1_025I



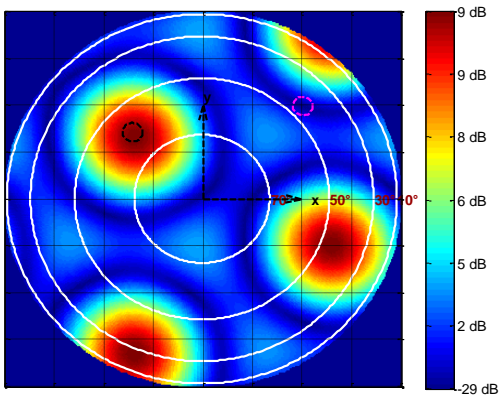
V-75Real Patch Gain – c6m1_025I



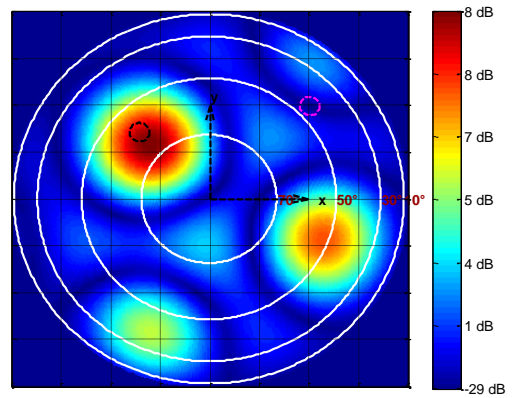
V-76 Uniform Patch Gain – c6m1_05I



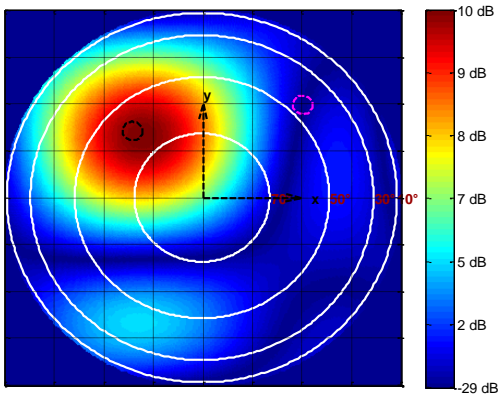
V-77Real Patch Gain – c6m1_05I



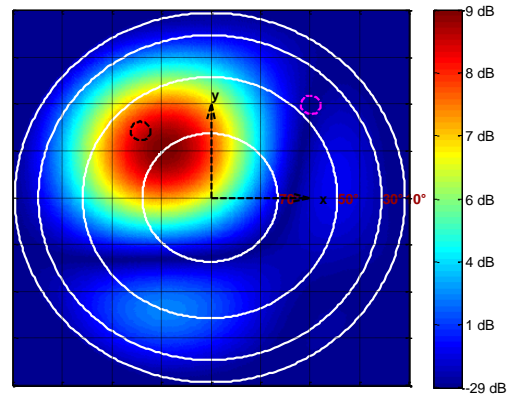
V-78 Uniform Patch Gain – c6m1_1I



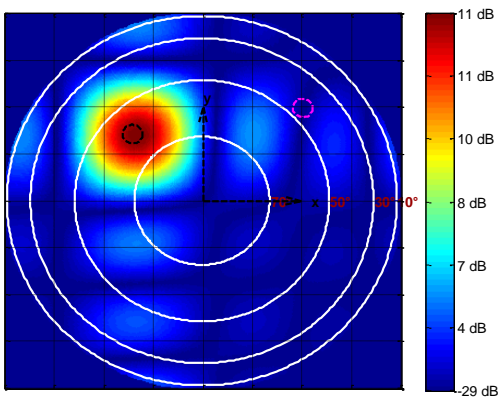
V-79Real Patch Gain – c6m1_1I



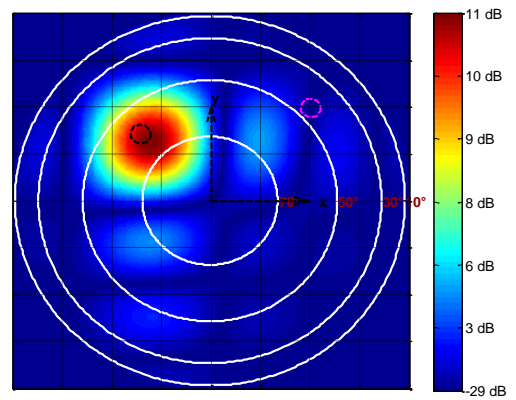
V-80 Uniform Patch Gain – rect9_05I



V-81Real Patch Gain – rect9_05I



V-82 Uniform Patch Gain – rect25_05I

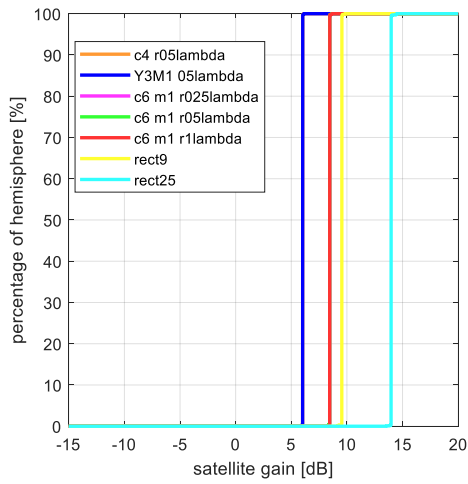


V-83Real Patch Gain – rect25_05I

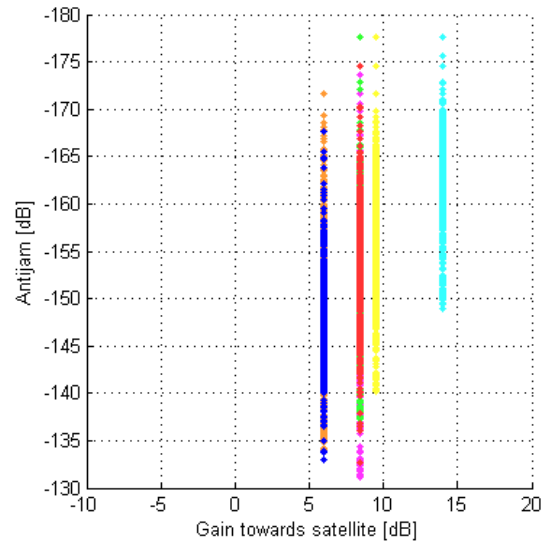
In the following figures, again the cumulative percentage of satellite damping and antijam across the hemisphere is shown, for

- MVDR beam forming using 0dB uniform element reception pattern
- MVDR beam forming using non uniform real element reception pattern

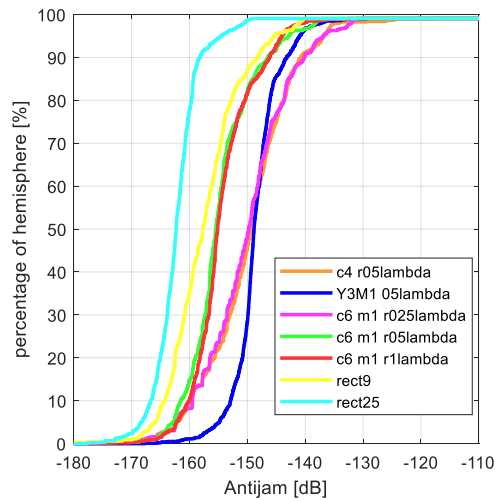
Satellite gain and antijam distribution – MVDR beam forming with uniform 0dB element reception pattern



V-84 cumulated sat gain

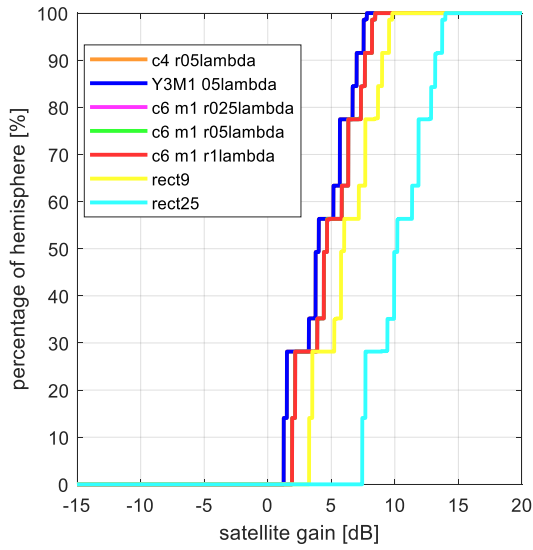


V-85 antijam vs. satellite gain distribution

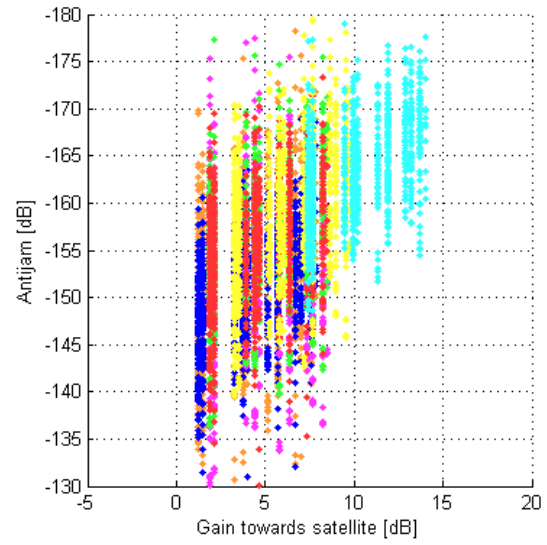


V-86 cumulated antijam

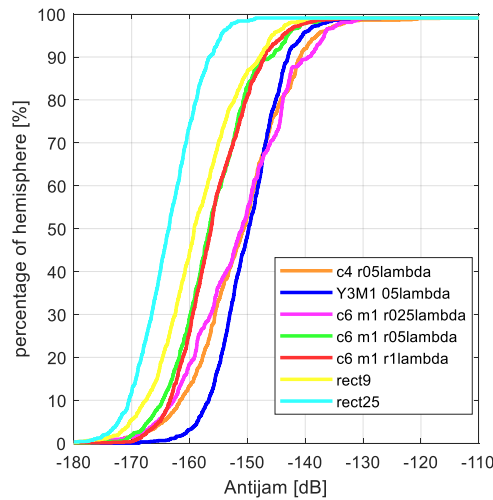
Satellite gain and antijam distribution – MVDR beam forming with real non uniform element reception pattern (maximum gain of single patch normed to 0dB)



V-87 cumulated sat gain – MVDR beam forming



V-88 antijam vs. satellite gain distribution



V-89 cumulated antijam

The figures V-87, V-88 and V-89 compare the performance of different array geometries for the MVDR beam forming approach, where the real non uniform single element reception pattern was used.

It gets obvious, that arrays having many array elements, show the best satellite gain. The satellite gain is the most important criteria for pattern selection, because already 10dB damping towards a satellite would increase the risk for loss of lock tremendously, given bad signal to noise environments.

In figure V-84, the relationship $G_{Array} = 10 \cdot \log_{10}(N_{elements})$ gets obvious. Depending on the number of array elements, the cumulated gain towards the satellite changes. Additionally, the analysis shows that the gain towards the satellite does not depend on the satellite azimuth and elevation – given

array elements with a uniform 0dB spherical gain pattern.

In case of non-uniform element reception pattern, the maximum satellite gain remains G_{Array} , but depends on the satellite elevation within the antenna coordinate system. The maximum gain is not changed, because even for real non uniform element reception pattern, the maximum gain for each patch element is 0dB. At low elevations, the used non uniform element reception patterns have a gain lower than 0dB, which lead also to a lower array reception gain. The steps in figure V-87 correlate with the number of different elevations used for analysis, on the geodetic dome.

The rectangular array having 25 elements is by far the best one, but difficult to implement on small flight platforms. Therefore, for further analysis, the rectangular element with 9 elements and the circular array “c6_m1_r1lambda” will be used as MVDR beam forming reference.

Comparing the antijam in case of a uniform patch gain and a real patch gain, it gets obvious that regarding antijam, an unknown respectively not considered non uniform patch gain has only little influence. The antijam is still high enough. (Important to notice at this point is that the high antijam is due to the assumption of antenna arrays having no geometry errors and due to the assumption of no attitude error towards the satellite and jammer. In later sections, the antijam degradation will be analyzed, given non-perfect antenna arrays and non-perfect attitude information)

In order to identify the best geometry, satellite gain is the most important criteria, because already 10dB lower satellite gain can cause loss of lock in bad signal to noise environments.

A compensation of the lower satellite gains provides no benefit regarding signal to noise ratio. Applying some amplification would amplify the noise about the same amount and leaves the signal to noise ratio therefore unchanged.

From that perspective, optimizing the element reception pattern is the only possibility, if large arrays are not possible.

V - 6.4 Multi satellite beam forming

In the previous chapter, beam forming was introduced for one satellite and one jammer.

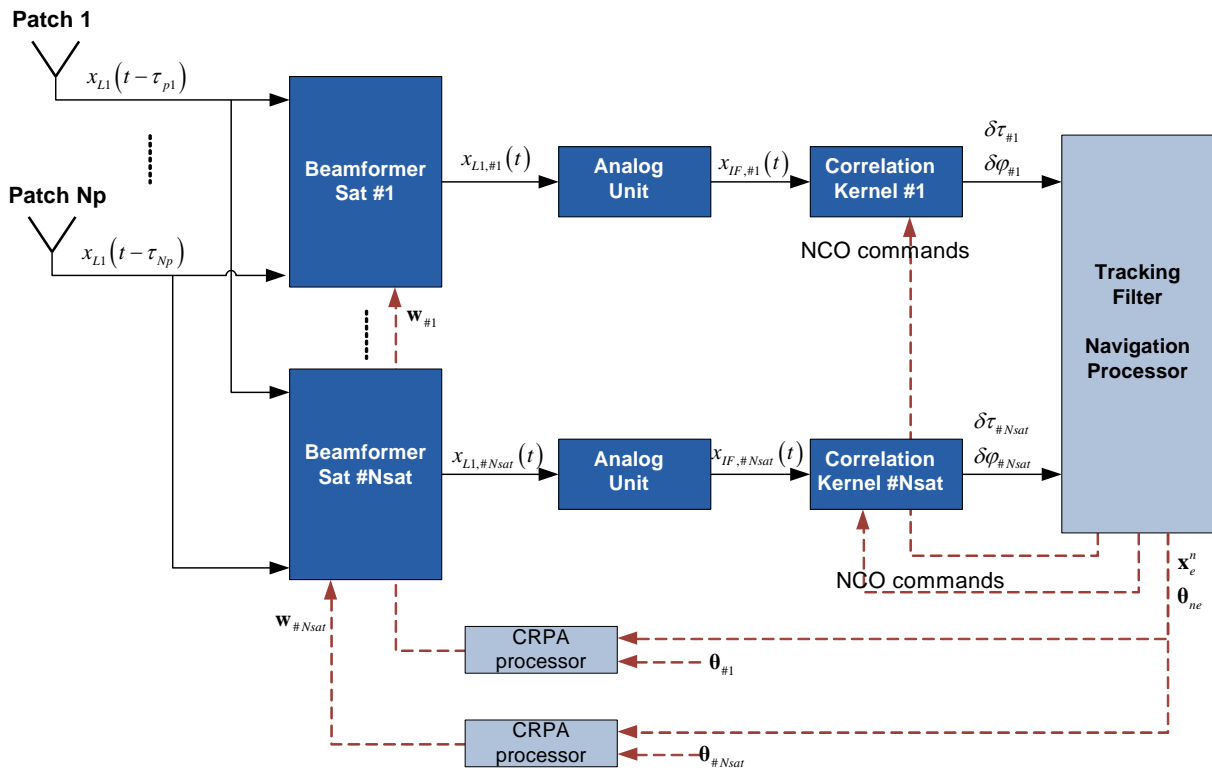
Beam forming for reception, differs fundamentally from beam forming for transmission.

The huge advantage of beam forming for receiving signals is, to calculate an optimal reception pattern for each reception direction, respectively satellite. There is no need to superimpose or heterodyne the different beam forming solutions, because for each satellite, an own signal processing channel is available. This is also the huge advantage of reception pattern forming, compared to transmission pattern forming.

In this section, this advantage shall be shown, by comparing the separate reception gains towards all satellites with a beam forming solution, being calculated similarly to the transmission pattern shaping.

V - 6.4.1 Virtual superimposed multi satellite reception pattern

In the following approach, each satellite respectively tracking channel has its own beam forming solution. The figure below shows the corresponding architecture.



V-90 Separate satellite beam forming in L1 band

Beam forming is realized within L1 band. For every satellite channel an own beam forming solution is calculated. For every single beam forming solution, an own analog unit with signal down conversion to an intermediate frequency is necessary.

The different beam forming solutions are not combined.

This is the huge advantage of reception pattern shaping in contrast to transmission pattern shaping, because in case of transmission pattern shaping there is only one solution, not a separate solution for each source, respectively target.

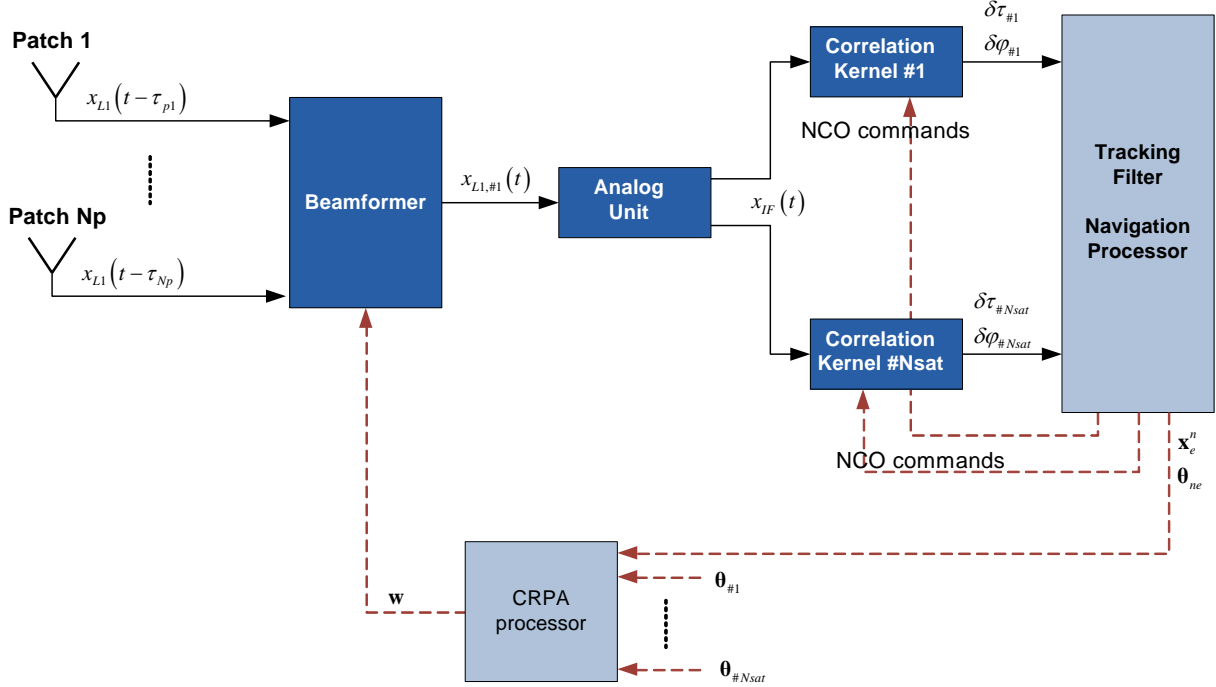
In figure V-92, it is assumed that 5 satellites are in view. For each of these 5 satellites, an own beam forming solution is calculated. Just to show the overall reception pattern, all independent reception patterns are plotted into one polar plot. This polar plot is a virtual superposition according to

$$B_{\Sigma}(\omega, \theta, \phi) = \max(B_1(\omega, \theta, \phi) \cdots B_{N_{sat}}(\omega, \theta, \phi)) \quad (V-69)$$

In signal processing, the different beam forming solutions are not superimposed, because every beam forming solution is processed in its own tracking channel.

V - 6.4.2 Beam forming like transmission pattern shaping

Instead of calculating an independent beam forming solution for every satellite, it is possible to calculate a single beam forming solution having more than one main lobe, steered towards the wanted satellites. The following figure gives the corresponding signal processing architecture.



V-91 Multibeam beam forming in L1 band

As can be seen from the given architecture, there is only one analog unit necessary, compared to figure V-90, where for every satellite an own analog unit is needed. The multi beam MVDR beam forming is therefore much more easier to implement from point of hardware design.

The derivation of the multibeam minimum variance distortion less response beam former is similar to the derivation given in equations (V-62) to (V-66). Instead of one, now more constraints are given.

$$L = \mathbf{w}^H \cdot \mathbf{S}_n(\omega) \cdot \mathbf{w} + (\mathbf{w}^H \cdot \mathbf{V} - \mathbf{1}^{1 \times N_{sat}}) \cdot \boldsymbol{\lambda} + \boldsymbol{\lambda}^H \cdot (\mathbf{V}^H \cdot \mathbf{w} - \mathbf{1}^{N_{sat} \times 1}) \quad (V-70)$$

The matrix \mathbf{V} contains the array manifold vectors towards the different satellite directions.

$$\mathbf{V} = [\mathbf{v}(\mathbf{k}_{\#1}, \mathbf{p}) \quad \dots \quad \mathbf{v}(\mathbf{k}_{\#N_{sat}}, \mathbf{p})] \quad (V-71)$$

In order to find the array weighting vector \mathbf{w} , the expression (V-70) has to be minimized.

