TECHNISCHE UNIVERSITÄT MÜNCHEN
Fachgebiet Methoden der Signalverarbeitung

Kernel Methods for ill-posed localization problems

Dipl.-Tech. Math. Univ. Daniel Kotzor

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universitat München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. rer. nat. Martin Kleinsteuber

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Wolfgang Utschick
2. Univ.-Prof. Dr.-Ing. Klaus Diepold

Die Dissertation wurde am 17.11.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 11.11.2011 angenommen.

Contents

RKHS	Reproducing Kernel Hilbert Space	8
AOA	Angle Of Arrival	11
GNNS	Global Navigation Satellite System	11
GPS	Global Positioning System	11
IR	Infrared Sensor	11
RFID	Radio-Frequency IDentification	11
RSS	Received Signal Strength	11
WLAN	Wireless Local Area Network	11
TDOA	Time Difference Of Arrival	12
TOA	Time Of Arrival	12
UWB	Ultra Wide Band	12
ML	Maximum Likelihood	14
PDF	Probability Density Function	17
CRLB	Cramér Rao Lower Bound	18
GDOP	Geometric Dilution Of Precision	19
HDOP	Horizontal Dilution of Precision	19
PDOP	Positional Dilution of Precision	19
VDOP	Vertical Dilution Of Precision	19
IMU	Inertial Measurement Unit	23
INS	Inertial Navigation System	23
IFOG	Interferometric Fibre-Optic Gyro	27
RLG	Ring Laser Gyro	27
MEMS	Micro-Electro-Mechanical Systems	28
MAP	Maximum A Posteriori	30
EKF	Extended Kalman Filter	32
PF	Particle Filter	32
SMC	Sequential Monte Carlo	33
UKF	Unscented Kalman Filter	32
ROLAM	Range Only Localization And Mapping	35
SLAM	Simultaneous Localization And Mapping	39
LaSLAT	Simultaneous Localization And Tracking	44
SVN	Support Vector Machines	52
PSD	Power Spectral Density	62
LTI	Linear Time Invariant	64
SDE	Stochastic Differential Equation	68
MC	Monte Carlo	89
RMS	Root Mean Square	113
LOS	Line Of Sight	120

1. Introduction

Autonomous systems that are able to change their behavior in response to unanticipated events during operation are of great research interest at present and for the next decades. In recent years the capabilities of such systems and their domains of application have expanded significantly with first successes in both civilian and military applications. Although the technology still is in its fledgling stage one can say that it holds an immense potential for enabling entirely new functionalities in environments where direct human control is not physically possible. While autonomous systems may vary widely in their capabilities, purposes and levels of autonomy [1, 2, 3, 4], they all require the basic ability of navigation. Consequently, self localization represents one of the essential key-technologies to put autonomous systems into practice.

While satellite based localization technologies like the *Global Positioning System* (GPS) work well with an accuracy that is sufficient for most outdoor applications, the GPS signal is too weak to penetrate most buildings so that GPS based indoor localization persists intractable for a lot of applications with regard to accuracy, availability and reliability. At present a lot of effort is made in developing alternative techniques to expand the range of service from outdoor to indoor. To achieve fast deployment there is also a strong tendency towards self adjusting systems that simultaneously localize and map the environment at the same time. Within indoor environment however, reliable sources of information are rare.

Motivated by an ill-posed range only localization and mapping problem based on a sparse amount of data, this thesis presents an approach to fill the information gap in form of a novel dynamic model for arbitrary robot movement. The proposed model design is based on scientific insights from the fields of *stochastic processes* and *kernel regression*. The model can be seamless incorporated into the usual localization framework and substitute or support control information. This is demonstrated for the problem at hand. However, the final methodology is generic enough to be applied on a wide range of ill-posed localization problems.