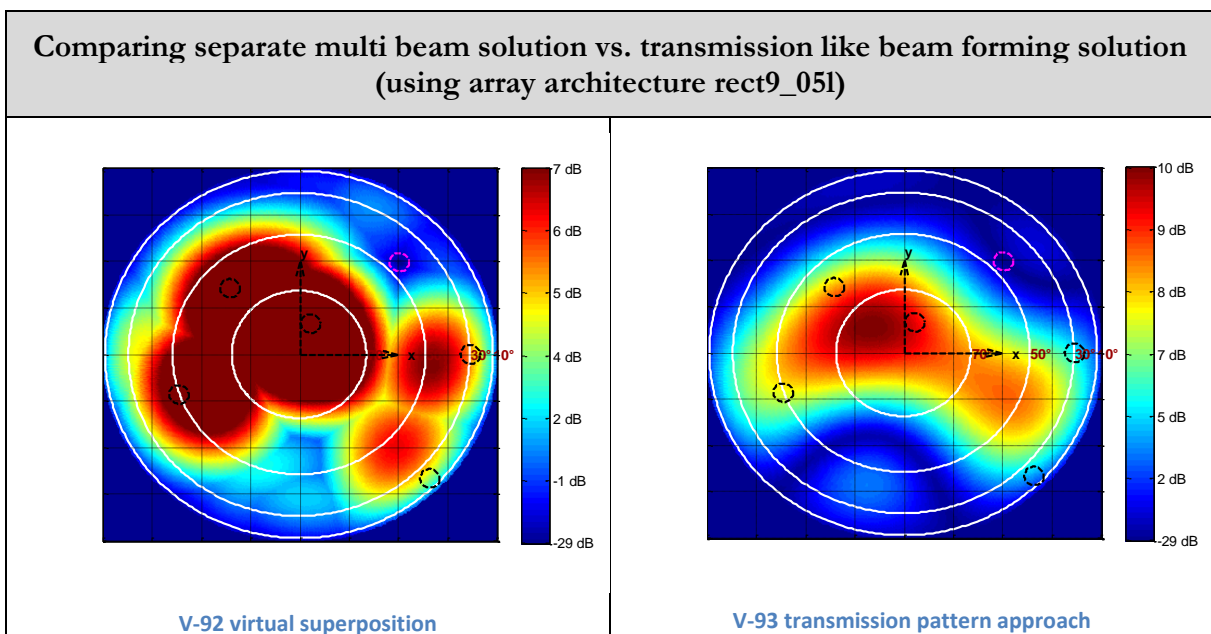
$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}^H} &= \mathbf{S}_n \cdot \mathbf{w} + \mathbf{V} \cdot \boldsymbol{\lambda} = \mathbf{0}^{N_{sat} \times 1} \\ \frac{\partial L}{\partial \boldsymbol{\lambda}^H} &= \mathbf{V}^H \cdot \mathbf{w} - \mathbf{1}^{N_{sat} \times 1} = \mathbf{0}^{N_{sat} \times 1} \end{aligned} \quad (V-72)$$

Equation (V-72) can be written as linear equation system and solved by using the pseudo inverse, or also numerically.

$$\begin{bmatrix} \mathbf{s}_n & \mathbf{V} \\ \mathbf{V}^H & \mathbf{0}^{N_{\text{sat}} \times N_{\text{sat}}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{N_{\text{patch}} \times 1} \\ \mathbf{1}^{N_{\text{sat}} \times 1} \end{bmatrix} \quad (\text{V-73})$$

(This approach can also be used for nulling, which will be discussed in a later section. In case of nulling, fictive satellites are used in order to maximize the antenna gain over a wide part of the hemisphere. The jammer direction is well known within the antenna coordinate system, even if only nulling is applied.)

In order to compare the separate beam forming processing for each satellite and the above derived, multi satellite beamforming solution, an example with 5 satellites is used. For the following example, the real single patch, non-uniform reception pattern is used.



From the figures above it gets obvious, that calculating an independent beam forming solution for every satellite, gives the best results with the already mentioned disadvantage of a higher processing load.

The alternative approach, calculating only one beam forming solution, having main lobes to all wanted satellites, suffers an important drawback. The number of main lobes and therefore the number of satellites which can be considered is limited by the number of array elements.

$$N_{\text{Sat,Max}} = N_{\text{patch}} - N_{\text{jammer}} - 1 \quad (\text{V-74})$$

V - 6.5 Performance of different nulling algorithms

V - 6.5.1 Introduction

In the previous sections on beam forming, the jammer and satellite position within the antenna coordinate system were necessary.

The problem of beam forming is, it is not applicable as a standalone solution for example within the antenna electronics. A special GPS receiver is needed for realization.

In this section, nulling will be introduced. The huge advantage of nulling is that this approach minimizes the reception pattern gain towards the jammer direction, without the need of information about the receiver position or the antenna attitude.

The huge disadvantage of nulling is the unwanted satellite damping. The reception pattern is calculated only with the goal to minimize the gain towards the estimated line of sight to the jammer in the local antenna coordinate system. The reception pattern is not optimized towards high gain in satellite direction.

In the literature like in [119], [120], [121] and [107], mainly two basic Nulling approaches are discussed, which will be introduced in the following.

Additionally, a novel method for nulling will be developed, based on superimposed MVDR beam forming.

The main goal of this section is to figure out, to what extent nulling is comparable to beam forming. Moreover, the different nulling approaches will be compared and the optimal array geometry for the different nulling approaches will be identified.

V - 6.5.2 Nulling with a reference element

The first nulling approach being discussed in this section, is based on the idea of using a reference element in the antenna array. The array weight for the defined reference element is set to 1. The weights for all other array elements are calculated in that way to get a maximum cancellation towards the assumed jammer direction.

The reception gain pattern can be written as

$$B(\omega, \theta, \phi) = B(\omega, \mathbf{k}) = \mathbf{w}^T \cdot \begin{bmatrix} e^{-j\mathbf{k}^T \cdot \mathbf{p}_1} \\ \vdots \\ e^{-j\mathbf{k}^T \cdot \mathbf{p}_{N_p}} \end{bmatrix} \cdot \mathcal{X}(\omega) = \mathbf{w}^T \cdot \mathbf{v}_k(\mathbf{k}) \cdot \mathcal{X}(\omega) \quad (\text{V-75})$$

The amplitude of the received signal, respectively the received power spectral density $\mathcal{X}^2(\omega)$ is equal at every element. The received signals at the different elements differ only within their relative phase relationship.

In order to get a “Null” at a defined direction, the beam weights \mathbf{W} must be calculated to get the following result.

$$B(\omega, \mathbf{k}) = \mathbf{w}^T \cdot \begin{bmatrix} e^{-j\mathbf{k}^T \cdot \mathbf{p}_1} \\ \vdots \\ e^{-j\mathbf{k}^T \cdot \mathbf{p}_{N_p}} \end{bmatrix} \cdot \mathcal{X}(\omega) = 0 \quad (\text{V-76})$$

The trivial solution for equation (V-76) would be

$$\mathbf{w} = \mathbf{0} \quad (\text{V-77})$$

to get a jammer damping as good as possible. But $\mathbf{w} = \mathbf{0}$ would not only reject jamming sources, but also the satellite signal itself.

Without any side condition, like maximizing the reception gain towards the satellite direction, the best solution for a minimum gain towards the jamming source, is a zero-weighting vector \mathbf{w} .

Therefore, the challenge in case of completely independent nulling is, to find a weight vector, which minimizes the reception gain towards the jamming source, without being zero. Moreover, the task is to find an array weighting vector, which minimizes the reception pattern towards the jammer direction and maximizes the array gain towards all other directions on the hemisphere.

The reference signal is defined as

$$y_{ref}(t) = \hat{\mathbf{a}}_{ref}(t) \cdot e^{-j(\omega t + \varphi_{ref}(t))} \quad (\text{V-78})$$

All other patch elements are the auxiliary elements. The received signals from all these auxiliary elements are added using complex weighting coefficients

$$\mathbf{w}^T = \left[a_{w,1} \cdot e^{-j\varphi_{w,1}} \quad \dots \quad a_{w,N_{patch}-1} \cdot e^{-j\varphi_{w,N_{patch}-1}} \right] \quad (\text{V-79})$$

The signal from an auxiliary element is defined according to the next equation.

$$y_{aux,n}(t) = \hat{\mathbf{a}}_{aux,n}(t) \cdot e^{-j(\omega t + \varphi_{aux,n}(t))} \quad (\text{V-80})$$

The superposition of all auxiliary elements can be written as

$$\begin{aligned} y_{aux,\Sigma}(t) &= \hat{\mathbf{a}}_{aux,\Sigma}(t) \cdot e^{-j(\omega t + \varphi_{aux,\Sigma}(t))} = \sum_{n=1}^{N_{patch}-1} w_n \cdot \hat{\mathbf{a}}_{aux,n}(t) \cdot e^{-j(\omega t + \varphi_{aux,n}(t))} \\ y_{aux,\Sigma}(t) &= \sum_{n=1}^{N_{patch}-1} a_{w,n}(t) \cdot e^{-j\varphi_{w,n}(t)} \cdot \hat{\mathbf{a}}_{aux,n}(t) \cdot e^{-j(\omega t + \varphi_{aux,n}(t))} \end{aligned} \quad (\text{V-81})$$

The goal is to find complex weighting coefficients w_n , which give a summarized signal $y_{aux,\Sigma}(t)$ being 180° phase shifted in relation to the received signal from the reference element. In the following, the phase and amplitude conditions for the weighting coefficients for a N patch array antenna are derived.

The direction of jamming signal arrival, respectively the corresponding wave vector, is given as

$$\mathbf{k}_J = \frac{2 \cdot \pi}{\lambda_{L1}} \cdot \begin{bmatrix} -\cos(\theta_J) \cdot \cos(\phi_J) \\ -\cos(\theta_J) \cdot \sin(\phi_J) \\ -\sin(\theta_J) \end{bmatrix}. \quad (V-82)$$

In order to get a “Null”, respectively minimum reception gain towards the jammer direction, amplitude and a phase condition must be fulfilled.

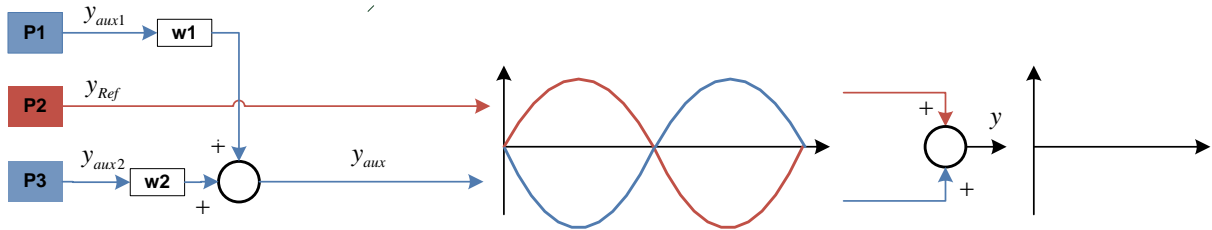
The necessary phase relationship between the combined auxiliary signals and the reference signal is given below.

$$\begin{aligned} & \left(\sum_{n=1}^{N_{patch}-1} a_{w,n} \cdot e^{-j\varphi_{w,n}} \cdot \hat{a}_{aux,n} \cdot e^{-j(\omega t + \varphi_{aux,n})} \right) \\ & = \left(\sum_{n=1}^{N_{patch}-1} a_{w,n} \cdot e^{-j\varphi_{w,n}} \cdot \hat{a}_{aux,n} \cdot e^{-j\omega t} \cdot e^{-j\mathbf{k}_J^T \cdot (\mathbf{p}_n - \mathbf{p}_{ref})} \right) = \left(a_{ref} \cdot e^{-j\omega t} \cdot e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_{ref}} \cdot e^{-j\pi} \right) \end{aligned} \quad (V-83)$$

As reference element $\mathbf{p}_{ref} = \mathbf{p}_{Npatch}$ is defined. All received signals from other patches are related to this reference element and phase. The phase of reference element is defined as zero without any restriction.

$$\varphi_{ref}(t) = \mathbf{k}_J^T(t) \cdot \mathbf{p}_{ref} = 0 \quad (V-84)$$

All auxiliary phases are refer to the phase of the reference element.



V-94 Phased array with three patches and reference element

$$\text{phase condition: } \sum_{n=1}^{N_{patch}-1} \varphi_{w,n} + \mathbf{k}_J^T \cdot (\mathbf{p}_n) = \pi \quad (V-85)$$

$$\text{amplitude condition: } \sum_{n=1}^{N_{patch}-1} a_{w,n} \cdot \hat{a}_{aux,n} = \hat{a}_{ref} \quad (V-86)$$

Equation (V-85) shows that the degree of freedom is $N_{patch} - 1$, which in turn gives the possibility to realize $N_{patch} - 1$ minima within the reception pattern.

Instead of solving the phase and amplitude condition separately, it is much easier to demand the

absolute value of the following equation to be zero.

$$\left| V_{ref} - \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \right|^2 = \epsilon, \quad \mathbf{v}_{k,aux}^T = \left[e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_1} \quad \dots \quad e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_{Npatch-1}} \right] \quad (V-87)$$

$$V_{ref} = e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_{ref}} \quad (V-88)$$

In order to find a weighting vector \mathbf{w} minimizing ϵ , the least squares approach will be applied.

$$\begin{aligned} \left| V_{ref} - \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \right|^2 &= \left(V_{ref} - \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \right)^T \cdot \left(V_{ref} - \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \right) \\ &= V_{ref}^2 - V_{ref} \cdot \mathbf{w}^T \cdot \mathbf{v}_{k,aux} - \mathbf{v}_{k,aux}^T \cdot \mathbf{w} + \mathbf{v}_{k,aux}^T \cdot \mathbf{w} \cdot \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \end{aligned} \quad (V-89)$$

$$\frac{\partial}{\partial \mathbf{w}^T} \left| V_{ref} - \mathbf{w}^T \cdot \mathbf{v}_{k,aux} \right|^2 = 0 \quad (V-90)$$

$$\mathbf{w} = - \left(\mathbf{v}_{k,aux} \cdot \mathbf{v}_{k,aux}^T \right)^{-1} \cdot \mathbf{v}_{k,aux} \cdot V_{ref} \quad (V-91)$$

This nulling method assumes that an estimation of the jamming line of sight direction respectively the jammer position within the antenna coordinate system is available. This can be realized within the antenna itself, by setting up the noise covariance matrix between all array elements. This matrix contains the jammer direction.

V - 6.5.3 Gram-Schmidt orthogonalization based nulling

This nulling approach introduced in this section is based on the Gram-Schmidt orthogonalization. Similar nulling approaches are given in [122], [108] and [123]. In general, Gram-Schmidt orthogonalization is given in [124].

Starting point is an arbitrary array weighting vector $\mathbf{w} \neq \mathbf{0}$, which could be for example

$$\mathbf{w}^T = [1 \quad \dots \quad 0] \quad (V-92)$$

or the weighting vector resulting from pure beam steering, to become a main lobe towards a wanted satellite.

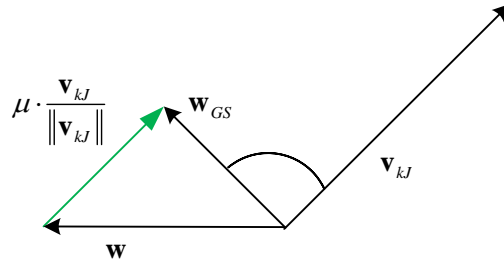
The array manifold vector towards the jammer direction, where the reception pattern shall be minimized, is given as

$$\mathbf{v}_{kJ}^T = \left[e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_1} \quad \dots \quad e^{-j\mathbf{k}_J^T \cdot \mathbf{p}_{Npatch-1}} \right]. \quad (V-93)$$

The goal is to find an array weighting vector $\mathbf{w}_{GS} \neq \mathbf{0}$, for which holds

$$\mathbf{w}_{GS}^T \mathbf{v}_{kJ} = 0 \quad (V-94)$$

Equation (V-94) can also be represented in vector form, as the following figure shows.



V-95 Gram-Schmidt orthogonalization

Starting with the arbitrary array weighting vector \mathbf{w} , the difference vector $\delta\mathbf{w}$, which is needed to get the vector \mathbf{w}_{GS} , is calculated according to following equation [108, p. 39].

$$\mathbf{w}_{GS} = \mathbf{w} + \delta\mathbf{w} = \mathbf{w} + \mu \cdot \frac{\mathbf{v}_{kJ}}{\|\mathbf{v}_{kJ}\|} \quad (\text{V-95})$$

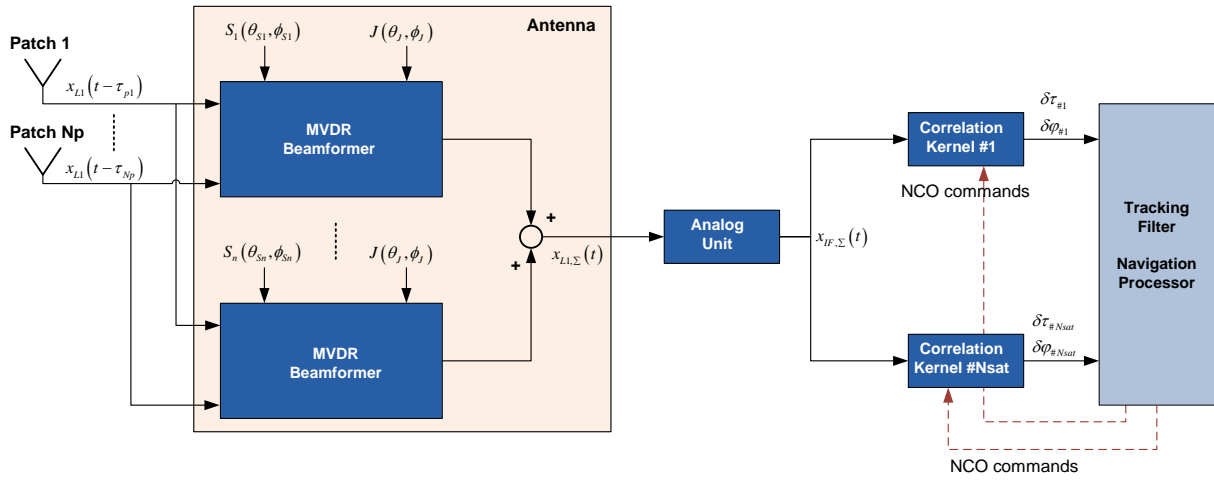
$$\mathbf{w}_{GS} = \left(\mathbf{I}^{N_{patch} \times N_{patch}} - \frac{\frac{\mathbf{v}_{kJ}}{\|\mathbf{v}_{kJ}\|} \cdot \mathbf{v}_{kJ}^T}{\mathbf{v}_{kJ}^T \cdot \frac{\mathbf{v}_{kJ}}{\|\mathbf{v}_{kJ}\|}} \right) \cdot \mathbf{w} \quad (\text{V-96})$$

V - 6.5.4 Heterodyned MVDR beam forming with synthetic satellites

As already mentioned, the advantage of nulling is, it can be realized completely within the antenna and is therefore applicable to all GPS receivers without any need to modify the receiver itself. This measure does not need any information about the antenna attitude or the receiver position. The disadvantage of nulling is the damping of the satellite signal and therefore deterioration of the signal to noise ratio.

Beamforming however needs the attitude information of the antenna and also the satellite position. Beamforming is therefore only possible, if the GPS receiver provides this information.

A novel approach will be introduced in this section. Here, the advantage of nulling, working without any knowledge about the actual antenna attitude, together with the advantage of beamforming, causing almost no unwanted degradation of the satellite signal, will be combined.



V-96 Signal processing architecture – heterodyned MVDR beam forming with synthetic satellites

The figure above shows the principle architecture of this novel approach.

For this method, a set of fictive satellites is defined within the local antenna coordinate system. The angles θ and ϕ represent the attitude of the jammer and satellite in the local antenna coordinate system. The fictive satellites are evenly distributed about the antenna hemisphere using a geodetic grid. The number of fictive satellites is not limited by the array topology respectively number of array patterns, because for every satellite an own MVDR beamforming solution is calculated.

The only limit in high gain hemisphere coverage is the processing capability of the antenna, because for every fictive satellite, an own beam forming solution must be calculated.

The MVDR beam forming solution is calculated with the already introduced equation

$$\mathbf{w} = (\mathbf{v}^H \cdot \mathbf{S}_n^{-1} \cdot \mathbf{v})^{-1} \cdot \mathbf{v}^H \cdot \mathbf{S}_n^{-1} \quad (\text{V-97})$$

whereby the jammer direction is a direct result of the measured noise covariance matrix \mathbf{S}_n , which can be realized also within antenna electronic.

The output signals of all these independent beamforming solutions are superimposed afterwards within the antenna electronics, to create a heterodyned reception pattern. The receiver gets at its input only one cable with the already heterodyned solution.

All these measures can be implemented within the antenna electronics, with no need to communicate with the GPS receiver.

In case of normal beam forming, the satellite and jammer position are known exactly within the antenna coordinate system and within the platform coordinate system. Here, for every satellite, a separate beam forming solution is calculated and used in the corresponding tracking channel. No real superposition of the different beam forming solutions is necessary.

For the approach discussed here, the situation is different. All beam forming solutions for the different fictive satellites need to be superimposed to get one combined reception pattern.

The superposition is realized according to the following equation. In order to get the heterodyned

reception gain, again the frequency space representation is used.

$$B_{\Sigma}(\omega, \theta, \phi) = \sum_n |B_n(\omega, \theta, \phi)| \cdot e^{j(\omega t + \varphi_n(\theta, \phi))} \quad (V-98)$$

A disadvantage of this approach is the possibility of signal cancellation instead of superposition in certain directions. This effect becomes obvious by considering an example of only two separate beam forming solutions. Assuming a phase difference $\varphi_n(\theta, \phi)$ of 180 degrees between the two independent MVDR beam forming solutions in a certain line of sight direction, the resulting reception pattern gain would be zero in the corresponding direction, if both gain values are equal.

The resulting heterodyned signal in time space can be written as

$$\mathcal{F}^{-1}\{B_{\Sigma}(\omega, \theta, \phi)\} = x_{\Sigma}(t, \theta, \phi) = |x_{\Sigma}(t, \theta, \phi)| \cdot e^{j(\omega t + \varphi_{\Sigma}(\theta, \phi))} = a_{\Sigma}(t, \theta, \phi) \cdot x_{L1}(t - \tau_{\Sigma}(\theta, \phi)) \quad (V-99)$$

The factor $a_{\Sigma}(t, \theta, \phi)$ affects the amplitude of the superimposed signal.

In the analysis used for comparing beam forming and nulling, 16 fictive satellites are used which are evenly distributed over the hemisphere on a geodetic grid.

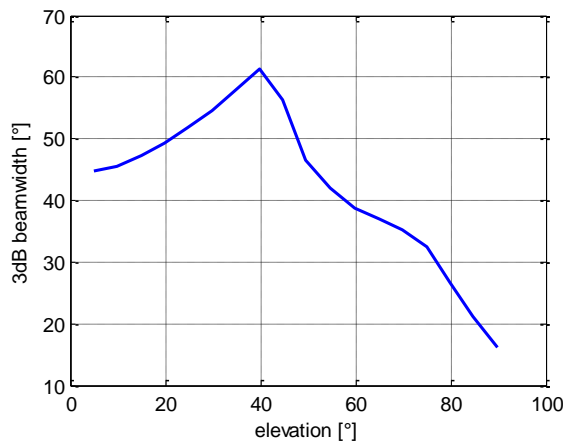
V - 6.5.4.1 Necessary beam forming solutions for complete hemisphere coverage

In the previous section, nulling was realized by superimposing many beam-forming solutions, having main lobes to different fictive satellites. All beam forming solutions have the same null towards the identified jammer.

The solution becomes better, the more main lobes respectively fictive satellites are used. In this section the question will be answered, how many fictive satellites respectively beam forming solutions are necessary to get complete hemisphere coverage.

For calculation it is assumed, that a 3dB beam width of each single main lobe, provides a good coverage.

The following figure shows the 3dB beam width of a single MVDR solution, based on a rectangular array having 9 antenna elements.



V-97 3dB beam width depending on elevation

The complete surface of the hemisphere with a radius of 1 is given by

$$A = 4 \cdot \pi \cdot r^2 = 4 \cdot \pi \quad (\text{V-100})$$

A spherical segment, covered by the 3dB beam width is given by

$$A_{3dB} = 4 \cdot \pi \cdot \sin^2 \left(\frac{\theta_{3dB}}{2} \right) \quad (\text{V-101})$$

Assuming $\theta_{3dB} = 17^\circ$, the total number of needed MVDR beam forming solutions is at least 23 for covering the hemisphere, in case of evenly distributed fictive satellites.

But even in case of 23 or much more fictive satellites, there may be areas on the hemisphere with low gain, due to subtractive superposition.

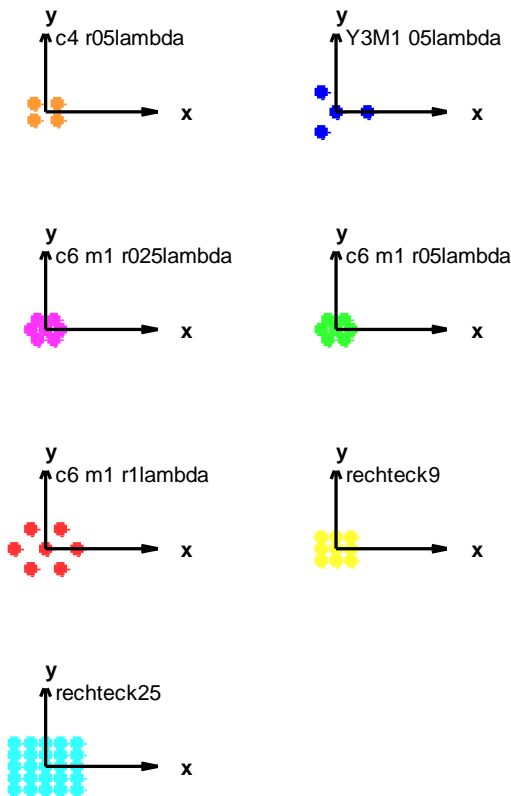
V - 6.5.5 Comparing different nulling algorithms – assuming uniform element reception pattern

In previous sections, three different nulling algorithms were introduced and developed. In this section, the nulling algorithm, providing the best jammer damping and simultaneously minimum satellite damping, will be identified. Additionally, the optimal array geometry for the different nulling algorithms is determined.

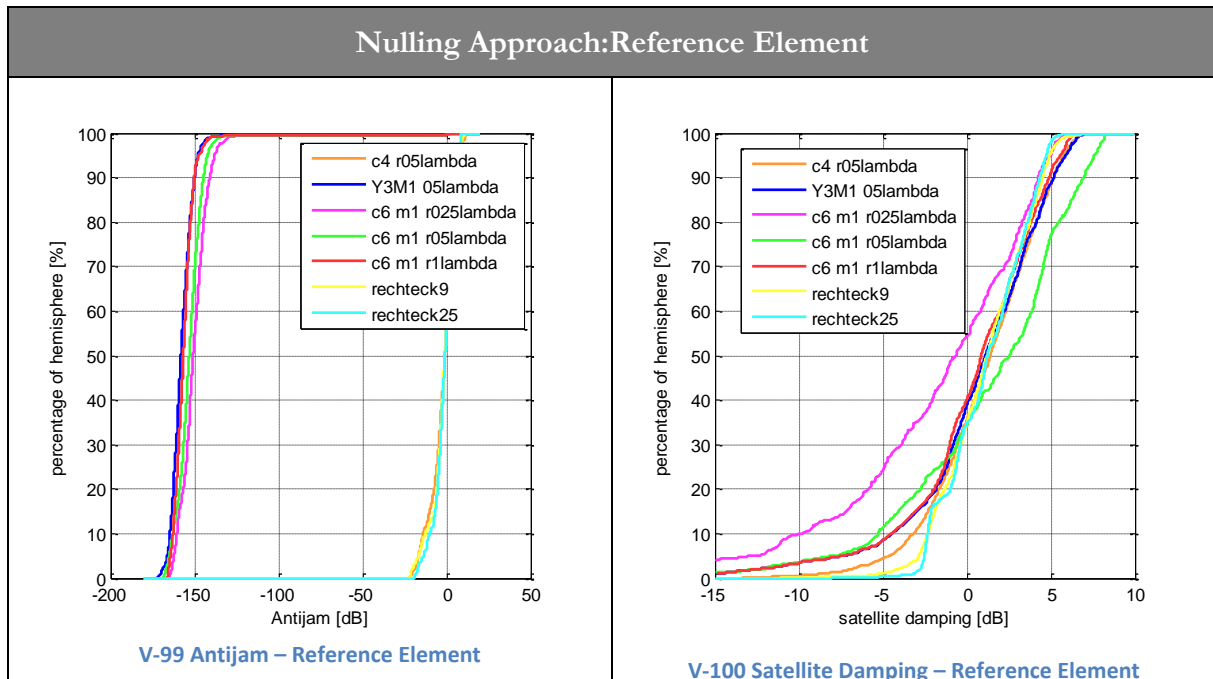
This analysis is only done with uniform element reception pattern. The best performing algorithm and geometry in case of uniform element reception pattern, would be also the best performing algorithm, if non uniform element reception pattern are used.

For each array geometry and nulling algorithm, a fictive jammer is moved across the hemisphere. In order to get good hemisphere coverage, the jammer is moved on a geodetic grid across the hemisphere.

The following figure shows the architectures, used for nulling analysis.



V-98 Relation: Color <-> Array Type

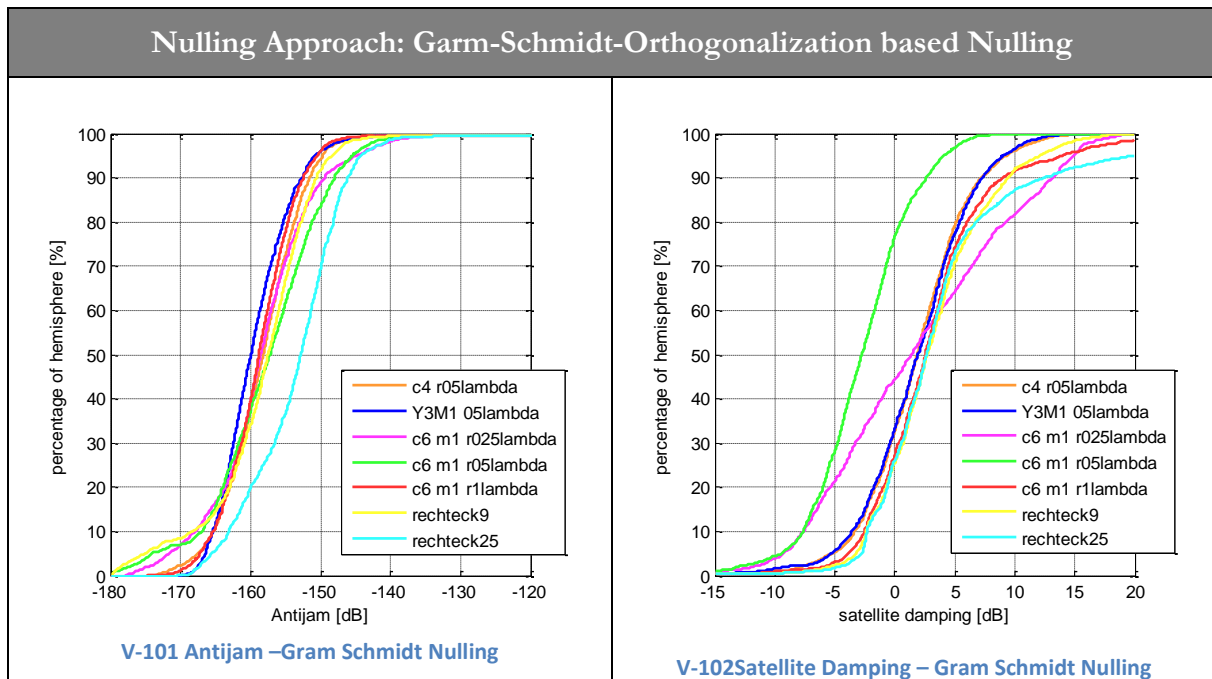


The figures above show, which percentage of the hemisphere has an antijam respectively satellite gain, lower than the given value on the x axis.

The antijam evaluation shows, that antenna arrays having different radii to the elements from the reference element, are not suitable for the algorithm which uses a reference element.

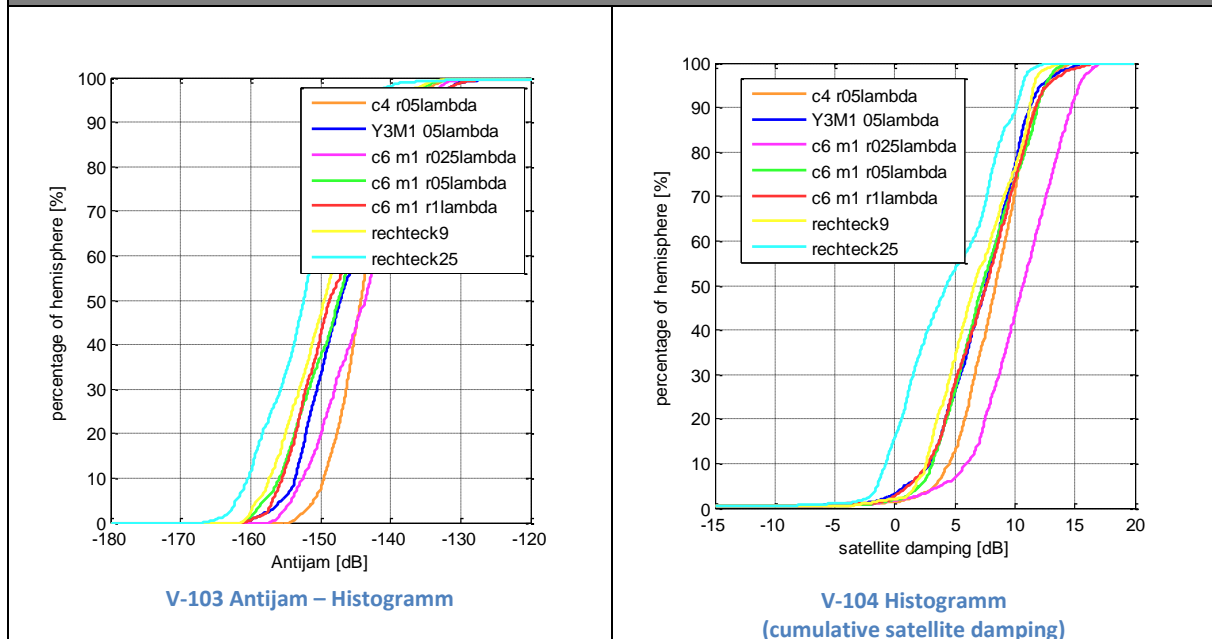
Besides the jammer damping, it is very important to consider the unwanted satellite damping. An unwanted satellite damping of more than 10dB may already be critical for loss of lock in bad signal to noise environments.

One of the best performing array geometries is the circular “c6m1_05lambda” architecture.



Using Gram-Schmidt orthogonalization based nulling, all array architectures show a suitable antijam and at the same time, an acceptable satellite damping. For further comparison with other approaches, the architecture “Y3M1” will be used, because this architecture provides a high antijam and also a low satellite damping. For some parts of the hemisphere, there is even some satellite amplification.

Nulling Approach: Heterodyned beamforming with synthetic satellites



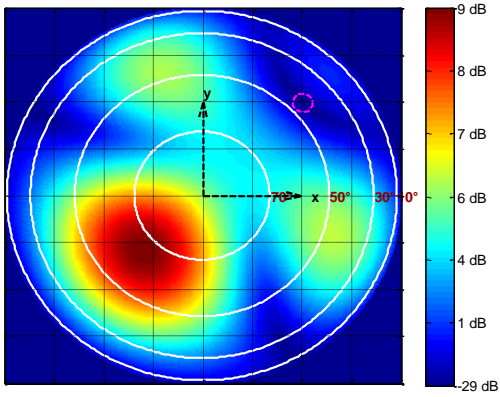
Using the novel nulling approach, which superimposes the beam forming solution to many fictive satellites, the best performing array geometry regarding antijam would be the rectangular shaped array, having 25 elements. As satellite damping is the more important criteria, the best performing array geometry having acceptable antijam and at the same time a small satellite damping is the Y3M1_05lambda.

For nulling in general, array geometries with less array elements provide better results than array geometries having many patches. For beam forming, it's exactly the opposite.

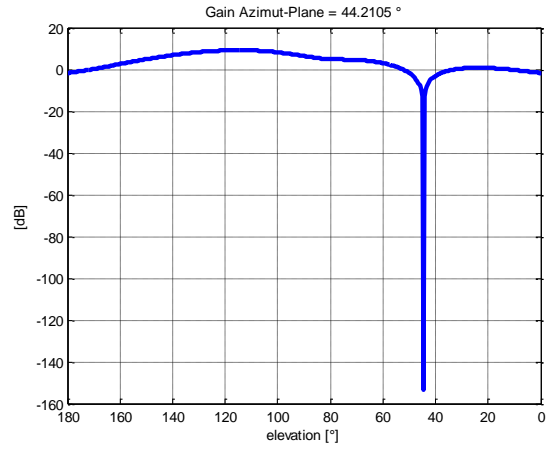
For nulling it is important that the lobes of the reception pattern are very broad. Thus, a large area of hemisphere is covered by broad reception pattern with high gain, which is important, because the satellite position in the antenna coordinate system is not known. By realizing a wide area of the hemisphere with high gain, there is a high probability that the real satellite is within this high gain part.

In case of beam forming, the satellite position in the antenna coordinate system is known. The main lobe is exactly steered towards the satellite. It is advantageous that all parts of the hemisphere with no satellites have a reception gain as small as possible, in order to prevent also undetected jammers from entering the signal processing chain.

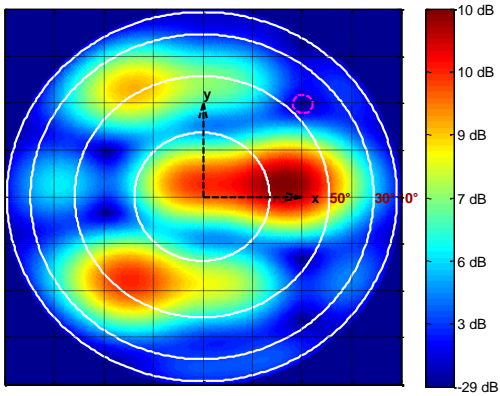
In order to get an impression of the reception pattern shape in case of nulling, the following figures show the polar reception pattern of the Gram-Schmidt-Orthogonalization based nulling approach for different array geometries.