The thesis is organized as follows. Chapter 2 starts by introducing general approaches to localization problems and sets the basement for the main insights of the following chapters. Chapter 3 focuses on the range based localization and mapping problem that is going to be solved within this thesis. The definition of the problem and an overview of state-of-the-art approaches reveals the need for

additional scientific work and motivates this thesis. Chapter 4 thoroughly derives the framework of kernel regression with respect to the tackled localization problem. As a consequence the space of possible solutions is represented by a function space called *Reproducing Kernel Hilbert Space* (RKHS). The solution to the ill-posed problem is then achieved by balancing data compliance and smoothness property with respect to the norm defined within the RKHS. As the design of the kernel function has a major impact on the quality of the solution this topic marks the main part of this chapter. Chapter 5 reveals practical aspects of the proposed method. Two procedures are presented that make the approach useable for real life applications. An offline procedure that allows to obtain an initial solution to the problem and an online procedure for continuous estimation in real time. Chapter 6 presents results based on multiple synthetic and real life tests. The thesis closes in Chapter 7 with a final discussion.

2. Localization principles

This chapter presents a generic view on localization problems with a focus on problems and approaches that can be related to the problem of range only localization and mapping tackled in this thesis. The concepts and the notation introduced in this chapter is referenced thereinafter. It introduces absolute, relative and hybrid navigation methodologies and their technologies. This chapter allows to classify the localization problem that motivates this thesis.

2.1. Formulation as state observer

The ultimate goal of localization in general is to determine a position or pose of an object in relation to some relative or absolute coordinate system. As a common standard the localization problem can be mathematically formulated in form of a time invariant dynamic system [5, 6]

$$\dot{\boldsymbol{s}} = \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{u}) \tag{2.1}$$

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{u}). \tag{2.2}$$

Adopted to localization problems Eq. (2.1) comprises the equations of motion for the object in form of a differential equation. Here $\boldsymbol{s}$ denotes the state vector of the dynamic model that also covers the information on the desired pose of the object. $\boldsymbol{u}$ consists of all control variables that influence the system in a projected way like a steering parameter for a vehicle or also in an unintentional way like crosswinds for an airplane. Eq. (2.2) denotes the observation model with $\boldsymbol{z}$ as the observation vector. Observations may be absolute coordinates that arrive periodically like a fully processed position update or only consist of fractional information like range information to an object of known position. The explicit formulation of a localization problem as a system strongly depends on the available data. Without any information on the control data $\boldsymbol{u}$ only the observation model can be used for localization. This is practiced for example by using a third party system in a car that only has *Global Positioning System* (GPS) data for disposal. In contrast, build-in systems for cars often make use of additional information like the steering angle and car velocity to improve the accuracy. Although in most cases the information on control data can greatly improve the localization process, this information can be quite costly to acquire and to process.

For most real world applications, the data can not be obtained continuously but only discrete. Therefore, Eq. (2.1)) and (Eq. (2.2) are mostly used in their discrete formulations

$$\boldsymbol{s}_{i+1} = \boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{u}_i) \tag{2.3}$$

$$\boldsymbol{z}_{i+1} = \boldsymbol{h}(\boldsymbol{s}_i, \boldsymbol{u}_i). \tag{2.4}$$

The discrete representation is used throughout the thesis. In general, localization methods can be roughly divided into the following three groups:

- Absolute localization methods that solve Eq. (2.4).
- Relative localization methods that solve Eq. (2.3).
- Hybrid localization methods that combine absolute and relative localization.

All three localization methods are summarized in the following section.

2.2. Absolute localization

In order to determine a position by using Eq. (2.4), one or multiple observations with respect to some coordinate system are required. Therefore, absolute navigation techniques often require specific infrastructure or presume a priori knowledge on the environment. Because this is strongly related with the used technologies Sec. 2.2.1 gives a short introduction of the most relevant ones. Sec. 2.2.2 focuses on range based localization as this thesis is motivated by a range based localization and mapping problem. Theoretical bounds on the accuracy for standard range based localization problems are given in Sec. 2.2.3.

2.2.1. Technologies

This section briefly introduces a taxonomy, the relvant signal metrics and a short overview of sensor accuracy for absolute positioning.

Taxonomy

In [7] the following taxonomy of absolute localization techniques is proposed. Three general categories are distinguished.

- Proximity based methods,
- Fingerprinting and
- Triangulation.

As the wording suggests, *proximity* techniques use proximity information to localize an object of unknown position. If a signal is received at several known locations, it is possible to intersect the coverage areas of that signal to determine a containing location area. Typical sensors used for this kind of localization are *Infrared Sensors* (IR) that use short range transmission of modulated IR light to transmit the identity of a mobile device to a fixed receiver in a particular known location. A receiver is typically placed in every location in which a mobile device might be found. Another example of proximity sensors are *Radio-Frequency IDentification* (RFID) tags applied to or incorporated into a product, animal, or person for the purpose of identification and tracking by using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader.

Fingerprinting is a technique that examines a scene such as a room from a certain vantage point and searches for unique features (fingerprints) that allows to localize the object. Quite often, indoor positioning systems utilize the location fingerprinting method by comparing signal strengths from all access points to entries in a table that was generated offline. The closest entry in the table is the most probable location of the user. The higher the resolution of the map, the better the accuracy of the result. This method can be found for various localization systems in the *Wireless Local Area Network* (WLAN) area, as for example in [8, 9, 10]. One big advantage of this method is that it is not model-based and thus can be used where disturbances generated through geometric constraints anticipate the use of *triangulation* for instance. The disadvantage is the static setup and relatively high investment by the need to generate a map.

As already mentioned the third method is *triangulation*. The method derives its name from the trigonometric calculations that have to be performed in order to get a solution. A typical source of information for triangulation are cell-networks with active beacons. A set of beacons at known positions are gathering geometrical information from a target of unknown position. By resolving Eq. (2.4) it is possible to determine the position of the target. Probably the most prominent occurrence of this kind is the *Global Positioning System* (GPS) which is a *Global Navigation Satellite System* (GNNS). Originally developed by the United States Department of Defense it is today also used by civilians for navigation purposes. It uses a constellation of between 24 and 32 satellites in medium earth orbit that transmit precise radiowave signals which allow GPS receivers to determine their current location, the time, and their velocity. For cell-network localization the typical signal metrices can be typically divided into four groups [7].

- *Angle Of Arrival* (AOA): measures direction in form of angles
- *Received Signal Strength* (RSS): measures ranges based on received signal strengths

- *Time Of Arrival* (TOA): measures ranges based on signal propagation delay
- *Time Difference Of Arrival* (TDOA): measures range differences based on signal propagation delay

Depending on the field of application, all metrics can be found in practice. AOA combined with TDOA for example is used for the *Ultra Wide Band* (UWB) based localization system from Ubisense [11]. GPS for example is based on an augmented form of TOA. Instead of using true range measurements, GPS is based on so called pseudoranges that also incorporate clock errors [12]. RSS is often utilized for indoor localization systems based on WLAN [13, 14]. The biggest problem for active cell network localization is the problem of finding an accurate model for the signal propagation. Multipath effects and damping of the signal energy through walls and other materials often renders the problem infeasible for a lot of applications. For the following sections, trilateration is the relevant technique.

Accuracy

In an outdoor environment, GPS works extremely well with an accuracy that is sufficient for most outdoor applications. Unfortunately, the signal from the GPS satellites is too weak to penetrate most buildings, making indoor GPS localization almost impossible. Likewise the requirements for indoor localization are much more restrictive for precise localization of autonomous vehicles. Therefore a lot of alternative technologies for indoor localization have been envisioned.

The chart in Fig. 2.1, adopted from [15], shows different sensor technologies used for localization. They are classified by achievable accuracy and degree of development.

2.2.2. Range based localization

After introducing the technologies for absolute localization this section focuses on the special case of range based localization. The issue of technical implementation and how to obtain such measures is not broached further. As seen in the former section a lot of cell network based localization systems rely on range information.