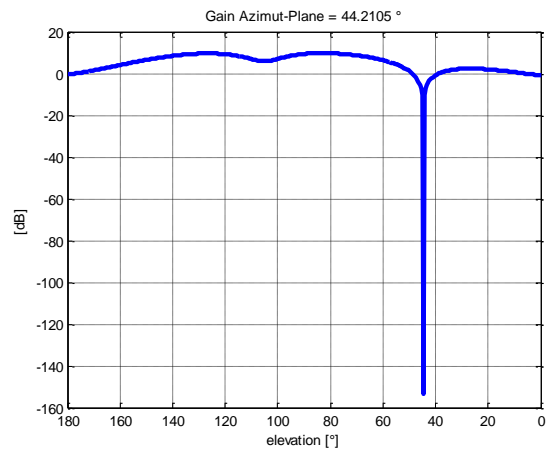
V-105 Polar – c4_05lambda



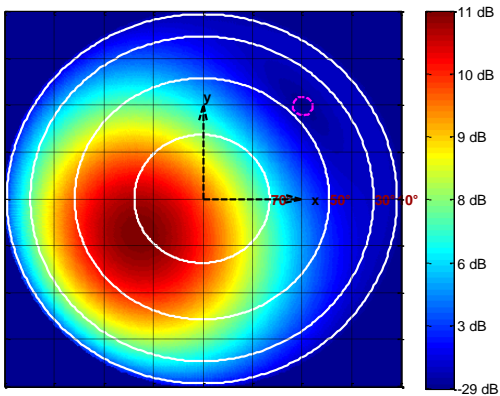
V-106Sectional Plane – c4_05lambda



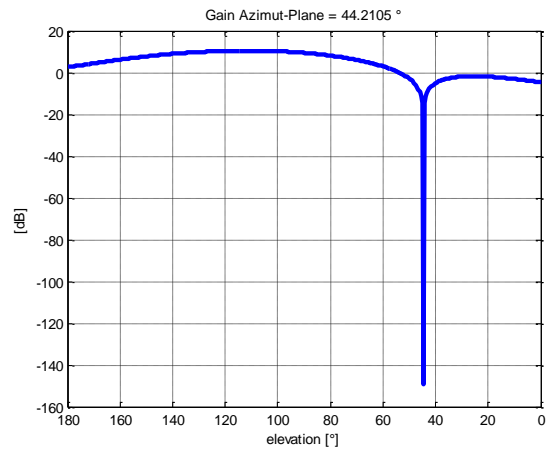
V-107 Polar –Y3M1_05lambda



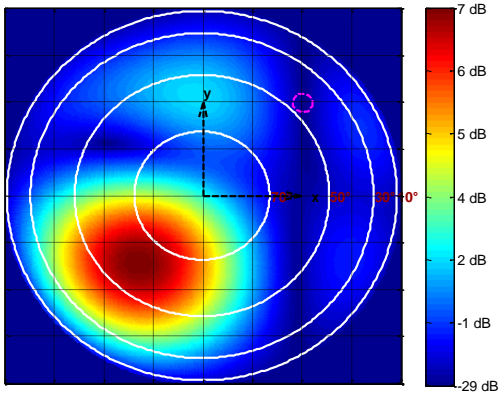
V-108Sectional Plane –Y3M1_05lambda



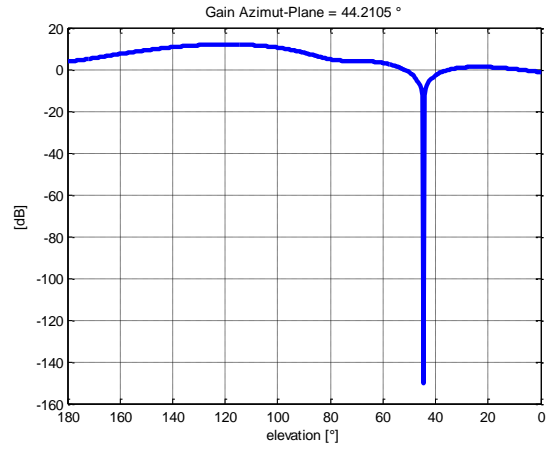
V-109 Polar – c6m1_025lambda



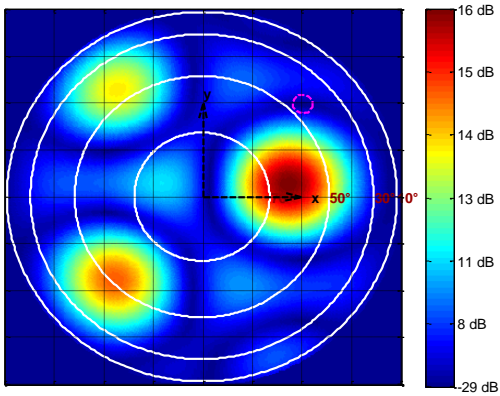
V-110Sectional Plane – c6m1_025lambda



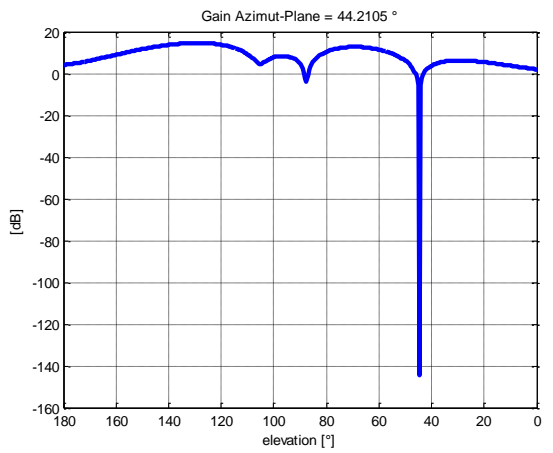
V-111 Polar – c6m1_05lambda



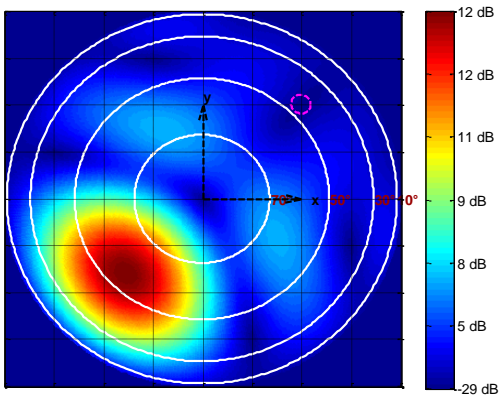
V-112Sectional Plane – c4_05lambda



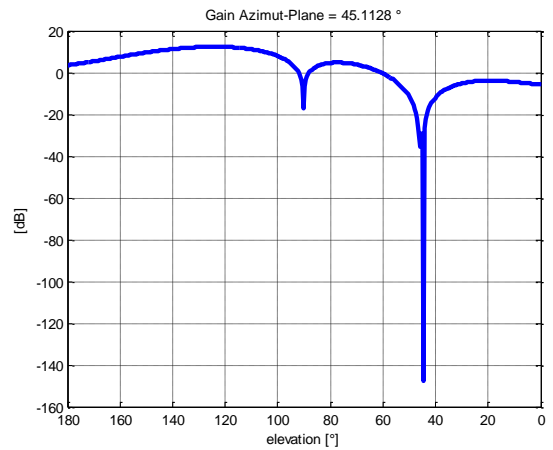
V-113 Polar – c6m1_1lambda



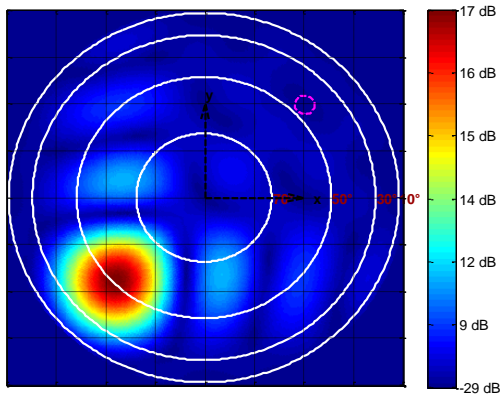
V-114Sectional Plane – c6m1_1lambda



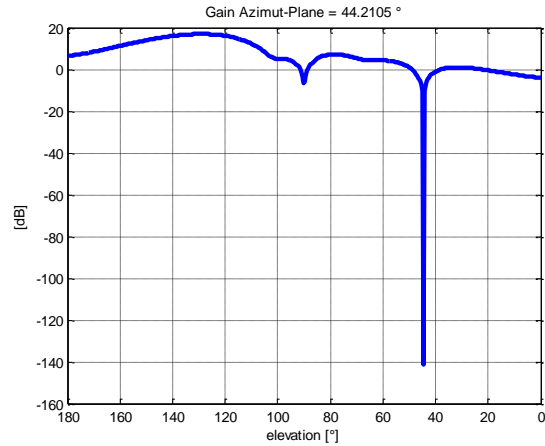
V-115 Polar – rectangle9



V-116Sectional Plane – rectangle9



V-117 Polar – rectangle25



V-118Sectional Plane – rectangle25

The polar plots above show, that the null is very steep. Small attitude errors of the estimated jammer direction would worsen the provided nulling capability tremendously.

All array geometries provide a very deep jammer damping of up to more than 140 dB. It is important to keep in mind, that this is just a theoretical value, which can only be achieved in case of perfectly known array geometry and perfectly known attitudes of the jammer. In case of attitude errors, the provided antijam becomes worse due to the steep null characteristic. Also, in case of unknown array geometry errors, the provided antijam becomes worse. These effects will be analyzed in a following section.

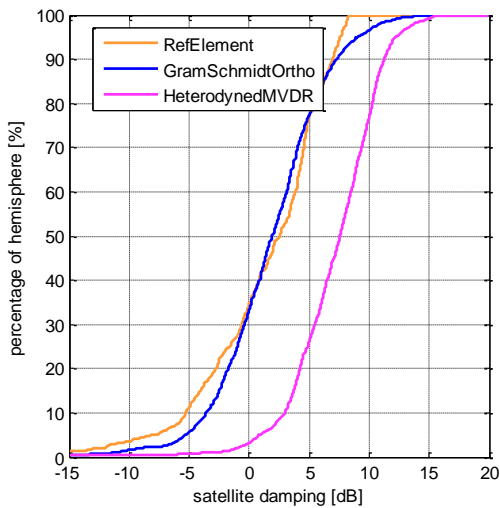
As preparation for the following section on comparing nulling and beam forming, in this section the best nulling algorithm together with the optimal array geometry shall be determined.

Up to now, three nulling algorithms were evaluated, where for each algorithm, also different array types were used.

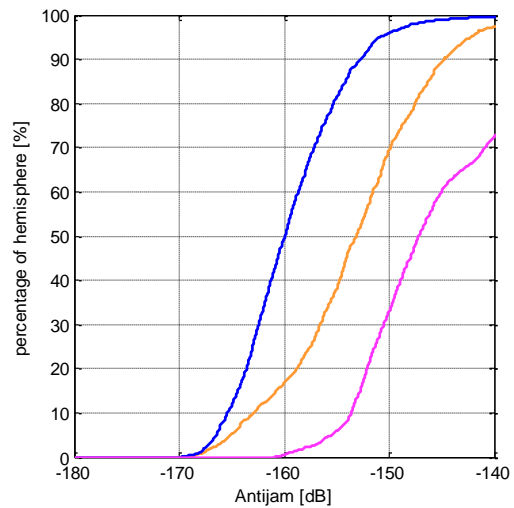
The following figure compares the performance of the three introduced nulling algorithms, using for each the best performing array geometry.

The following list shows these combinations.

- Nulling with reference element: c6m1_05lambda
- Nulling with Gram-Schmidt orthogonalization: Y3M1
- Heterodyned Nulling: Y3M1



V-119 Best of Nulling – SatGainDamping



V-120 Best of Nulling – Antijam

The evaluations above show that the MVDR based nulling, using fictive satellites, provides with respect to unwanted satellite damping, definitely the best results. The provided antijam is worse compared to the other algorithms, but still high enough. The vital criteria for tracking robustness is in this case the unwanted satellite damping, in order to enable tracking also in low signal to noise environments.

The huge disadvantage of MVDR based nulling with fictive satellites, is the high processing load. If for example 15 fictive satellites would be used, in parallel, 15 MVDR beam forming solutions must be calculated in real time within the receiver.

Nulling with Gram-Schmidt based calculation of the array weighting vector, provides also a very good performance. The satellite gain is lower than the MVDR based nulling approach, but the antijam is much higher at a much lower processing demand.

In the following, the Gram-Schmidt based nulling algorithm will be used for further evaluations.

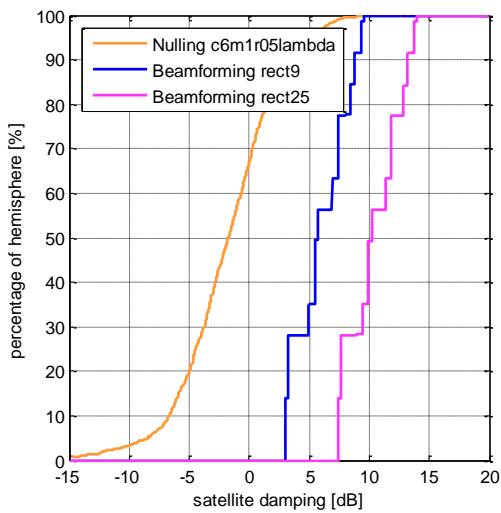
V - 6.6 Comparing beam forming and nulling

In the following comparison, the two best nulling algorithms will be compared with beam forming. For nulling, the following algorithm – array combinations will be used.

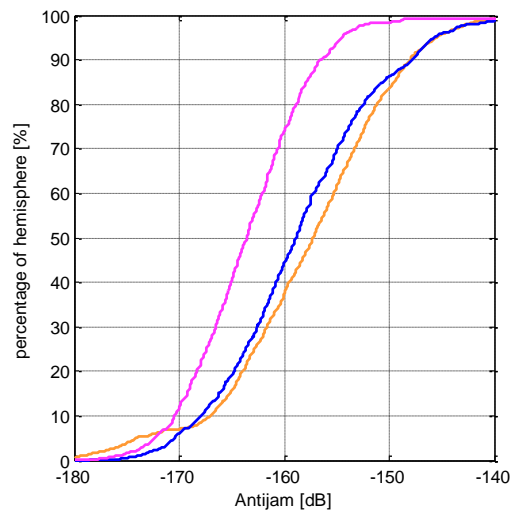
- Gram-Schmidt orthogonalization based nulling – Y3M1
- MVDR beam forming with fictive satellites – c6m1_05lambda

For beam forming, the “rectangle9” architecture and also the better “rectangle25” architecture will be used. For both, nulling and beam forming, real non uniform element reception pattern are used.

The following figures show at first the Gram-Schmidt orthogonalization based nulling.

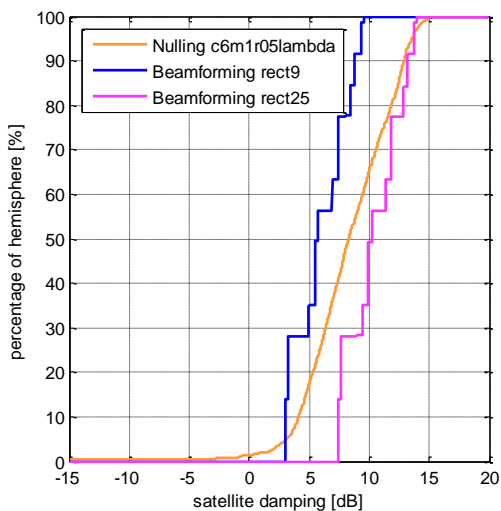


V-121 Gram-Schmidt nulling vs. beam forming

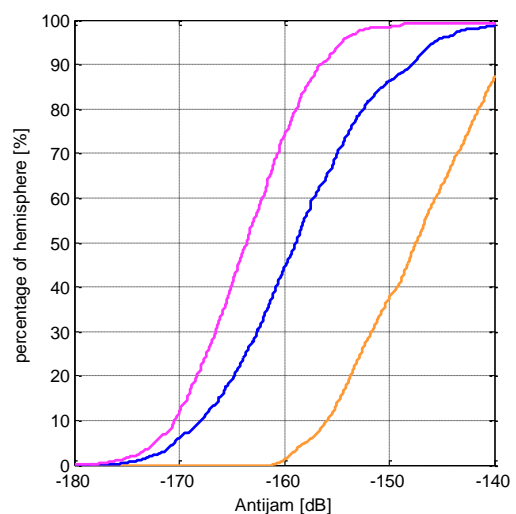


V-122 Gram-Schmidt nulling vs. beam forming

The simulations above show that beam forming has a gain higher than zero towards the satellites, whereas nulling worsens the received satellite signal. Regarding antijam, also the beam forming approach provides better results.



V-123 SatGainDamping –Nulling method: Heterodyned mvdr



V-124 Antijam –Nulling method: Heterodyned mvdr

MVDR based beam forming with fictive satellites is comparable to beam forming regarding unwanted satellite damping. The provided antijam is worse, but still high enough.

V - 6.7 Comparing beam forming and nulling in case of more than one jammer

In the previous sections, the optimal array geometry for MVDR beam forming was identified, which is the rectangular array with 9 elements. (Apart from the rectangular array with 25 elements, being not practicable for small flying platforms)

Also, the most promising nulling algorithm together with best fitting array geometry was identified, which is the orthogonalization based nulling together with the “c6m1_05lambda” array geometry.

(The MVDR based beam forming with fictive satellites will not be used for further evaluation, because at the time being, this approach is not practicable due to its high processing load)

Up to now, only one jammer was considered. In this section, the performance of beam forming and nulling in case of two and three jammers will be evaluated. A analytical analysis of nulling performance, given more than one jammer is given in [125].

MVDR beam forming with more than one jammer

In equation (V-56), the noise covariance matrix was introduced, which is the central element of information about the jammer within the MVDR beam forming algorithm.

If more than one jammer is received, the received noise for all array elements can be combined into one vector.

$$\mathbf{x}_{noise} = \mathbf{n} + \mathbf{j}_1 + \mathbf{j}_2 = \begin{bmatrix} n_1 \\ \vdots \\ n_{Npatch} \end{bmatrix} + \begin{bmatrix} j_{1,1} \\ \vdots \\ j_{1,Npatch} \end{bmatrix} + \begin{bmatrix} j_{2,1} \\ \vdots \\ j_{2,Npatch} \end{bmatrix} \quad (V-102)$$

For calculating the noise covariance matrix, at first it is assumed, that the different jamming signals are uncorrelated. Also the noise components from the different antenna patches are assumed to be uncorrelated.

$$\begin{aligned} \mathbf{C}_{corr,Noise}(\tau) = & \frac{1}{T_{corr}} \cdot \int_{T_{corr}} \delta \mathbf{n}(t) \cdot \delta \mathbf{n}^T(t-\tau) dt + \frac{1}{T_{corr}} \cdot \int_{T_{corr}} \mathbf{j}_1(t) \cdot \mathbf{j}_1^T(t-\tau) dt + \underbrace{\frac{1}{T_{corr}} \cdot \int_{T_{corr}} \mathbf{j}_1(t) \cdot \mathbf{j}_2^T(t-\tau) dt}_{=0} \\ & + \underbrace{\frac{1}{T_{corr}} \cdot \int_{T_{corr}} \mathbf{j}_2(t) \cdot \mathbf{j}_1^T(t-\tau) dt}_{=0} + \frac{1}{T_{corr}} \cdot \int_{T_{corr}} \mathbf{j}_2(t) \cdot \mathbf{j}_2^T(t-\tau) dt \end{aligned} \quad (V-103)$$

By using

$$\mathbf{s}_{noise} = \mathcal{F}(\mathbf{C}_{corr,Noise}(\tau)) \quad (V-104)$$

the noise covariance matrix for multi jammer reception gets

$$\begin{aligned}
\mathbf{s}_{noise} = & \sigma_{noise}^2 \cdot \mathbf{I}^{Npatch \times Npatch} + |\mathcal{J}_1(\omega)|^2 \cdot \begin{bmatrix} 1 & e^{j\mathbf{k}_{J1}^T(\mathbf{p}_2 - \mathbf{p}_1)} & \dots & e^{j\mathbf{k}_{J1}^T(\mathbf{p}_{Npatch} - \mathbf{p}_1)} \\ e^{j\mathbf{k}_{J1}^T(\mathbf{p}_1 - \mathbf{p}_2)} & 1 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\mathbf{k}_{J1}^T(\mathbf{p}_1 - \mathbf{p}_{Npatch})} & e^{j\mathbf{k}_{J1}^T(\mathbf{p}_2 - \mathbf{p}_{Npatch})} & \dots & 1 \end{bmatrix} \\
& + |\mathcal{J}_2(\omega)|^2 \cdot \begin{bmatrix} 1 & e^{j\mathbf{k}_{J2}^T(\mathbf{p}_2 - \mathbf{p}_1)} & \dots & e^{j\mathbf{k}_{J2}^T(\mathbf{p}_{Npatch} - \mathbf{p}_1)} \\ e^{j\mathbf{k}_{J2}^T(\mathbf{p}_1 - \mathbf{p}_2)} & 1 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\mathbf{k}_{J2}^T(\mathbf{p}_1 - \mathbf{p}_{Npatch})} & e^{j\mathbf{k}_{J2}^T(\mathbf{p}_2 - \mathbf{p}_{Npatch})} & \dots & 1 \end{bmatrix} + \dots
\end{aligned} \tag{V-105}$$

Orthogonalization based nulling with more than one jammer

In equation (V-9), the array weighting vector was calculated, considering one received jammer. Basis for derivation was the criteria $\mathbf{w}_{GS}^T \mathbf{v}_{kJ} = \mathbf{0}$, which requests the array weighting vector being orthogonal to the array manifold vector \mathbf{v}_{kJ} of the jammer.

In case of more than one jammer, this criterion gets

$$\mathbf{w}_{GS}^T \cdot [\mathbf{v}_{kJ1} \quad \mathbf{v}_{kJ2} \quad \dots \quad \mathbf{v}_{kJNj}] = \mathbf{0}^T \tag{V-106}$$

The vectors $[\mathbf{v}_{kJ1} \quad \mathbf{v}_{kJ2} \quad \dots \quad \mathbf{v}_{kJNj}]$ span a N_j dimensional subspace. In order to find a weighting vector \mathbf{w}_{GS}^T , being orthogonal to this subspace, also the Gram-Schmidt orthogonalization is used.

At first, a set of orthogonal vectors is calculated, spanning the same subspace as the vectors $[\mathbf{v}_{kJ1} \quad \mathbf{v}_{kJ2} \quad \dots \quad \mathbf{v}_{kJNj}]$.

For this orthogonal set of vectors, a further orthogonal vector can be found, which can be used as array weighting vector.

Such a set of orthogonal vectors $[\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots]$ can be calculated according to the following equations, which are also given in [108, p. 42].

$$\mathbf{x}_1 = \mathbf{v}_{kJ1} \tag{V-107}$$

$$\mathbf{x}_2 = \mathbf{M}_1(\mathbf{x}_1) \cdot \mathbf{v}_{kJ2} = \left(\mathbf{I}^{Npatch \times Npatch} - \frac{\mathbf{x}_1 \cdot \mathbf{x}_1^T}{\|\mathbf{x}_1\|^2} \right) \cdot \mathbf{v}_{kJ2} \tag{V-108}$$

$$\mathbf{x}_3 = \mathbf{M}_2(\mathbf{x}_1, \mathbf{x}_2) \cdot \mathbf{M}_1(\mathbf{x}_1) \cdot \mathbf{v}_{kJ3} \tag{V-109}$$

$$\mathbf{x}_{N_j} = \mathbf{M}_{N_j-1} \cdot \dots \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{v}_{k_j, N_j-1} \quad (\text{V-110})$$

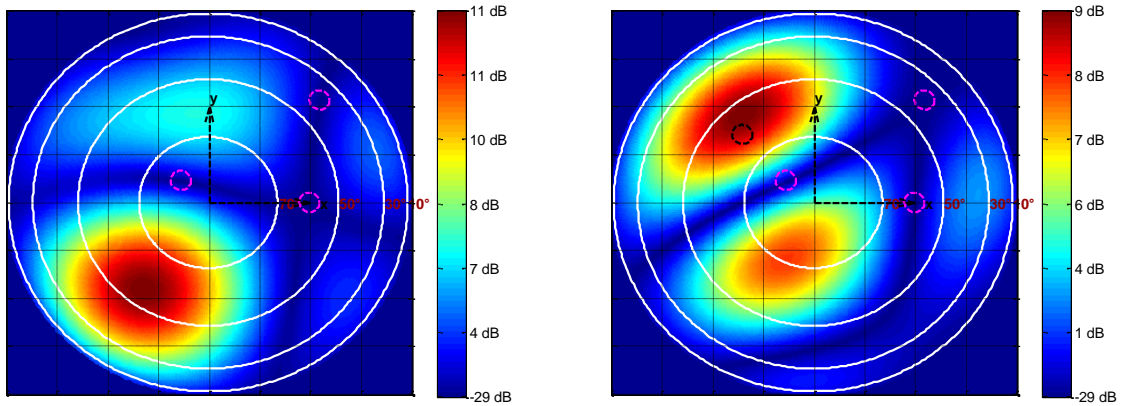
The dependence of \mathbf{M} on the already calculated orthogonal vectors is omitted, in order to simplify the expressions.