Definition

For a scenario with k fixed beacons, let

$$\boldsymbol{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$$

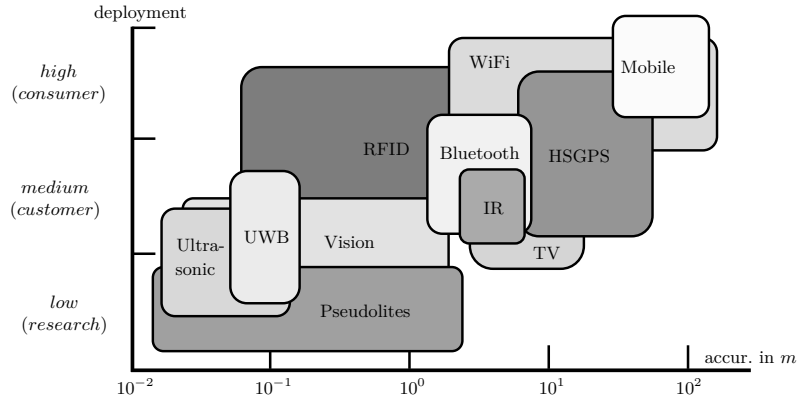

Figure 2.1.: Indoor localization accuracy

denote the seeked position of the target and

$$\boldsymbol{p}_j \triangleq \begin{bmatrix} p_{1,j} \\ p_{2,j} \\ p_{3,j} \end{bmatrix} \in \mathbb{R}^3, \quad 1 \leq j \leq k$$

known positions of the beacons. Let further $r_1, ..., r_n$ denote all available range measurements for the localization process. To cover multiple range measurements to the same beacons the map

$$a : \{1, ..., n\} \to \{1, ..., k\}, \quad i \mapsto a_i$$

is introduced that covers the data to measurement association such that the range measurement can be modeled as

$$r_i \triangleq ||\boldsymbol{x} - \boldsymbol{p}_{a_i}|| + \eta_i, \quad 1 \leq i \leq n. \tag{2.5}$$

This way, the measurement with index i is associated to the according beacon by $a_i \in \{1, ..., k\}$ that allows a unique identification. In Eq. (2.5) η_i denotes an additive noise that corrupts the measurement. It is further assumed that the noise is mutually independent

$$\mathrm{p}(\eta_i, \eta_j) = \mathrm{p}(\eta_i) \cdot \mathrm{p}(\eta_j), \quad \forall i \neq j. \tag{2.6}$$

With these prerequisites the goal is to estimate the targets position $\boldsymbol{x}$ using the measurements r_i to beacons at the positions $\boldsymbol{p}_{a_i}$. Fig. 2.2 illustrates the setup for this typical scenario.

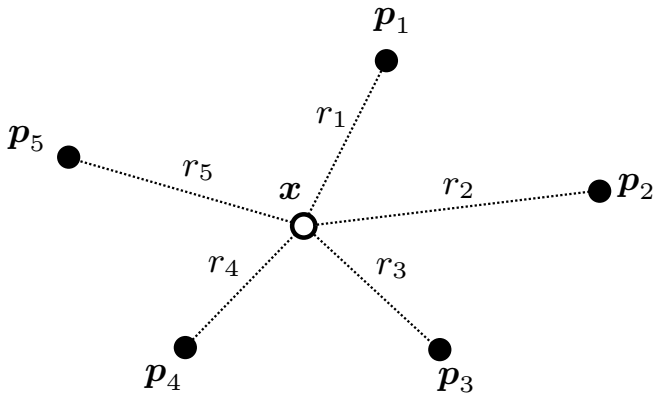

Figure 2.2.: Beacons at positions $\boldsymbol{p}_j \in \mathbb{R}^3$ and target at position $\boldsymbol{x} \in \mathbb{R}^3$

Methods for range based localization

In order to estimate $\boldsymbol{x}$ several methods have been proposed so far. If the noise η_i remains small a common approach called *trilateration* can be applied. This method offers a direct formula for exactly three measures to three beacons [16]. Ambiguities in the method are dealt with additional information like another distance measure or the knowledge that the targets position has a positive z coordinate.

Multilateration is a straightforward method based on range difference measurements to a pre-selected but arbitrary reference point [17]. Because one measurement becomes the the reference, $n-1$ linear equations result for n available measurements. The choice of the reference measurement may greatly influence the result.

A more typical method is to determine the *Maximum Likelihood* (ML) estimator for the position of the target $\boldsymbol{x}$.