The orthogonal weighting vector is calculated according to the next equation

$$\mathbf{w}_{\text{GS}}^T = \mathbf{M}_{N_j} \cdot \dots \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{w} \quad (\text{V-111})$$

with \mathbf{w} being some arbitrary weighting vector.

The following example shows the polar reception pattern for the nulling approach and also for the beam forming approach.



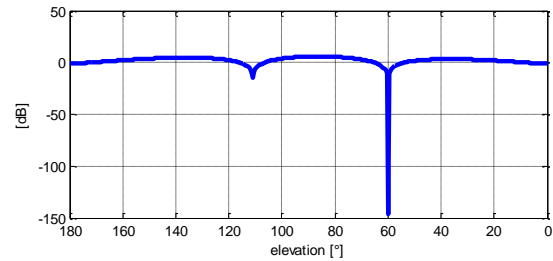
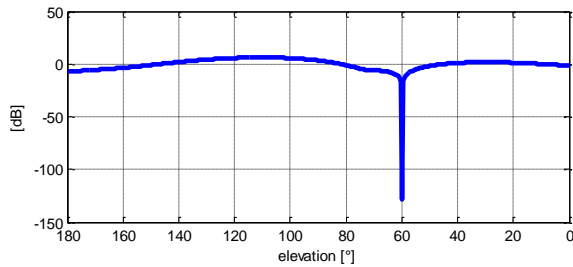
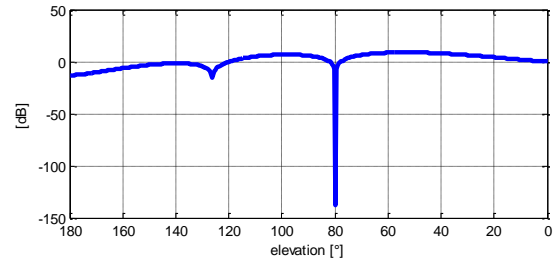
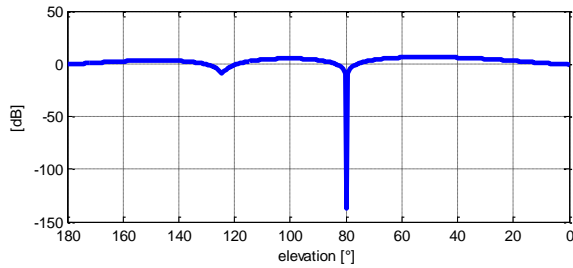
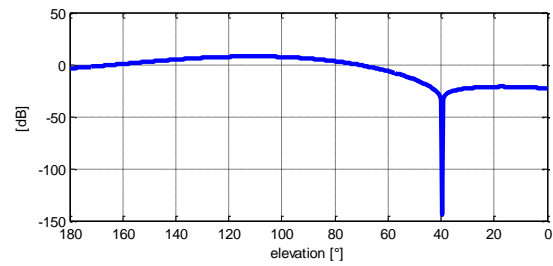
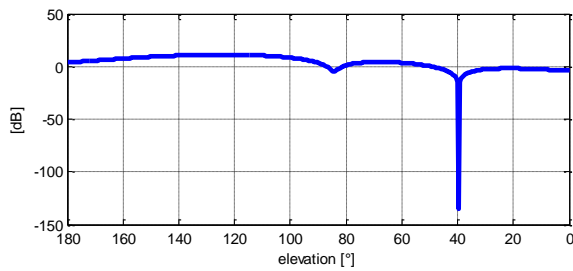
V-125 Nulling – 3 jammer – c6m1_05lambda

V-126 MVDR – 3 jammer – rect9

Additionally, to the polar reception pattern, in the following figures, the reception pattern in the elevation – azimuth plane is shown. There it gets obvious, the “Null” is located at the correct elevation.

The jammer attitudes used for simulation, are given below

$$\begin{pmatrix} \theta_{j_1} = 40^\circ \\ \phi_{j_1} = 45^\circ \end{pmatrix}, \begin{pmatrix} \theta_{j_2} = 80^\circ \\ \phi_{j_2} = 140^\circ \end{pmatrix}, \begin{pmatrix} \theta_{j_3} = 60^\circ \\ \phi_{j_3} = 0^\circ \end{pmatrix} \quad (\text{V-112})$$



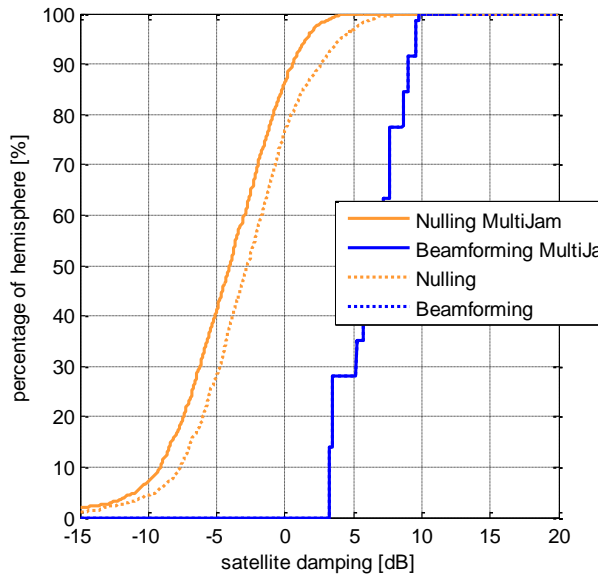
V-127 Nulling – 2D jammer plane

V-128 MVDR – 2D jammer plane

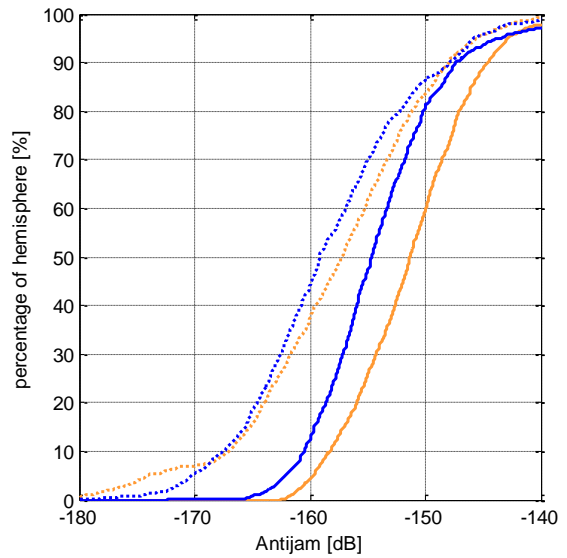
In order to compare the performance of nulling and beam forming in case of three jammers, similar to previous sections, also here the satellites and jammers are permuted across the geodetic grid.

In the previous analysis, the jammer and satellite were moved across a set of geodetic grid points, whereas for every jammer – satellite constellation, a beam forming or nulling solution was calculated. In this section with three jammers, two of them are fixed, whereas the remaining jammer is again moved across the geodetic dome.

For nulling, the Gram-Schmidt orthogonalization based nulling algorithm is used, together with the circular array geometry “c6m1_05lambda”. For beam forming, the MVDR algorithm is used, together with the rectangular array “rect9”.



V-129 Satellite gain – 3 Jammer



V-130 Antijam – 3 Jammer

The simulations above show, that in case of three jammers, the performance of both, nulling and beam forming becomes worse regarding the provided antijam. But beam forming still provides a better antijam than nulling.

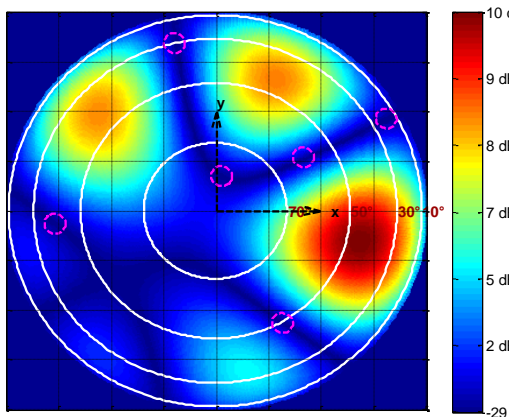
More important than antijam, is the unwanted satellite damping. The simulation shows that the satellite gain of beam forming remains independent on the number of jammers, contrary to nulling. In case of nulling, there is already tremendous satellite damping, which becomes even worse if more than one jammer must be considered

Maximum number of possible jammers

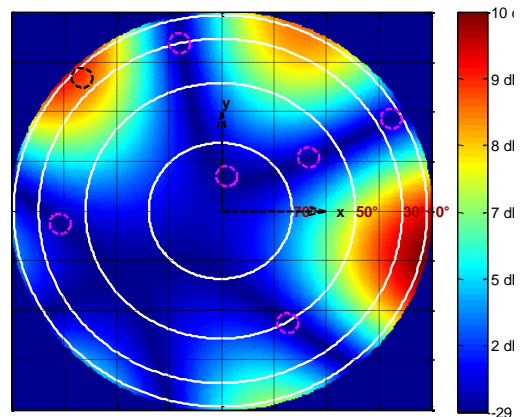
The maximum number of jammers which can be eliminated by nulling or beam forming is $N_{element} - 1$.

One of the best performing nulling arrays was the c6m1_05lambda, having 7 antenna elements. One of the best performing beam-forming arrays was the rectangular array rect9, having 9 antenna elements.

For comparing the performance of beam forming and nulling in case of maximum possible jammers, 6 jammers are chosen. The following two figures show an example constellation with 6 jammers.

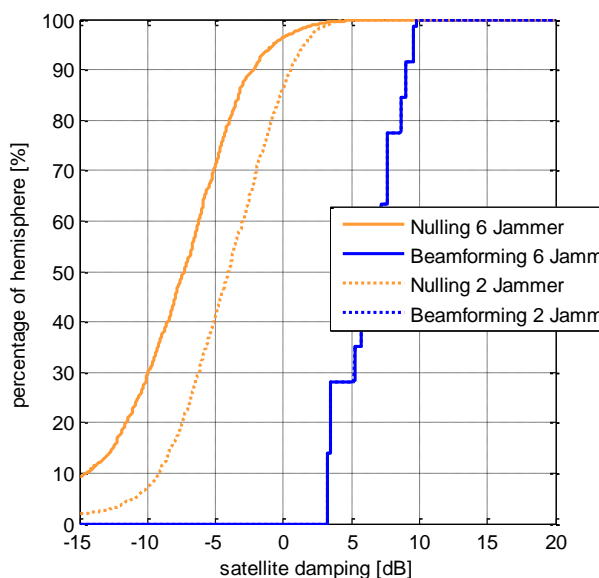


V-131 Nulling - c6m105lambda - 6 Jammer

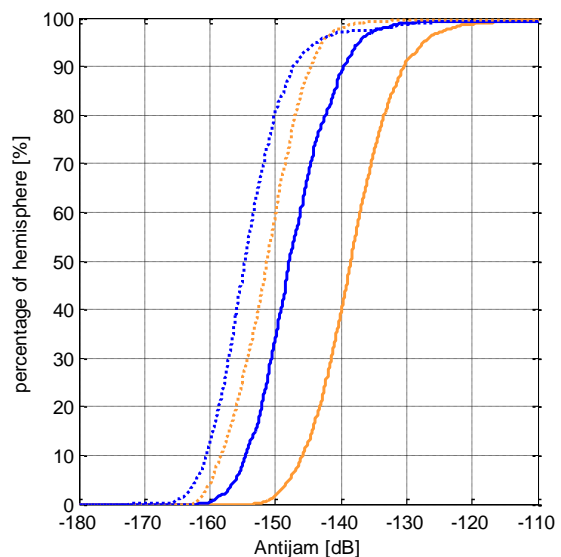


V-132 MVDR - rect9 - 6 Jammer

Additionally to the fixed constellation above, the following figures show a simulation, where one jammer is moved together with a satellite about the hemisphere, considering all possible permutations on a defined geodetic dome grid. All other 5 jammers remain fixed.



V-133 Satellite gain – 6 Jammer



V-134 Antijam – 6 Jammer

The simulation shows that in case of 6 jammers, the provided antijam becomes worse. But even in case of 6 jammers, the beam forming solution using the MVDR approach, gives better results. Especially in case of beam forming, there is no satellite signal degradation.

V - 6.8 Sensitivity of beam forming and nulling regarding array geometry errors

In the previous chapters, antenna arrays without any errors were assumed. The array element positions are known perfectly.

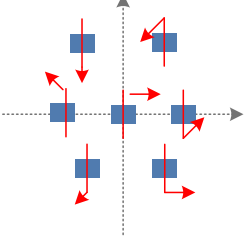
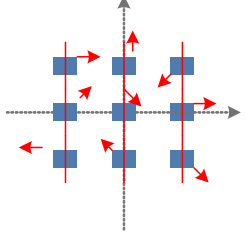
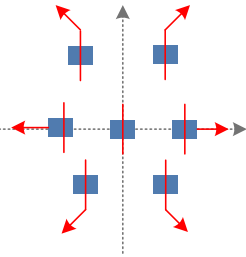
But this does not correspond with the real world. In reality, the antenna array, respectively the positions of the array elements have errors, which are not known exactly.

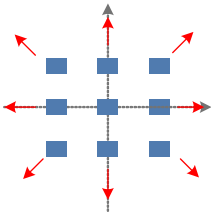
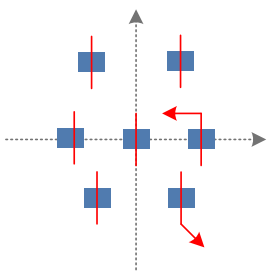
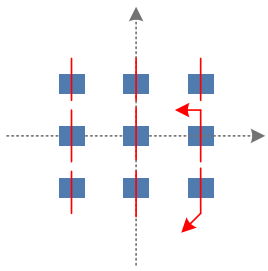
This section evaluates beam forming and nulling with respect to such errors.

Principally, two error types for geometry errors are possible.

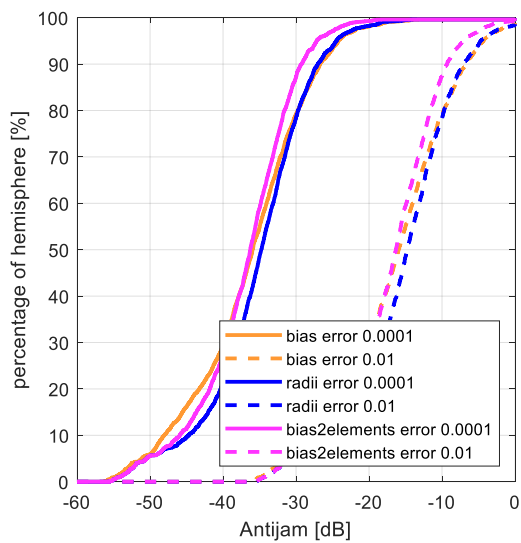
- Bias-like phase center errors of the antenna element: The phase center of the single element is not located in the middle of the element. The phase center shifts can be assumed to be random.
- Location errors of the single antenna element within the antenna array.

The following table lists the array geometries and error topologies, which will be analyzed.

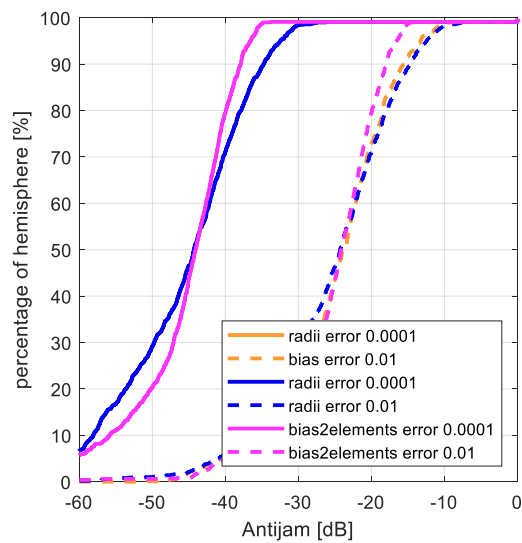
<p>Bias like phase center shift of all antenna elements, with arbitrary error direction.</p> <p>Phase shifts to be analyzed: 0.0001λ , 0.01λ</p>	<p>C6M1_05lambda (for Nulling)</p> 
<p>Wrong position of elements – radius error of all elements</p> <p>Errors to be analyzed 0.0001λ , 0.01λ</p>	<p>RECT9 (for beam forming)</p> 
	<p>C6M1_05lambda</p> 

	<p style="text-align: center;">RECT9</p>  <p>A 3x3 grid of blue squares representing antenna elements. Red arrows point outwards from each element in the four cardinal directions (up, down, left, right). Dotted lines with arrowheads at the center represent the x and y axes.</p>
<p>Wrong position of two elements Errors to be analyzed 0.0001λ , 0.01λ</p>	<p style="text-align: center;">C6M1_05lambda</p>  <p>A 3x3 grid of blue squares. The top-right and bottom-right elements are shifted from their regular grid positions. Red arrows point outwards from each element. Dotted lines with arrowheads at the center represent the x and y axes.</p>
	<p style="text-align: center;">RECT9</p>  <p>A 3x3 grid of blue squares. The top-right and bottom-right elements are shifted from their regular grid positions. Red arrows point outwards from each element. Dotted lines with arrowheads at the center represent the x and y axes.</p>

The following figures shall give an impression, to which extent the nulling performance becomes worse, depending on geometry errors. The effect will be shown for Gram-Schmidt nulling approach and MVDR beam forming, each with the corresponding optimal array geometry.



V-135 Antijam given geometry errors - Nulling



V-136 Antijam given geometry errors - MVDR

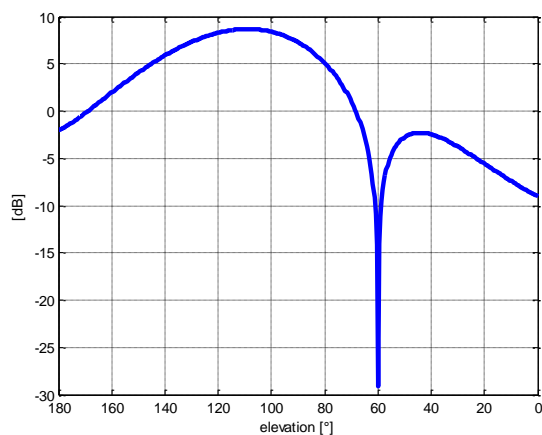
The simulations show, that already small bias like errors of the phase center or radii errors, worsens the nulling performance tremendously.

The gain towards the satellites are almost unaffected.

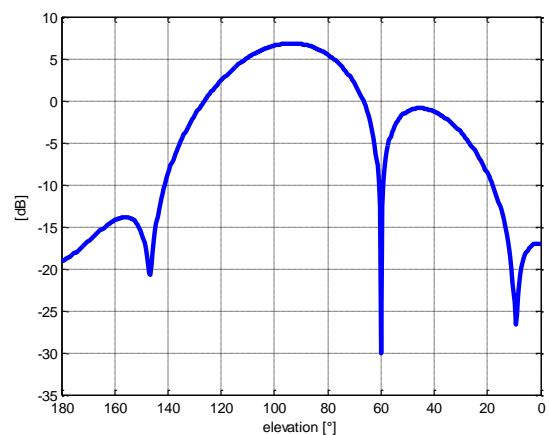
The following two figures give a visual example for the reception pattern, given geometry errors. The figures below show the reception gain pattern in the azimuth plane of the jammer.

As example, the following jammer location was used.

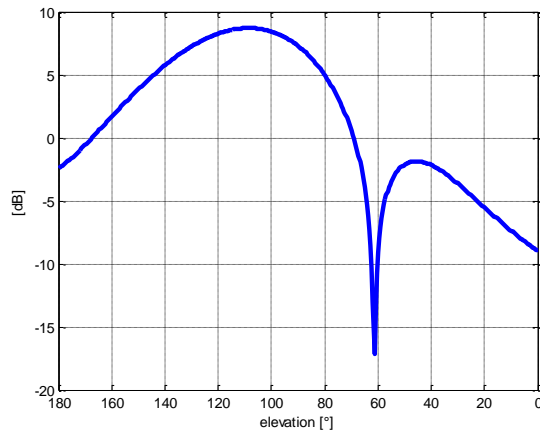
$$\mathbf{\theta}_J = \begin{bmatrix} \phi_J = 20^\circ \\ \theta_J = 60^\circ \end{bmatrix} \quad (V-113)$$



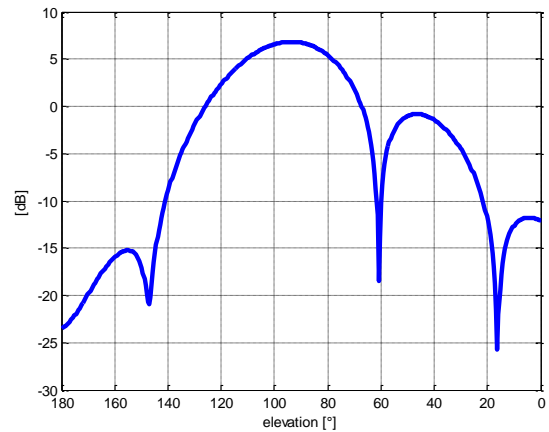
V-137 Nulling - radii error - 0.0001lambda



V-138 MVDR - radii error - 0.0001lambda



V-139 Nulling – radii error – 0.01 lambda



V-140 MVDR – radii error – 0.01 lambda

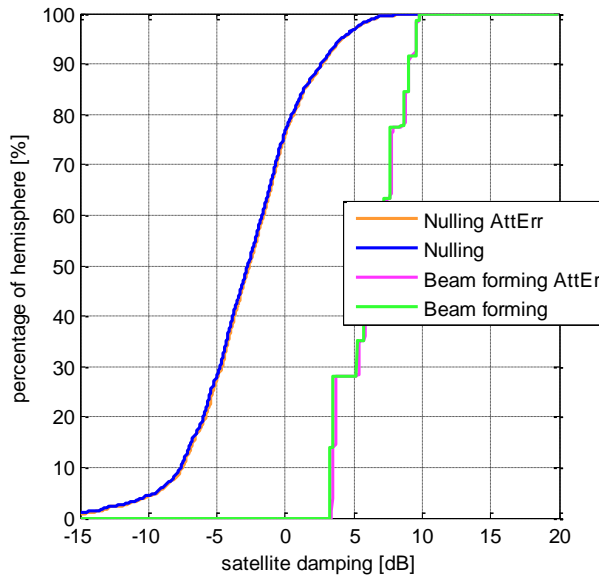
This section shows that the theoretical nulling performance with 140 dB null depth or more, is not possible in real applications. Small array errors are not avoidable. In order to limit the effect of such geometry errors, an antenna calibration is necessary. The other possibility is to use the analytical derived nulling, respectively beam forming solution as a start solution and deepening the null, by using a real time gradient search approach.

This analysis on unknown geometry errors shows, regarding antijam, beam forming has no advantage compared to nulling anymore. Nevertheless, the gain towards satellites given beam forming, remains almost unchanged. Therefore, beam forming should be preferred anyway.

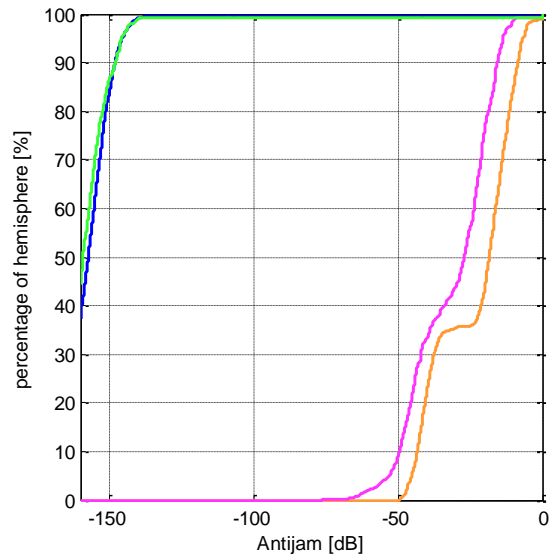
V - 6.9 Beam forming and nulling performance in case of jammer attitude error

In the previous sections, an error free knowledge of the jammer and satellite attitude was assumed. But as the null is very steep within the reception pattern, already small attitude errors worsen the nulling performance tremendously. Therefore, this section is to analyze the nulling performance loss in case of small attitude errors towards the jammer. Moreover, also small attitude errors towards the satellites are assumed, which worsens the satellite gain.

As attitude error in the following simulation, an elevation error of 1 degree is assumed.



V-141 Satellite gain – Attitude Error



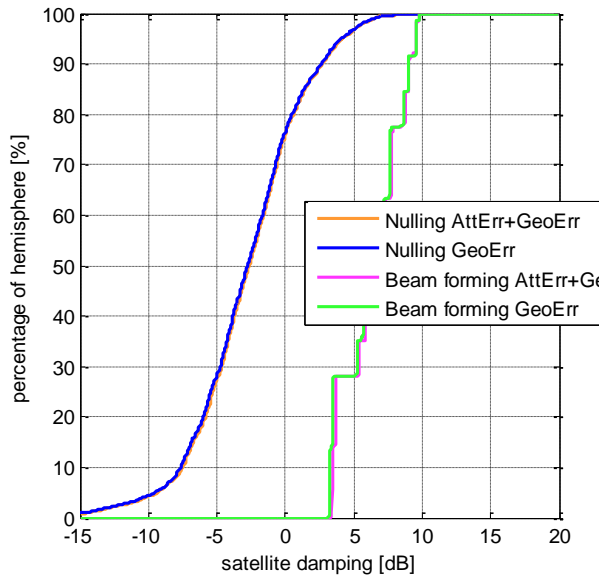
V-142 Antijam – Attitude Error

The figures V-141 and V-142 compare the satellite gain and the resulting antijam in case of the assumed attitude error. In both cases, an error free antenna array was assumed.

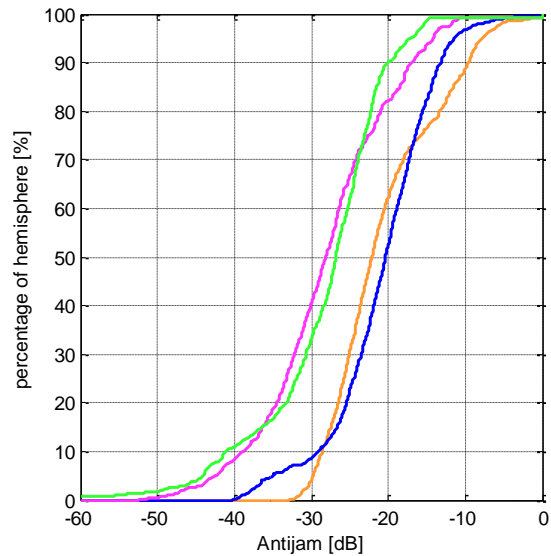
Because the null is very steep in the reception pattern, the impact of an attitude error on antijam is very high. The antijam reduction is more than 100 dB; in case of nulling and in case of beam forming. The attitude error towards the satellite direction has only minor influence on the satellite gain, due to a relatively broad main lobe.

In the following, the impact of attitude errors will be analyzed in case of an antenna array with unknown geometry errors, as they were introduced in the previous section.

The null depth in case of unknown array geometry errors is reduced anyway up to only -30dB. But the null is much broader compared the sharp null, given error free array geometries. Therefore, the impact of attitude errors in addition to already existing array geometry errors, is very small.



V-143 SatGain - Attitude error and geometry error



V-144 Antijam - Attitude error and geometry error

V - 6.10 Conclusion reception pattern shaping

The main intention of this section was to evaluate in depth, the advantages and disadvantages of beam forming respectively nulling.

Nulling has the huge advantage that it can be implemented within the antenna electronics. Therefore, nulling can be used to increase antijam of all standard GPS receivers. In order to realize beam forming, a special receiver is necessary.

As the evaluation of nulling and beam forming has shown, beam forming provides a slightly better jammer damping as nulling. But what is much more important, beam forming causes no satellite damping as it is in case of nulling. Already 10 dB satellite damping could cause a loss of lock in bad signal to noise environments.

An additional contribution of this section is the analysis of nulling respectively beam forming performance, depending on the array topology. Many pattern geometries are possible. The analysis of this section has shown that for nulling and beam forming, different array geometries must be used to get optimal results.

Besides the influence of array geometry, beam forming and nulling were also evaluated with respect to their sensitivity to unknown array geometry errors and jammer attitude errors.

Even here, beam forming is less sensitive regarding these errors.

If array geometry errors and attitude errors occur at the same time, there is only a small remaining dependence on attitude errors. Unconsidered, respectively unknown geometry errors influence the shape of the reception pattern. One major effect is that the “Null” at the estimated jammer attitude becomes wider and less deep. As a result, even if the real jammer attitude deviates from the estimated attitude, the “Null” shows almost the same damping.

Additionally, this section developed a novel nulling approach, providing almost equal performance as beam forming. Although this novel algorithm is based on beam forming, no pre knowledge of the satellite positions within the antenna coordinate system is necessary.

The nulling and beam forming algorithms analyzed in this section, give analytical beam forming and nulling solutions respectively, complex array weighting vectors. As the analysis in this section has shown, given unknown phase center shifts of antenna pattern or in case of unknown pattern position errors, the jammer nulling capability is worsened. In order to compensate these errors, an array calibration is necessary, or an iterative nulling solution can be used, starting with the analytically derived array weighting vector, as initial solution.

V - 7 Mission planning in jammed environments

Mission planning is a collective term for all necessary activities and steps in order to choose all parameters of a planned flight, which could be a drone observation mission or even a military operation, in such a way that the maximum possible quality of GPS navigation is available along the mission.

The parameters to be selectable for the user are mission trajectory, which could in case of assumed jammers, optimized to get a GPS aided navigation as long as possible. Moreover, the mission time can be chosen in such a way to get a dense satellite coverage in the defined area. For system developers, whose purpose is to set up an aided GPS navigation, the overall performance of the combined GPS – Inertial Navigation system can be evaluated pre mission and improved.

This all is possible by using the equivalent base band simulation concept, developed in chapter IV. Especially in safety critical applications like air taxi operations or flight auto landing, or in military operations, jamming poses a vital security issue.

In section V - 2.5 of this chapter, the discriminator error variance in case of different jamming scenarios was calculated. But even with this information, a deterministic loss of lock prediction based on the assumed jamming power is not possible. The tracking loop is a closed loop system, which can be represented as a nonlinear differential equation system. The nonlinearity is caused by nonlinear tracking filter equations and – what is the vital nonlinearity with respect to loss of lock – the nonlinear discriminator function. The discriminator noise, caused by jamming, can be viewed as a superposition to the misalignment measurement of the discriminator and provides a stochastic stimulus of the nonlinear differential equation – making a stochastic nonlinear differential equation. In most cases there is no deterministic solution, which noise variance causes the system to become instable. Selected examples are analyzed in [104] or [126], solving the Langevin differential equation. But in most cases, a simulative determination of the loss of lock, doing many simulation runs, is the best approach. This can be done by using the developed equivalent base band simulation approach. Because even long missions can be simulated in very short times and many different jamming realizations can be simulated.

The figure IV-2 summarizes all necessary steps for mission simulation. All these steps are already developed in this thesis in previous chapters.

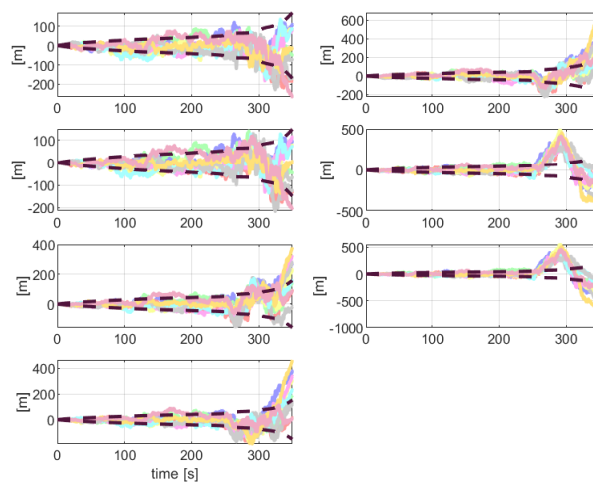
The simulation must be carried out with different jammer realizations respectively in case of white noise jamming, with different seeds.

In the following example, a jamming scenario with a white noise jammer is evaluated together with the DA42 approach. The jammer is located at the final mission trajectory coordinates, at a height of 10 km.

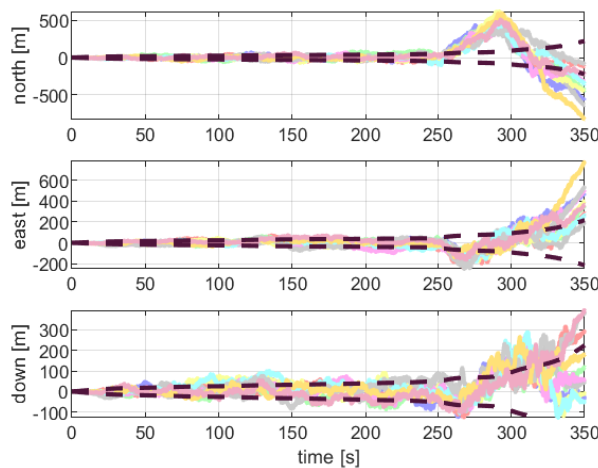
The jamming power is selected to be 1 kW. The used jammer is a noise jammer, whereby the power is evenly distributed about 2 MHz.

Several simulation runs are evaluated with each having a new jammer signal realization.

The following figures show the tracking behavior and especially the loss of lock behavior.



V-145 Pseudoranges - given jammed environment



V-146 Position solution - given jammed environment

Each scenario uses equal dynamics and equal jamming power. The only difference is the seed for the Gaussian noise jammer. As the simulations show, the loss of lock occurs not deterministically, but stochastically at some most probable value.

VI Summary

Contributions from “Robust Accuracy and Dynamic Continuity”

This thesis classifies the possible GPS tracking architectures into categories and compares the best performing and robust tracking architecture of each category against the best one of all others. This procedure gives a structured and representative comparison of possible tracking architectures in different scenarios and different environments. The categories being compared are:

- Classical scalar fixed gain tracking loops against scalar optimal filter tracking loops
- Scalar tracking against vector tracking
- Total state realizations against error state realizations
- Unaided tracking against aided tracking

For comparison, a metric based on error variances and error correlation times for raw data and position solution is developed.

The following table summarizes the results of the comparison.

Classical scalar fixed gain	Scalar optimal filter tracking
The loop filter order must be selected based on heuristic methods. An analysis regarding some remaining control error must be done.	The tracking filter order is an inherent result of the assumed platform dynamic by using the corresponding dynamic model in the Kalman filter propagation equation.
Tuning of a tracking loop is a gradual walk between minimum bandwidth in order to keep the raw data error variance as small as possible and on the other side, selecting a closed loop tracking bandwidth high enough, to follow the line of sight dynamic. The tuning of the scalar fixed tracking loop is mainly bandwidth oriented and requires some experience, what an appropriate bandwidth design is.	Application of scenario matched tuning is possible. All platform dynamics and noise environment characteristics can be mapped bijective on the Kalman filter tuning parameters. The closed loop tracking bandwidth is a direct consequence of the selected dynamic model and the applied tuning parameters. Especially in case of planned missions, the dynamic behavior is known and a corresponding dynamic model can be selected. All deviations from this dynamic model are also known pre mission and can be used to model the errors within the Kalman filter. This approach gives an optimal tuning
Classical scalar fixed gain tracking does not provide inherently some error covariance information. This thesis shows that in parallel to tracking, a dynamic state space error covariance model can be derived, based on interference. But it is not possible to consider within the error covariance information the dynamic stress, respectively the incapability of the tracking loop to follow the line of sight dynamic.	Inherently available tracking error covariance information by the Kalman filter state error covariance. The error covariance includes noise caused errors and also dynamic caused errors.
	Easy online adaptive tuning. Changing platform dynamic characteristics and a changing signal to noise environment, changes automatically the tuning, by mapping these values on the Kalman filter tuning parameters

Scalar optimal filter tracking	Vector optimal filter tracking
In case of tracking outage of one satellite, time-consuming reacquisition is necessary.	As long as 4 or more satellites are available, immediate reacquisition, if one satellite is temporarily not available
No inter satellite aiding. Each channel-wise satellite tracking loop works independent.	The line of sight dynamic of different satellites is correlated. Due to vector tracking, the satellites aide one another. A smaller closed loop tracking bandwidth for each tracking channel is the consequence, with higher robustness against interference.
<p>For positioning solution, two filter stages are cascaded. Raw data tracking is the first filter stage. The scalar tracking loops provide raw date error variance information. Due to filtering, the raw data errors are strongly correlated in time. The raw data are measurements for the subsequent positioning filter. There, the time correlation of raw data error's must be considered.</p> <p>Cross correlation of raw date errors is not considered within the positioning filter.</p>	<p>Positioning solution and tracking within one filter stage. No filter cascades and therefore no correlated coupling errors.</p> <p>Better results for position error covariance information, because it is derived directly, based on the modeled dynamic and discriminator error covariance.</p> <p>Error cross correlations between different satellites are considered.</p>

Unaided total state realizations	Unaided error state realizations
<p>The platform dynamic model is integrated within the total state tracking filter.</p> <p>Errors of a wide dynamic range must be separated by the tracking filter, which are the line of sight dynamic between receiver and satellite and also for example the low dynamic clock error or ionosphere delay. This wide range of dynamics makes the filter very sensitive regarding tuning. An optimal separation of different errors is not always possible.</p>	<p>In order to set up an error state filter without external aiding, in addition to the error state tracking filter, a total state model for dynamic propagation must be realized.</p> <p>Depending on the validity of the selected dynamic model, the error state filter still has to cope with all residual dynamic – which can be almost the total state dynamic in high dynamic scenarios.</p> <p>But nevertheless, tuning is simplified if an additional external total state dynamic model for state propagation is used.</p>
	In low dynamic scenarios, where the external total state dynamic model fits more the real dynamic, better tracking results with lower raw data and position error variance are possible.

Unaided tracking	Aided tracking
<p>In case of a low aiding rate, the performance of an unaided vector tracking approach, given low dynamic scenarios like the used DA42 approach, is equivalent to the aided tracking. (Only valid for the unjammed case)</p> <p>But already in high dynamic scenarios, like the F-22 fighter trajectory, the aided approach provides better results, even with a 1 Hz aiding rate. (Only valid for the unjammed case)</p>	<p>Aided vector tracking in error state realization provides best tracking results.</p> <p>All possible aiding errors can be estimated by such a tracking architecture and also compensated.</p> <p>Aiding errors which matter are: Noise like aiding errors, aiding errors due to low aiding rate, aiding delay and tilt aiding errors.</p> <p>Given a 40 Hz aiding rate, robust tracking, even in case of high jamming attacks is given.</p>
	<p>Aided vector tracking in total state realization is possible, but the filter is very sensitive regarding tuning of the implemented aiding error models.</p> <p>The dynamic range within the total state filter is very high, thus despite aiding, in the filter the full platform dynamic is present.</p> <p>Better results regarding aiding error estimation are possible with an error state realization.</p>
	<p>Aided tracking together with scalar tracking filters is possible, but shows worse results in case of tilt aiding errors.</p> <p>It is not possible to estimate the tilt aiding errors in the scalar filter.</p> <p>Estimation of the tilt aiding errors in the subsequent positioning filter is possible, but the estimation results are of bad quality. This is because parts of the tilt aiding errors are already compensated by the scalar raw data tracking filters, by mapping some parts on other errors like the local estimated aiding delay. Thus, not the full tilt aiding errors are included within the estimated pseudoranges.</p>

Besides the comparison of the mentioned tracking architectures, this thesis provides further vital contributions.

For classical scalar fixed gain tracking loops, a state space representation is developed. Based on this state space representation, an error covariance model is realized, providing the raw data error covariances.

With focus on optimal filter tracking, a novel tuning approach is developed, being called “scenario matched tuning”. The platform dynamic characteristics and also the assumed noise environment is mapped directly onto the tuning figures of the Kalman filter. This approach realizes a kind of “optimal tuning” and also adaptive tuning while being on a mission. Especially adaptive tuning is very easy to implement, because actual signal to noise ratio measurements along the mission can be directly mapped onto the corresponding Kalman filter parameters. Also estimated platform dynamic, being derived for example online from flight control outputs, can directly be mapped on

the Kalman filter dynamic model parameters. For all tracking architectures being developed, a detailed discussion of the applied “scenario matched tuning” is provided.

In the context of scalar optimal filters, also an optimal filter-based carrier aided code tracking loop is developed, where the applied Kalman filter works as a loop filter and additionally, realizes the fusion of code and carrier tracking results. In contrast to fixed coupling between carrier and code tracking loop in classical architectures, here an optimal adaptive coupling between carrier and code tracking loop is realized, based on the respective state error covariances.

In order to realize an unaided scalar optimal filter-based error state approach, a new tracking architecture is developed. The optimal filter is realized in error state space, while an external additional total state dynamic model is used for state propagation.

An optimal filter-based positioning filter is developed, designed to interface the analyzed scalar optimal filters. Especially the error covariance information, provided within the state error covariance matrix of the scalar filters, can be used within the measurement error covariance matrix of the positioning filter.

In literature, if unaided error state vector tracking architectures are given, only the error state filter is discussed, but the whole tracking architecture is missing. This section develops in detail an error state vector tracking approach, consisting out of an error state Kalman filter and an additional total state filter for dynamic propagation.

For aided tracking architectures, it is of great importance to consider, estimate and compensate the aiding errors. This thesis develops models for noise like aiding errors, for aiding errors due to low aiding rate, for aiding delays and for tilt aiding errors. The mentioned aiding error models are developed in such way to fit the following three aided tracking architectures. In each aided tracking architecture, a detailed tuning discussion of the aiding error models is given.