Let

$$r^{\langle n \rangle} \triangleq \{r_1, ..., r_n\}$$

be the set of n range measurements,

$$a^{\langle n \rangle} \triangleq \{a_1, ..., a_n\}$$

the set of all data associations and

$$\boldsymbol{p}^{\langle k \rangle} \triangleq \{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\}$$

the set of k beacon positions. The ML-estimator determines the parameter vector $\boldsymbol{x}$ which is most likely to have caused the measurements, i.e.

$$[\boldsymbol{x}]_{\mathrm{ML}} = \arg\max_{\boldsymbol{x}} \mathrm{p}\left(r^{\langle n\rangle}|\boldsymbol{x}, \boldsymbol{p}^{\langle k\rangle}, a^{\langle n\rangle}\right).$$

As the measurement error is assumed to be mutually independent a change in random variables gives way to the simplification

$$[\boldsymbol{x}]_{\mathrm{ML}} = \arg\max_{\boldsymbol{x}} \prod_{i=1}^{n} \mathrm{p}\left(\eta_i\right) \tag{2.7}$$

because

$$\mathrm{p}_{R_i}\left(r_i|\boldsymbol{x}, \boldsymbol{p}_{a_i}, a_i\right) = \mathrm{p}_{E_i}\left(\eta_i\right)^{1}.$$

For normally distributed noise

$$p(\eta_i) \triangleq \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(||\boldsymbol{x}-\boldsymbol{p}_{a_i}||-r_i)^2}{2\sigma_i^2}}, \quad 1 \leq i \leq n \tag{2.8}$$

the negative log-likelihood becomes

$$\begin{aligned} \ell(\boldsymbol{x}) &\triangleq -\log(\mathrm{p}\left(r^{\langle n\rangle}|\boldsymbol{x}, \boldsymbol{p}^{\langle k\rangle}, a^{\langle n\rangle}\right)) = \\ &= \frac{1}{2}\sum_{i=1}^{n} \log(2\pi\sigma_i^2) + \frac{1}{2}\sum_{i=1}^{n} \frac{1}{\sigma_i^2}(||\boldsymbol{x} - \boldsymbol{p}_{a_i}|| - r_i)^2. \end{aligned} \tag{2.9}$$

The first term of Eq. (2.9) is a constant, therefore it is straightforward to show that the ML-estimator is the minimizer of

$$\breve{\boldsymbol{x}} = [\boldsymbol{x}]_{\mathrm{ML}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{n} \frac{\Delta r_i(\boldsymbol{x})^2}{\sigma_i^2},$$

with the definition

$$\Delta r_i(\boldsymbol{x}) \triangleq ||\boldsymbol{x} - \boldsymbol{p}_{a_i}|| - r_i.$$

A standard approach for this non linear least squares problem is the *Gauss-Newton algorithm* that is briefly described in this context [18, p. 834]. For the unconstrained minimization problem, each component of the gradient has to be zero at the optimum

$$\nabla\ell(\breve{\boldsymbol{x}}) = 0, \tag{2.10}$$

[1] The change in random variables here is marked by the subscripts R_i for a random variable for the range r_i and E_i for a random variable for the error η_i. For convenience this notation is neglected when the definition of the respective random is clear from the context.

with $\check{\boldsymbol{x}}$ denoting the optimal vector. Deriving the Jacobian $\boldsymbol{J}$ by

$$J_{i1} = \frac{\partial \Delta r_i(\boldsymbol{x})}{\partial x} = \frac{x_1 - p_{1,a_i}}{||\boldsymbol{x} - \boldsymbol{p}_{a_i}||}$$
$$J_{i2} = \frac{\partial \Delta r_i(\boldsymbol{x})}{\partial y} = \frac{x_2 - p_{2,a_i}}{||\boldsymbol{x} - \boldsymbol{p}_{a_i}||}$$
$$J_{i3} = \frac{\partial \Delta r_i(\boldsymbol{x})}{\partial z} = \frac{x_3 - p_{3,a_i}}{||\boldsymbol{x} - \boldsymbol{p}_{a_i}||},$$

with $1 \leq i \leq n$ and defining a diagonal weighting matrix $\boldsymbol{W}$ with the elements

$$W_{ii} \triangleq \frac{1}{\sigma_i^2},$$

the gradient writes as

$$\nabla \ell(\boldsymbol{x}) = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \Delta \boldsymbol{r}(\boldsymbol{x}), \tag{2.11}$$

with the definition

$$\Delta \boldsymbol{r}(\boldsymbol{x}) \triangleq (\Delta r_1(\boldsymbol{x}), ..., \Delta r_n(\boldsymbol{x}))^{\mathrm{T}}. \tag{2.12}$$