At first an aided error state vector tracking architecture is developed. Two different approaches, how aiding is integrated into the tracking architecture, are developed. The first approach uses an additional total state propagation, where aiding is used as an excitation of the corresponding additional, total state differential equation system. The second approach uses two coupled Kalman filters. An error state Kalman filter and a total state Kalman filter. The total state Kalman filter integrates the velocity aiding as a measurement. Within this additional total state Kalman filter, already noise like aiding errors are modeled. Thus, this filter is able to prevent aiding with bad quality from entering the tracking loop. Moreover, this additional total state filter is able to estimate the platform acceleration based on the aiding velocity. This acceleration is necessary for estimating aiding delay.

Contributions from “GPS Modeling and Simulation”

In this thesis, an equivalent base band simulation architecture is developed, being only based on code- and carrier phase dynamic together with a realistic stimulation of discriminator noise or interference caused error variance. Thus, tracking with all kinds of tracking filters can be simulated, without the need of real signal or software space segment simulators and GPS receivers or software receivers. Also, any signal to noise scenarios and interference scenarios are considered. Moreover, this equivalent base band approach is much faster than real time, enabling pre mission simulation

of tracking behavior and in case of jamming scenarios, also simulative loss of lock predictions. Due to short simulation times, many scenarios can be simulated in short time in order to gain a deep understanding of the tracking behavior in planned missions.

In a first stage, the correlation core is modeled as first order differential equation system, using the measured line of sight code phase dynamic and the estimated code phase dynamic as an excitement. For each satellite, such a differential equation system is set up. The discriminator measurement is a state of such a differential equation system and equals the discriminator measurement, a real discriminator provides. The tracking filter equations are the same as they are in a real receiver or software receiver. Because the tracking filter equations are the one of the real tracking filter implementations, the corresponding Kalman gains and state error covariances are calculated while executing this simulation. These Kalman gains are stored and needed for the second stage of simulation, as described below.

In a second stage, the tracking filter is also written as a coupled first order nonlinear differential equation system and combined with all differential equations of the correlation cores, into one big first order nonlinear differential equation system. This representation provides the advantage, that the whole tracking loop, together with all mutual couplings, is represented in one state space system. Using the corresponding linearized state space matrix, the closed loop tracking bandwidth for each satellite channel can be calculated, also in case of vector tracking. Especially in vector tracking, this special insight provides the information, which satellite is aided most and which is less correlated with all other satellites. (A high closed loop tracking bandwidth suggests little aiding by other satellites). Moreover, the eigenvalues of the system matrix can be used as an analytical evaluation method of tracking stability. For receiver designers, this feature is highly valuable because it provides the information, if the designed tracking architecture in general and the tuning provides a stable tracking. Also, for mission planning and for homologation, this approach is extremely helpful. For any trajectory, in advance, the stability can be verified and the closed loop tracking bandwidth. It provides even the possibility to apply Monte Carlo methods, by varying all possible tuning parameters and also mission parameters, like dynamic and interference in defined limits, in order to evaluate the worst-case stability and bandwidth margins.

As well the closed loop tracking bandwidth and the eigenvalues can be calculated with a selectable sampling rate along the mission, respectively trajectory. Thus, the bandwidth and eigenvalue behavior along a planned trajectory can be evaluated pre mission.

This thesis develops for each tracking architecture, both mentioned state space realizations.

In addition to the equivalent base band simulation architecture, an IF band, software-based space segment simulator and IF band GPS software receiver is developed. The major intention of this software-based space segment simulation and GPS receiver is the verification of the developed equivalent base band model and the analysis of non-stationary tracking situations, especially in case of loss of lock. Because the focus is on analyzing tracking itself, the limitation to IF band is absolutely sufficient. The space segment sided up conversion to L1 band and the GPS receiver sided down conversion to IF band, would provide no benefit.

For both, the equivalent base band approach and the IF band space segment simulator, at first a detailed model of the line of sight dynamic between satellite and receiver is derived, used for code phase simulation and upon, code and carrier signal construction. For line of sight dynamic, besides

the pure satellite and platform dynamic, also the phase influence of ionosphere is considered and the phase influence of the antenna itself, which is especially very important, if beam forming or nulling algorithms for antenna reception pattern forming are used. Besides the line of site phase dynamic model, also a satellite signal link budget model and jammer signal link budget model are developed, including besides the free space transmission, also the influence of the antenna reception pattern and also receiver internal signal processing steps.

Contributions from “Reception Resilience”

For tracking filter tuning and for simulation of tracking, especially with the purpose to determine the maximum possible jamming power before loss of lock occurs, a precise calculation of the discriminator error variance is vital. Therefore, this thesis develops a modified discriminator error variance calculation scheme, especially for high interference scenarios. For that, for a white noise jammer and PN code jammer, the remaining jammer power and signal form in time and frequency at different stages within the GPS receiver are calculated. As a result, the residual jamming power after correlation is derived, considering also the de-spreading with the replicated spread spectrum code. This residual post correlation jamming power, together with the satellite signal power at IF interface, can be used for exact discriminator error variance calculation.

For antijam improvement within the GPS receiver, the optimal design of the IF filter bandwidth in combination with selecting the correlation time, are important measures. For a white noise jammer and PN code jammer, a combined optimal design of IF bandwidth and correlation time is analyzed in this thesis. For PN code jammer in particular, the impact of the special PN jammer signal structure in combination with the IF bandwidth and correlation time, on the discriminator error variance, are evaluated.

Moreover, the design of the down conversion stage from antenna interface till IF filter and automatic gain control, are evaluated. This stage does not provide additional antijam, but contributes to maximum antijam through a minimum noise figure design. Formulas for noise figure, depending on an active antenna front end, in dependence of the amplifier position, are developed. As well for one amplifier, or in case of long necessary cabling, also two amplifiers.

One of the most powerful resilience measures against antijam is active reception pattern control of the antenna. This thesis starts with comparing different Nulling algorithms and develops upon a novel Nulling algorithm, being based on beam forming with virtual satellites. The latter algorithm performs better than Nulling algorithms suggested by literature, at the cost of high processing load and carrier phase manipulations, varying across the hemisphere.

This thesis also introduces a beam forming algorithm, based on literature.

For Nulling and beam forming, different possible pattern array geometries are evaluated in order to find the best fitting array geometry for Nulling and beam forming. Nulling performs best with only few patch elements. In this case, the resulting main lobes are very broad and thus compensating the main disadvantage of nulling, which is the unintentional satellite damping. Nulling places the minimum reception gain towards the jammer, but does not maximize the reception gain towards the satellites. Therefore, many broad main lobes ensure a good high gain

coverage of the hemisphere. Moreover, besides the Nulling algorithm, being based on beam forming with virtual satellites, an antenna array with a central element is best performing.

For beam forming, antenna arrays with many array elements are beneficiary. Here, there is no unintentional satellite damping, because the reception pattern is maximized towards the known satellites and minimized towards the jammer direction. Many array elements give very thin and sharp main lobes, which minimizes the danger of unintentional amplification of a jammer, in case the jammer attitude is similar to the satellite attitude, especially given low satellite elevations within the antenna coordinate system.

Beam forming and also Nulling show a damping towards the interference, of more than 150 dB. However, this damping is more theoretical in nature, due to perfect jammer attitude estimation and a perfect array geometry.

Real arrays are imperfect due to small patch positioning errors of different characteristics. This thesis does some detailed analysis, which impact do different geometry errors have on jammer damping and satellite gain, given Nulling and Beam forming. The evaluations show that in case of real, imperfect array geometries, the damping towards the jammer is reduced to only 30 to 50 dB, with minima, being much broader. This is valid for beam forming, as well for Nulling. The satellite gain is only slightly affected.

Due to very sharp minima in case of a perfect array geometry, already small attitude errors reduce the damping to only about 30 to 50 dB. In case of real, imperfect array geometries and thus, much broader minima, small attitude errors only have smaller effects. As conclusion, given real imperfect arrays and small jammer attitude errors, Nulling can offer an almost comparable performance like beam forming.

This thesis does additionally some comparing of beam forming and Nulling in case of more than one jammer up to the maximum number of jammers, which depends on the array elements. In such cases, beam forming always shows up to 10 dB better jammer damping than Nulling.

VII Appendix

VII - 1 Error modeling

VII - 1.1 Receiver clock error model

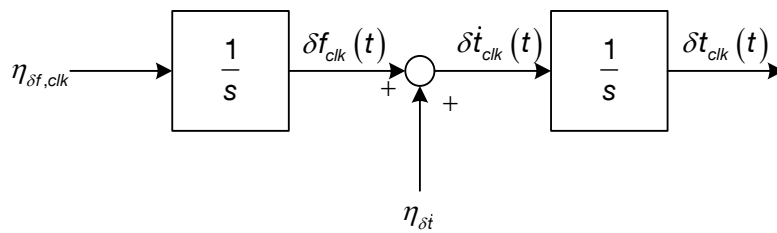
Within tracking loops, the receiver clock error constitutes an essential error part. The code and carrier tracking loops intend to keep the phase of the replicated code and carrier signal aligned with the received code and carrier phase by controlling the NCO frequency.

The receiver internal clock provides the base frequency for the NCO. A deviation of this base frequency caused by a receiver clock error leads to a phase error which must be compensated by the code and carrier tracking loop.

This frequency adjustment cannot be distinguished from a necessary frequency adjustment due to line of sight movement and the resulting doppler frequency in a single satellite tracking channel.

Depending on the oscillator type and the used crystal, the error characteristic is different. In [33] can be found a detailed analysis of different clock error types. In this work, the clock error is modeled with a bias and drift.

The following figure shows the composition of the receiver clock error.



VII-1 receiver clock error model

$$\begin{aligned}
 \delta t_{clk}(t) &:= \text{clock error} & (VII-1) \\
 \delta t_{clk}^{\dot{}}(t) &:= \text{clock drift} \\
 \delta f_{clk}(t) &:= \text{changing clock drift} \\
 \eta_{\delta t} &:= \text{clock drift noise} \\
 \eta_{\delta f} &:= \text{frequency drift noise}
 \end{aligned}$$

The clock drift $\delta t_{clk}^{\dot{}}(t)$ is a combination of a clock drift noise and a changing clock oscillator frequency $\delta f_{clk}(t)$, whereby this changing frequency is modeled as a Wiener process. The complete clock error model in state space form can be written according to the next equation.

$$\begin{bmatrix} \delta t_{clk}^{\dot{}} \\ \delta f_{clk}^{\dot{}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta t_{clk} \\ \delta f_{clk} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \eta_{\delta t} \\ \eta_{\delta f} \end{bmatrix} \quad (VII-2)$$

As reference for the clock errors, a Novatel OEM628 GPS receiver is used. Real measurements given in [33, p. 122] show a clock drift noise variance $\sigma_{\eta\delta i}^2 = \sigma_{\eta\delta i}^2 = \frac{0.05^2}{c^2} \left[\frac{m}{s} \right]$.

The oscillator frequency change $\delta f_{clk}(t)$ can be modeled as a Wiener process. The variance of this Wiener process can be derived according to next equation.

$$\begin{aligned} \sigma_{\delta f}^2(t) &= \mathcal{E} \left\{ (\delta f_{clk})^2 \right\} = \mathcal{E} \left\{ \int_0^t \eta_{\delta f}(t) dt \cdot \int_0^{\tau} \eta_{\delta f}(\tau) d\tau \right\} = \mathcal{E} \left\{ \int_0^t \int_0^{\tau} \eta_{\delta f}(t) \eta_{\delta f}(\tau) dt d\tau \right\} \\ &\dots = \mathcal{E} \left\{ \int_0^t \int_0^{\tau} \eta_{\delta f}(t) \eta_{\delta f}(\tau) \delta(t-\tau) dt d\tau \right\} = \mathcal{E} \left\{ \int_0^t \eta_{\delta f}^2 dt \right\} = \int_0^t \mathcal{E} \left\{ \eta_{\delta f}^2 \right\} dt = \sigma_{\eta\delta f}^2 \cdot t \end{aligned} \quad (VII-3)$$

Using (VII-3), the variance of the needed driving noise $\sigma_{\eta\delta f}^2$ can be calculated. Measurements according to [33] show a frequency drift of $0.5 \frac{m}{s}$ after 500 s.

$$c^2 \cdot \sigma_{\delta f}^2(t=500) = c^2 \cdot \sigma_{\eta\delta f}^2 \cdot 500 = 0.5^2 \quad (VII-4)$$

$$\sigma_{\eta\delta f} = \frac{0.5}{c \cdot \sqrt{t}} = \frac{0.5}{c \cdot \sqrt{500}} \quad (VII-5)$$

Matlab is used to generate many realizations of the clock error. As discrete implementation, the following state space system is used.

$$\begin{bmatrix} \delta t_{clk,k} \\ \delta f_{clk,k} \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta t_{clk} \\ \delta f_{clk} \end{bmatrix} + \begin{bmatrix} \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \cdot \begin{bmatrix} \eta_{\delta i,k} \\ \eta_{\delta f,k} \end{bmatrix} \quad (VII-6)$$

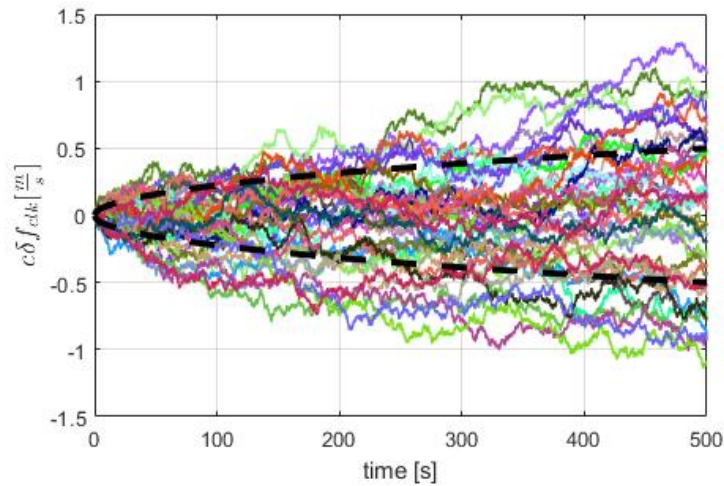
Considering the generation of the discrete seed noise, the following relationship between the time continuous and time discrete variance must be kept in mind.

$$\begin{aligned} \sigma_{\delta f,k}^2 &= \mathcal{E} \left\{ (\delta f_{clk,k})^2 \right\} = \mathcal{E} \left\{ \left(\sum_{k=1}^N \Delta T \cdot \eta_{\delta f,k} \right) \cdot \left(\sum_{n=1}^N \Delta T \cdot \eta_{\delta f,k} \right) \right\} = \mathcal{E} \left\{ \sum_{k=1}^N \Delta T^2 \cdot \eta_{\delta f,k}^2 \right\} = N \cdot \Delta T^2 \cdot \sigma_{\eta\delta f,k}^2 \\ N \cdot \Delta T^2 \cdot \sigma_{\eta\delta f,k}^2 &\stackrel{!}{=} t \cdot \sigma_{\eta\delta f}^2 \quad \text{using } N = \frac{t}{\Delta T} \rightarrow \sigma_{\eta\delta f,k}^2 = \frac{1}{\Delta T} \cdot \sigma_{\eta\delta f}^2 \end{aligned} \quad (VII-7)$$

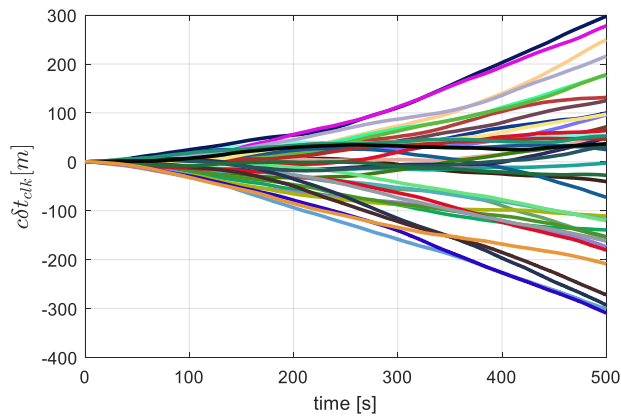
Within Matlab, the respective seed noise is generated using the following equation

$$\eta_{\delta f,k} = \frac{\sigma_{\eta\delta f}}{\sqrt{\Delta T}} \cdot \text{randn}(1,N) \quad (VII-8)$$

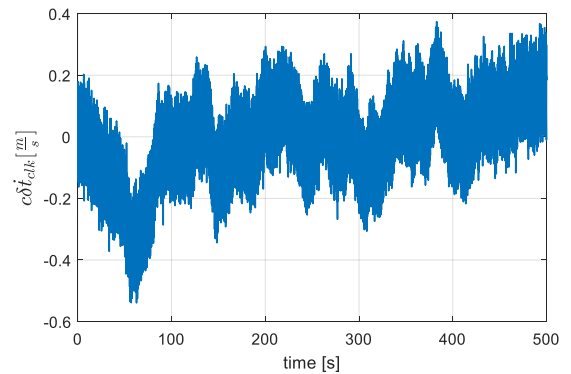
The following figure shows different realizations of δf_{clk} .



VII-2 Realizations: Changing clock frequency error



VII-3 Realizations: Clock error



VII-4 Single realization - clock drift

The figures above show the characteristics of the clock error model used in this work. Figure VII-3 shows the clock error normed to meter. Depicted are many realizations of the clock error in order to show the stochastic characteristics. In figure VII-2, the clock drift caused by the Wiener process is shown. Also here many realizations have been generated in order to validate the analytic variance prediction from equation (VII-5), represented by the dotted black line. Figure VII-4 shows a single realization of the complete clock drift, which is a combination of the mentioned wiener process and a noise like part.

VII - 1.2 Tuning a Gauss Markov process corresponding to line of sight dynamics

In this work, a Gauss Markov process is used quite often in different models. One application is the line of sight acceleration, which is modeled as a Gauss Markov process. Therefore, in this section, the tuning of a Gauss Markov process will be discussed.

The Gauss Markov process is represented by the following differential equation, driven by the seed noise q .

$$\dot{x}(t) = -\frac{1}{\tau} \cdot x(t) + q(t) \quad (\text{VII-9})$$

Considering the line of sight acceleration as a Gauss Markov process, its variance σ_x^2 and correlation time τ was derived by evaluation the line of sight acceleration in a planned mission according to scenario matched tuning in section III - 3.4 .

For tuning the Gauss Markov process, the variance of the seed noise σ_q^2 must be derived. From scenario matched tuning – like in case of the acceleration – only available is the variance of the acceleration process. The seed noise variance can be derived by calculating the variance of the solution of the differential equation above.

The complete solution can be written as a superposition of a homogeneous and an inhomogeneous solution according to [38, p. 33] as

$$\begin{aligned} x(t) &= \Phi(t, t_0) \cdot x(t_0) + \int_{t_0}^t \Phi(-\xi, t_0) \cdot q(\xi) d\xi \\ &= e^{-\frac{t}{\tau}} \cdot x(t_0) + \int_{t_0}^t e^{-\frac{t-\xi}{\tau}} \cdot q(\xi) d\xi \end{aligned} \quad (\text{VII-10})$$

It is assumed that $x(t_0) = \mathbf{0}$, which gives a zero homogeneous solution. Also, it is assumed that $t_0 = \mathbf{0}$. The variance of the solution, which is also the variance of the Gauss Markov process, can be written as follows

$$\begin{aligned} \sigma_x^2(t) &= \mathcal{E} \{ x^2(t) \} = \mathcal{E} \left\{ \int_{t_0}^t \int_{t_0}^t e^{-\frac{t-\xi}{\tau}} \cdot q(\xi) \cdot e^{-\frac{t-\chi}{\tau}} \cdot q(\chi) d\xi d\chi \right\} \\ &= \mathcal{E} \left\{ \int_{t_0}^t \int_{t_0}^t e^{-\frac{t-\xi-\chi}{\tau}} \cdot q(\xi) \cdot q(\chi) \cdot \delta(\xi - \chi) d\xi d\chi \right\} \\ &= \int_{t_0}^t e^{-\frac{2\xi}{\tau}} \cdot \mathcal{E} \{ q^2(\xi) \} d\xi = \sigma_q^2 \cdot \int_{t_0}^t e^{-\frac{2\xi}{\tau}} d\xi = \frac{\sigma_q^2 \cdot \tau}{2} \cdot \left(1 - e^{-\frac{2t}{\tau}} \right) \end{aligned} \quad (\text{VII-11})$$

After settling, the steady state gauss Markov process variance can be derived from equation (VII-11). The result can be used to calculate the necessary variance of the seed noise, driving the Gauss Markov process.

$$\sigma_x^2 = \lim_{t \rightarrow \infty} \sigma_x^2(t) = \frac{\sigma_q^2 \cdot \tau}{2} \quad \rightarrow \quad \sigma_q^2 = \frac{\sigma_x^2 \cdot 2}{\tau} \quad (\text{VII-12})$$

For discrete realization in Matlab, the seed noise can be generated, using the following equation.

$$q_k = \frac{\sigma_x^2 \cdot 2}{\tau} \cdot \sqrt{\Delta T} \cdot \text{randn} \quad (\text{VII-13})$$

A detailed derivation of the discrete variance can also be found in [34, p. 142].