The Taylor series expansion of Eq. (2.12) to the first order equations yields

$$\Delta \boldsymbol{r}(\check{\boldsymbol{x}}) \approx \Delta \boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{J}(\check{\boldsymbol{x}} - \boldsymbol{x}). \tag{2.13}$$

With Eq. (2.10) and Eq. (2.13) Eq. (2.11) becomes

$$\nabla \ell(\check{\boldsymbol{x}}) = 0 \approx \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \Delta \boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J}(\check{\boldsymbol{x}} - \boldsymbol{x}).$$

As long as $\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J}$ is invertible $\check{\boldsymbol{x}}$ can be approximated through

$$\check{\boldsymbol{x}} \approx \Phi(\boldsymbol{x}) \triangleq \boldsymbol{x} - (\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \Delta \boldsymbol{r}(\boldsymbol{x}) \tag{2.14}$$

and due to

$$\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \Delta \boldsymbol{r}(\check{\boldsymbol{x}}) = 0$$

it is necessary that

$$\check{\boldsymbol{x}} = \Phi(\check{\boldsymbol{x}}).$$

The problem can be interpreted of finding a fixed point for $\Phi(\boldsymbol{x})$. If $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ is a contraction mapping with the property that there is some real number $0 < \lambda < 1$, called the Lipschitz constant, such that for all $\boldsymbol{x}_1 \in \mathbb{R}^3$ and $\boldsymbol{x}_2 \in \mathbb{R}^3$

$$||\Phi(\boldsymbol{x}_1) - \Phi(\boldsymbol{x}_2)|| \leq \lambda ||\boldsymbol{x}_1 - \boldsymbol{x}_2||$$

the Banach fixed point theorem states that the iterated function sequence

$$\boldsymbol{x}, \Phi(\boldsymbol{x}), \Phi(\Phi(\boldsymbol{x})), \Phi(\Phi(\Phi(\boldsymbol{x}))), ...$$

converges to the fixed point $\check{\boldsymbol{x}}$ [18, p. 562]. $\check{\boldsymbol{x}}$ can be therefore calculated by iteratively evaluating $\boldsymbol{\Phi}$ and starting with an initial guess $\boldsymbol{x}_1$,

$$\boldsymbol{x}_{i+1} = \boldsymbol{\Phi}(\boldsymbol{x}_i).$$

Multiple stopping criteria for the iteration procedure can be defined. One commonly used criteria is to iterate until the square root of the residual becomes very small

$$||\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|| < \epsilon.$$

Although the rate of convergence of the Gauss-Newton algorithm can approach a quadratic fashion, the algorithm may converge slowly or not at all if the initial guess is far from the minimum or the matrix $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{J}$ is ill-conditioned [19]. There also exist more refined procedures concerning convergence and stability like *Hartley's Modification* and the *Levenberg-Marquardt* algorithm [20, 21]. Nevertheless, for all these methods a good starting value $\boldsymbol{x}_1$ has to be chosen. While the solution to trilateration for 3 out of the k beacons delivers an accurate staring value, there exists an even faster method called *Min-Max algorithm* that takes all measurements into consideration. The Min-Max method presented by Langendoen et al. [17] and Savvides et al. [22] offers significantly less computational burden than multilateration.

2.2.3. Theoretical bounds on the accuracy

All methods can only estimate the true position because the range measurements usually are corrupted by measurement errors. In order to investigate the quality of these estimators it is essential to deal with the so called Cramér Rao bound.

The Cramér Rao lower bound

Harald Cramér and Calyampudi Radhakrishna Rao derived a lower bound on the variance of estimators of a deterministic parameter[23, 24]. The bound is also known as the information inequality.
Let $\boldsymbol{T}(\boldsymbol{z})$ represent an unbiased estimate for the unknown parameter vector $\boldsymbol{x}$ based on observations covered by the random vector $\boldsymbol{z} \triangleq [z_1 \cdots z_n]^{\mathrm{T}}$ under the joint *Probability Density Function* (PDF) denoted by $\mathrm{p}(\boldsymbol{z}|\boldsymbol{x})$. Then

$$\mathrm{Var}[\boldsymbol{T}(\boldsymbol{z})] \geq \mathrm{E}\left[(\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}(\boldsymbol{z}|\boldsymbol{x}))(\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}(\boldsymbol{z}|\boldsymbol{x}))^{\mathrm{T}}\right]^{-1}, \tag{2.15}$$

provided the following regularity conditions are satisfied:

$$\frac{\partial}{\partial \boldsymbol{x}} \int \mathrm{p}(\boldsymbol{z}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{z} = \int \frac{\partial \mathrm{p}(\boldsymbol{z}|\boldsymbol{x})}{\partial \boldsymbol{x}} \mathrm{d}\boldsymbol{z} = 0$$

$$\frac{\partial}{\partial \boldsymbol{x}} \int T(\boldsymbol{z}) \mathrm{p}\left(\boldsymbol{z}|\boldsymbol{x}\right) \mathrm{d}\boldsymbol{z} = \int T(\boldsymbol{z}) \frac{\partial \mathrm{p}\left(\boldsymbol{z}|\boldsymbol{x}\right)}{\partial \boldsymbol{x}} \mathrm{d}\boldsymbol{z}.$$

Here, the integrals represent n-fold integration [25]. The denominator of Eq. (2.15) is also called the *Fisher Information*

$$I(\boldsymbol{x}) = \mathrm{E}\left[\left(\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}\left(\boldsymbol{z}|\boldsymbol{x}\right) \right) \left(\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}\left(\boldsymbol{z}|\boldsymbol{x}\right) \right)^{\mathrm{T}} \right].$$

Application of the Cramér Rao bound to range based localization

For range based localization problem the Cramér Rao bound can be applied as follows. The gradient of the likelihood function from Eq. (2.11) can be reused to express the Fisher Information for the range based problem

$$I(\boldsymbol{x}) = \mathrm{E}\left[\nabla \ell(\boldsymbol{x}) \left(\nabla \ell(\boldsymbol{x}) \right)^{\mathrm{T}} \right].$$

With the same definitions for $\boldsymbol{J}$, $\boldsymbol{W}$ and $\Delta \boldsymbol{r}(\boldsymbol{x})$ the Fisher Information matrix becomes

$$I(\boldsymbol{x}) = \mathrm{E}\left[\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \Delta \boldsymbol{r}(\boldsymbol{x}) \Delta \boldsymbol{r}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J} \right] = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \mathrm{E}\left[\Delta \boldsymbol{r}(\boldsymbol{x}) \Delta \boldsymbol{r}(\boldsymbol{x})^{\mathrm{T}} \right] \boldsymbol{W} \boldsymbol{J}.$$

Due to

$$\mathrm{E}[\Delta r_i(\boldsymbol{x}) \Delta r_j(\boldsymbol{x})] = \mathrm{E}[\Delta r_i(\boldsymbol{x})] \, \mathrm{E}[\Delta r_j(\boldsymbol{x})] = 0 \quad i \neq j$$

and

$$\mathrm{E}\left[\Delta r_i(\boldsymbol{x})^2 \right] = \sigma_i^2$$

the Fisher information matrix can be finally determined to

$$I(\boldsymbol{x}) = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{W}^{-1} \boldsymbol{W} \boldsymbol{J} = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J}.$$

Thus any estimator $\check{\boldsymbol{x}} = T(r^{\langle n \rangle})$ of the range based localization problem of the kind introduced in Sec. 2.2.2 has a covariance with the following property

$$\boldsymbol{v}^{\mathrm{T}} \left[\mathrm{Var}[\check{\boldsymbol{x}}] - \left(\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J} \right)^{-1} \right] \boldsymbol{v} \geq 0 \quad \forall \boldsymbol{v} \in \mathbb{R}^3. \tag{2.16}$$

By applying Eq. (2.14), $\mathrm{Var}[\check{\boldsymbol{x}}] = \mathrm{Var}[\Phi(\check{\boldsymbol{x}})] = \left(\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J} \right)^{-1}$ which equals the *Cramér Rao Lower Bound* (CRLB). This approach can therefore be considered optimal in the sense of variance minimization under the conditions from Eq. (2.8) and Eq. (2.6) stipulated in section Sec. 2.2.2.