As example, the tuning parameters derived in chapter III - 3.4 are used for the DA42 approach.

$$\sigma_x^2 = \sigma_{a,LoS}^2 = 0.79, \quad \tau = \tau_{a,LoS} = 7.083s \quad (\text{VII-14})$$

The simulation of the Gauss Markov process corresponding to the defined parameters can be found in chapter III - 3.4 .

VIII Literature

- [1] B. W. Parkinson and J. J. Spilker Jr., *Global Positioning System: Theory and Applications Volume I*, Washington: American Institute of Aeronautics, 1996.
- [2] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, 2008.
- [3] H. Lin, B. Xu, X. Tang and G. Ou, "Performance Analysis of Signal Tracking Filter and Navigation Filter Tight Integration in GNSS Receiver," *5th International Conference on Computer Science and Network Technology (IEEE)*, 2016.
- [4] E. D. Kaplan, *Understanding GPS Principles and Application*, Norwood: Artech House, 1996.
- [5] K. Borre, *A software defined GPS and Galileo receiver - A single frequency approach*, Birkhäuser, 2007.
- [6] B. N. Vu and M. Andrieu, "The code and carrier tracking loops for GPS signals," *Proceedings of the 16th International Conference on Mechatronics - Mechatronika 2014*, 2014.
- [7] A. Weltri and U. Bernhard, "Third-order delay-locked loop: mean time to lose lock and optimal parameters," *The Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1992.
- [8] P. L. Kazemi, "Optimum digital filters for GNSS tracking loops," *21st Int. Tech. Meeting Satell. Div. Inst. Navig., Savannah, GA*, 2008.
- [9] A. L. Weltri and B.-Z. Bobrovsky, "Mean Time to Lose Lock for a Coherent Second Order PN-Code Tracking Loop - The Singular Perturbation Approach," *IEEE Journal on selected areas in communications*, 1990.
- [10] A. Weltri and B. Bobrovsky, "Optimization of a second order PN-code tracking loop using mean exit time criterion," *IEEE 1989 International Conference on Systems Engineering*, 1989.
- [11] Y. Yang, J. Zhou and O. Loffeld, "GPS Receiver Tracking Loop Design based on a Kalman Filtering Approach," *International Symposium ELMAR*, 2012.
- [12] J.-H. Won, T. Pany and B. Eissfeller, "Characteristics of Kalman Filters for GNSS Signal Tracking Loop," *IEEE Transactions on Aerospace and Electronic Systems*, 2012.
- [13] A. Patapoutian, "On phase-locked loops and Kalman filters," *IEEE Transactions on Communications*, 1999.
- [14] X. Tang, G. Falco, E. Falletti and L. Lo Presti, "Theoretical analysis and tuning criteria of Kalman filter based tracking loop," *Springer-Verlag*, 2014.

- [15] M. L. Psiaki and H. Jung, "Extended Kalman Filter Methods for Tracking Weak GPS Signals," *ION GPS*, pp. 24-27, 2002.
- [16] C. O' Driscoll and G. Lachapelle, "Comparison of Traditional und Kalman Filter Based Tracking Architectures," *European Navigation Convergence*, 2009.
- [17] A. Papatoutian, "On Phase-Locked Loops and Kalman Filters," *IEEE Transactions on Communications*, 1999.
- [18] G. S. Christiansen, "Modeling of a PRML Timing Loop as a Kalman Filter," *IEEE*, 1994.
- [19] M. Lashley, Modeling and performance analysis of GPS vector tracking algorithms, Auburn: ProQuest Dissertations Publishing, 2009.
- [20] M. Lashley, D. M. Bevly and J. Y. Hung, "A valid comparison of vector and scalar tracking loops," *IEEE*, 2010.
- [21] Q. Li, W. Wang, X. Guo and D. Xu, "A design method and performance analysis of vector based tracking loop receiver," *IEEE 11th International Conference on Signal Processing*, 2012.
- [22] J. H. Won, D. Dötterböck and B. Eissfeller, "Performance comparison of different forms of Kalman Filter approaches for a vector based gnss signal tracking loop," *Journal of The Institute of Navigation*, 2010.
- [23] M. Dufour and C. Ouzeau, "Assessment of New Tracking Architectures for Future GNSS Receivers in Harsh Environments".
- [24] M. Xiaoyong and L. Baiqi, "Design of an INS aided high dynamic GPS receiver," *International Convergence on Electronics, Communications and Control (ICECC)*, 2011.
- [25] S. Alban and A. Dietrich, "Inertial aiding of Phase-Tracking Loops for Automotive GPS Attitude Determination," *ION 60th Annual Meeting*, 2004.
- [26] S. Alban, D. M. Akos, S. M. Rock and D. Gebre-Egziabher, "Performance Analysis and Architectures for INS-Aided GPS Tracking Loops".
- [27] P. D. Groves and D. C. Long, "Adaptive tightly-coupled, a low cost alternative anti-jam INS/GPS integration technique," *ION NTM*, 2003.
- [28] M. Lashley, D. M. Bevly and J. Y. Hung, "Analysis of Deeply Integrated and Tightly Coupled Architectures," *IEEE/ION Position, Location and Navigation Symposium*, 2010.
- [29] W. C. a. C. C. M. Lindsey, "A survey of digital phased-locked loops," *Proceedings of the IEEE*, 1981.

- [30] R. De Gaudenzi and M. Luise, "Decision-Directed Coherent Delay-Lock Tracking Loop for DS-Spread-Spectrum Signals," *IEEE Transactions on Communications*, VOL.39, NO. 5, 05 1991.
- [31] J. J. Spilker, "Delay-Lock Tracking of Binary Signals," *IEEE Transactions on Space Electronics and Telemetry*, 1963.
- [32] Lunze, *Regelungstechnik 1*, Springer, 2006.
- [33] B. Braun, Promotion Ben, München, 2016.
- [34] J. A. Farrell, *Aided Navigation*, McGraw-Hill, 2008.
- [35] M. Psiaki and H. Jung, "Extended Kalman Filter Methods for Tracking Weak GPS Signals," *ION GPS*, 2002.
- [36] R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE*, 4 December 1969.
- [37] S. M. Grewal, *Kalman Filtering Theory and Practice using Matlab*, Wiley, 2008.
- [38] P. B. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling*, Wiley, 2010.
- [39] D. Simon, *Optimal State Estimation*, Wiley, 2006.
- [40] R. Brown and P. Hwang, *Introduction to random signals and applied Kalman filtering*, John Wiley, 1992.
- [41] W. Arnold and A. Laub, "Generalized eigenproblem algorithm and software for algebraic riccati equations," *Proceedings of the IEEE*, 72, pp. 1746-1754, 1984.
- [42] H. Abou-Kandil, G. Freiling, V. Ionescu and G. Jank, *Matrix Riccati Equations in Control and System Theory*, Birkhäuser Verlag, 2003.
- [43] J. Guo, J. Ou, Y. Yuan and H. Wang, "Optimal carrier-smoothed-code algorithm for dual frequency GPS data," *Progress in Natural Science*, 2008.
- [44] P. Byungwoon and K. Changdon, "Optimal hatch filter with a flexible smoothing window width," *ION GNSS 18th International Technical Meeting of the Satellite Division*, 2005.
- [45] P. W. Ward, "Performance comparison between FLL, PLL and a Novel FLL-Assisted-PLL Carrier Tracking Loop under RF interference conditions," *Proceedings of the 11th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 1998.
- [46] F. M. G. Sousa, F. D. Nunes and J. Marcal, "Performance of an adaptive partitioned vector tracking algorithm with real scintillation data," *9th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*, 2018.

- [47] P. Bolla and E. S. Lohan, "Dual-frequency signal processing architecture for robust and precise positioning applications," *IEE/ION PLANS*, 2018.
- [48] P. Bolla, J. Nurmi, J.-H. Won and E. S. Lohan, "Joint Tracking of Multiple Frequency Signals from the same GNSS satellite," *International Conference on Localization and GNSS (ICL-GNSS)*, 2018.
- [49] T. Beran, D. Kim and R. B. Lanley, "High-Precision Single-Frequency GPS Point Positioning," *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pp. 1192 - 1200, 09 2003.
- [50] I. Villalon-Turrubiates, O. Ibarra-Manzano and Y. Shmaliy, "Three-dimensional optimal Kalman algorithm for GPS-based positioning estimation of the stationary object," *Proceedings of CAOL'2003. 1st International Conference on Advanced Optoelectronics and Lasers*, 09 2003.
- [51] V. M. Gomes, H. H. Kuga and R. V. Fonseca Lopes, "FILTERING GPS NAVIGATION SOLUTIONS FOR STATIC POSITIONING," *Proceeding of COBEM 2003*, 11 2003.
- [52] B. Hofmann-Wellenhof, H. Lichtenegger and E. Wasle, *GNSS - Global Navigation Satellite Systems*, Wien: Springer, 2008.
- [53] J. Chaffee, J. Abel and B. McQuiston, "GPS positioning, filtering, and integration," *Proceedings of the IEEE 1993 National Aerospace and Electronics Conference-NAECON 1993*, 05 1993.
- [54] Z. Zhenzhen, C. Zhu, T. Guangfu, L. Shufeng and H. Fukan, "EKF based Vector Delay Lock Loop algorithm for GPS signal tracking," *2010 International Conference On Computer Design and Applications*, 06 2010.
- [55] A. Hongfei, L. Chuanjun and A. Qi, "Review on vector tracking application to GNSS receiver," *12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 2015.
- [56] T. Pany and B. Eissfeller, "Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions," *2006 IEEE/ION Position, Location, And Navigation Symposium*, 04 2006.
- [57] E. N. Joshua, D. Yang, T. Jin and S. Z. Farooq, "Design of a Differential Vector Phase Locked Loop for Single Frequency RTK Receivers," *11th International Congress on Image and Signal Processing BioMedical Engineering and Informatics*, 2018.
- [58] R. Di, Y. Morton and S. Peng, "A multiple-frequency GPS software receiver design based on a Vector tracking loop," *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, 2012.
- [59] M. Lashley, D. M. Bevely and J. Y. Hung, "Performance Analysis of Vector Tracking Algorithms for Weak GPS Signals in High Dynamics," *IEEE*, 2009.

- [60] M. Lashley and D. M. Bevly, "Comparison of traditional tracking loops and vector based tracking loops for weak GPS signals," *National Tech. Meeting Inst. of Navigation, San Diego*, 2008.
- [61] G. Liu, R. Zhang, M. Guo and X. Cui, "Accuracy comparison of GNSS vector and scalar tracking loop," *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014.
- [62] F. A. Khan, J. Tian and N. Alam, "Theoretical Performance Analysis and Comparison of VDFLL and Traditional FLL Tracking Loops," *European Navigation Conference*, 2018.
- [63] M. Petovello and G. Lachapelle, "Comparison of vector-based software receiver implementations with application to ultra-tight GPS/INS integration," *Inst. of Nav. GPS/GNSS Conf., Fort Worth, TX*, 2006.
- [64] S. Hongwei, L. Yuli and C. Guangfeng, "Relations between the Standard Variance and the Allan Variance," *2010 International Conference on Computational and Information Sciences*, 2010.
- [65] S. Bhattacharyya and D. Gebre-Egziabher, "Development and validation of parametric models for vector tracking loops," *Journal of The Institute of Navigation*, 2010.
- [66] D. Benson, "Interference Benefits of a Vector Delay Lock Loop GPS Receiver," *Proceedings of the 63rd Annual Meeting of The Institute of Navigation*, 2007.
- [67] D. M. Bevly and M. Lashley, "Analysis of discriminator based vector tracking algorithms," *National Tech. Meeting Inst. of Nav., San*, 2007.
- [68] G.-I. Jee, J.-H. Song and K.-H. Kim, "The GPS vector tracking loop based on the iterated unscented Kalman filter under the large initial error," *European Control Conference (ECC)*, 2009.
- [69] J. Marcal and F. Nunes, "Robust Vector Tracking for GNSS carrier phase signals," *International conference on localization and GNSS (ICL-GNSS)*, 2016.
- [70] S. Alban, D. M. Akos and S. M. Rock, "Performance Analysis and Architectures for INS-Aided GPS Tracking Loops," *Journal of Navigation*, 2003.
- [71] K. Hyun-Soo, B. Sung-Chun, J. Gyu-In and P. Chan-Gook, "An Ultra-tightly coupled GPS/INS Integration using Federated Kalman Filter," *Institute of Navigation*, 2003.
- [72] N. A. Carlson, "Federated filter for distributed navigation and tracking applications," *ION 58th Annual Meeting/CIFGTF 21s Guidance Test Symposium*, June 2002.
- [73] G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering*, Magellan Book Company, 1993.

- [74] Y. Bar-Shalom, M. Mallick, C. Huimin and R. Washburn, "One-Step Solution for the General Out-of-Sequence Measurement Problem in Tracking," *IEEE Aerospace Conference Proceedings*, 2002.
- [75] J. Wendel, *Integrierte Navigationssysteme*, München: Oldenbourg Wissenschaftsverlag GmbH, 2007.
- [76] Larsen and T.D., "Incorporation of time delayed measurements in a discrete-time Kalman filter," *37th Conference on Decision and Control, CDC' 98*, 1998.
- [77] E. Amani, K. Djouani, J.-R. De Boer, A. Kurien, W. Vigneau and T. Junique, "Adaptive and Conjoint Scalar-Vector Tracking Loops for GNSS Tracking Robustness and Positioning Integrity," *European Navigation Conference*, 2017.
- [78] I. Groh, "Analytical computation of mean time to lose lock for langevin delay locked loops," *IEEE Transactions on Communications*, 11 2012.
- [79] W. Gurtner and L. Estey, "The Receiver Independent Exchange Format," July 14th, 2015.
- [80] L. Görcke, J. Dambeck and F. Holzapfel, "Aerodynamic flight simulation in inertial quality," *Proceedings of the ION International Technical Meeting*, 2013.
- [81] P. Misra, *Global Positioning System*.
- [82] A. V. Dierendonck, "Theory and Performance of Narrow Correlator Spacing in a GPS Receiver," *ION*, 1992.
- [83] S. Bhattacharyya, "Vector Loop Transfer Functions and Noise Bandwidths," *NAVIGATION, Journal of The Institute of Navigation*, Vol. 65, No. 1, 2018.
- [84] Lunze, *Regelungstechnik 2*.
- [85] K. Dietmayer, M. Saad, C. Strobel, D. F. Garzia, M. Overbeck and D. W. Felber, "Real time implementation of Vector Delay Lock Loop on a GNSS receiver hardware with an open software interface," *European Navigation Conference (ENC) IEEE*, 2019.
- [86] T. Pany, *Navigation Signal Processing for GNSS Software receivers*, Artech House, 2010.
- [87] D. Borio, H. Kuusniemi and L. Lo Presti, "Impact and Detection of GNSS jammers on consumer grade satellite navigation receivers," *Proceedings of the IEEE*, June 2016.
- [88] R. H. Mitch, R. C. Dougherty, M. L. Psiaki and S. P. Powell, "Signal characteristics of civil GPS jammers," *Journal of the institute of navigation*, pp. 20-23, 2011.

- [89] T. Kraus, R. Bauernfeind and B. Eissfeller, "Survey of In-Car Jammers - Analysis and Modeling of the RF signals and IF samples," *Journal of the institute of navigation*, pp. 20-23, September 2014.
- [90] D. Borio, C. O'Driscoll and J. Fortuny, "GNSS Jammers: Effects and countermeasures," *6th ESA Workshop on Satellite Navigation Technologies*, 2012.
- [91] D. Borio, C. O'Driscoll and J. Fortuny, "Jammer impact on Galileo and GPS receivers," *International Conference on Localization and GNSS*, 2013.
- [92] J. W. Betz, "Effect of Narrowband Interference on GPS Code Tracking Accuracy," *Proceedings of the 2000 National Technical Meeting of The Institute of Navigation*, 2000.
- [93] T. Kraus, "Low Budget GNSS Jammer," TU Mnchen, Kolloquium Satellitennavigation, 2012.
- [94] R. H. Mitch, R. C. Dougherty, M. L. Psiaki, S. P. Powell and B. W. O'Hanlon, "Know Your Enemy: Signal Characteristics of Civil GPS Jammers," *GPS World Vol. 24, No. 1*, 2012.
- [95] J. W. Betz, "Effect of Partial-Band Interference on Receiver Estimation of C/N0: Theory," *Proceedings of the 2001 National Technical Meeting of the Institute of Navigation*, 2001.
- [96] N. Kubo, M. Lu, M. Sgammini and G. Xingxin, "Protecting GNSS Receivers From Jamming and Interference," *Proceedings of the IEEE*, 2016.
- [97] L. D. F. C. J. T. Musumeci, "A Comparative Analysis of Adaptive Notch Filtering and Wavelet Mitigation against Jammers Interference," *NAVIGATION, Journal of The Institute of Navigation, Vol. 63, No. 4*, 2016.
- [98] M. S. Braasch and A. J. V. Van Dierendonck, "GPS Receiver Architectures and Measurements," *Proceedings of the IEEE*, 01 1999.
- [99] J. W. Betz, "Effect of Narrowband Interference on GPS Code Tracking Accuracy," *ION NTM*, 2000.
- [100] K. R. Kolodziejcki and J. W. Betz, "Effect of Non-White Gaussian Interference on GPS Code Tracking Accuracy," *The MITRE Corporation Technical Report MTR99B21R1*, 06 1999.
- [101] J. W. Betz and K. R. Kolodziejcki, "Generalized Theory of GPS Code Tracking Accuracy with an Early-Late Discriminator," *IEEE Transactions on Aerospace and Electronic Systems*, 2009.
- [102] G. W. Hein, J.-A. Avila-Rodriguez, S. Wallner, A. R. Pratt, J. Owen and J.-L. Issler, "MBOC: The new optimized spreading modulation recommended for Galileo L1 OS and GPS L1C," *IEEE Conference Paper*, 2006.
- [103] A. Van Dierendonck, "Theory and Performance of Narrow Correlator Spacing in a GPS Receiver," *Journal of The Institute of Navigation*, 1992.

- [104] I. Groh, C. Gentner and J. Selva, "Analytical Computation of Mean Time to Lose Lock for Langevin Delay-Locked Loops," *IEEE Transactions on Communications*, Volume: 60, Issue: 11, 2012.
- [105] D. M. Pozar, *Microwave Engineering*, John Wiley & Sons, 1998.
- [106] J. Detlefsen and U. Siart, *Grundlagen der Hochfrequenztechnik*, Oldenbourg, 2006.
- [107] H. L. Van Trees, *Optimum Array Processing*, John Wiley & Sons, Inc., 2002.
- [108] J. E. Hudson, *Adaptive Array Principles*, IEEE Electromagnetic Waves Series 11, London: Peter Peregrinus Ltd., 1989.
- [109] R. J. Mailloux, *Phased Array Antenna Handbook*, Boston: Artech House, 1994.
- [110] M. A. Richards, *Fundamentals of Radar Signal Processing*, McGraw-Hill, 2005.
- [111] C. A. Balanis, *Antenna Theory: analysis and Design*, Wiley, 1982.
- [112] G. F. Hatke, "Adaptive array processing for wideband nulling in GPS systems," *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers*, 1998.
- [113] L. E. Brennan and L. S. Reed, "Theory of adaptive radar," *IEEE Transactions on aerospace and electronic systems*, 1972.
- [114] Z. Hongwei and L. Boawang, "Space-Time Adaptive Processing for GPS anti-jamming Receiver," *Physics Procedia*, 2012.
- [115] M.-D. Nguyen and H.-T. Nguyen, "A Novel Multi-beamforming Multi-channel GNSS Receiver Design for Interference Mitigation," *International Conference on Advanced Technologies for Communications*, 2014.
- [116] S. R. J. C. M. Gunawardena, "Correlator beamforming for multipath mitigation in high-fidelity GNSS monitoring applications," *NAVIGATION, Journal of The Institute of Navigation*, Vol. 66, No. 1, 2019.
- [117] F. Shen, F. Chen and J. Song, "Robust Adaptive Beamforming Based on Steering Vector Estimation and Covariance Matrix Reconstruction," *IEEE Communications Letters*, 2015.
- [118] V. Carotenuto, C. Hao, D. Orlando, A. De Maio and S. Iommelli, "Detection of Multiple Noise-like Jammers for Radar Applications," *5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2018.
- [119] R. T. Compton, "The power inversion adaptive array: Concept and performance," *IEEE Transactions on aerospace and electronic systems*, 1979.

- [120] A. Gecan and M. Zoltowski, "Power minimization techniques for GPS null steering antenna," *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 1995.
- [121] D. Meng, Z. Feng and M. Lu, "Anti-Jamming with adaptive arrays utilizing power inversion algorithm," *Tsinghua science and technology*, Volume 13, 2008.
- [122] H. Liu, A. Ghafoor and P. H. Stockmann, "Application of Gram-Schmidt Algorithm to Fully Adaptive Arrays," *IEEE*, 1991.
- [123] T. Gao, "Pattern nulling synthesis based on adaptive array theory".*IEEE*.
- [124] Meyberg and Vachenauer, *Höhere Mathematik 1*, Springer, 2003.
- [125] J. T. Mayhan, "Nulling limitations for a multiple-beam antenna," *IEEE Trans. Antenna and Propagation*, vol. AP-24, 1976.
- [126] B. Z. Bobrowsky and A. Welti, "Mean time to lose lock for the "Langevin"-type delay-locked loop," *IEEE Trans. Commun.*, vol. 42, 1994.