By Eq. (2.16), the accuracy of the estimator is mainly influenced by two effects. The first is described by the noise matrix $\boldsymbol{W}$ for the measurements. The possible accuracy dilutes for greater variances in range errors as depicted schematically in Fig. 2.3

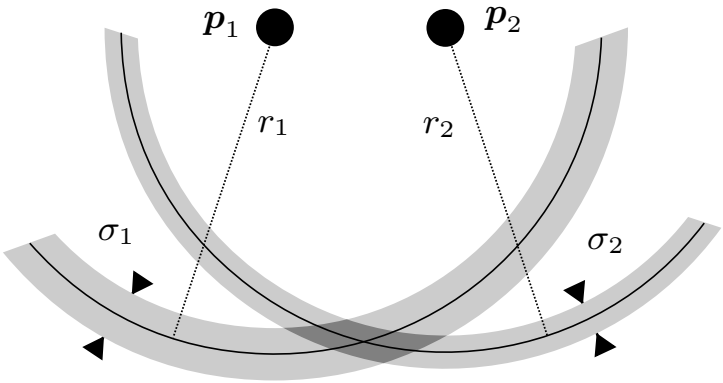

Figure 2.3.: Dilution of precision for range based localization systems

The Geometric Dilution of Precision

The second effect deals with the influence of the geometry. For evaluating geometries for range based localization systems an approach called *Geometric Dilution Of Precision* (GDOP) is widely spread in the community of GNNS localization [26]. By assuming equal variances $\sigma_i = \sigma$ for all measurements the range errors and geometric influences can be separated. Consequently,

$$\mathrm{Var}[\check{\boldsymbol{x}}] = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix} \geq \sigma^2 \cdot (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}.$$

With the definition

$$\begin{bmatrix} d_x^2 & d_{xy} & d_{xz} \\ d_{yx} & d_y^2 & d_{yz} \\ d_{zx} & d_{zy} & d_z^2 \end{bmatrix} \triangleq (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1},$$

the concept of dilution of precision can be transferred for pure range based tracking applications. The DOP factors can be divided in *Positional Dilution of Precision* (PDOP), *Horizontal Dilution of Precision* (HDOP) and *Vertical Dilution Of Precision* (VDOP). These values are defined through

$$\begin{aligned} \mathrm{PDOP} &\triangleq \sqrt{d_x^2 + d_y^2 + d_z^2} \\ \mathrm{HDOP} &\triangleq \sqrt{d_x^2 + d_y^2} \\ \mathrm{VDOP} &\triangleq d_z. \end{aligned}$$

With these definitions the position dependent estimate for minimum variance on the estimator can then be determined by

$$\sigma_P(\boldsymbol{x}) = \sigma \cdot \mathrm{PDOP}(\boldsymbol{x})$$
$$\sigma_H(\boldsymbol{x}) = \sigma \cdot \mathrm{HDOP}(\boldsymbol{x})$$
$$\sigma_V(\boldsymbol{x}) = \sigma \cdot \mathrm{VDOP}(\boldsymbol{x}).$$

Exemplary, the DOP values were determined for a simple set up consisting of 8 fixed beacons at the positions specified in Tab. 2.1.

Node	1	2	3	4	5	6	7	8
p_x	0	20	20	0	0	20	20	0
p_y	0	0	10	10	0	0	10	10
p_z	0	0	0	0	6	6	6	6

Table 2.1.: Beacon positions

In this setup all positions for $\boldsymbol{x}$ lie on a plane with z-coordinate $z = 0$. Moreover, it is assumed that there is exactly one range measurement available from each beacon. Fig. 2.4 illustrates the horizontal, vertical and positional DOP for this layout.

The geometry of the beacons is shown in the first picture drawn as a box with the beacon positions in the corners. For this setup, the achievable minimum variance strongly variates by the influence of the geometrical setup. Accurate position estimation in this layout is only possible within close range to the beacons. Furthermore, the vertical dilution of precision dominates the final error. This is due to the fact that the expansion of the geometry is much weaker effused in z-direction than in x- and y-direction.

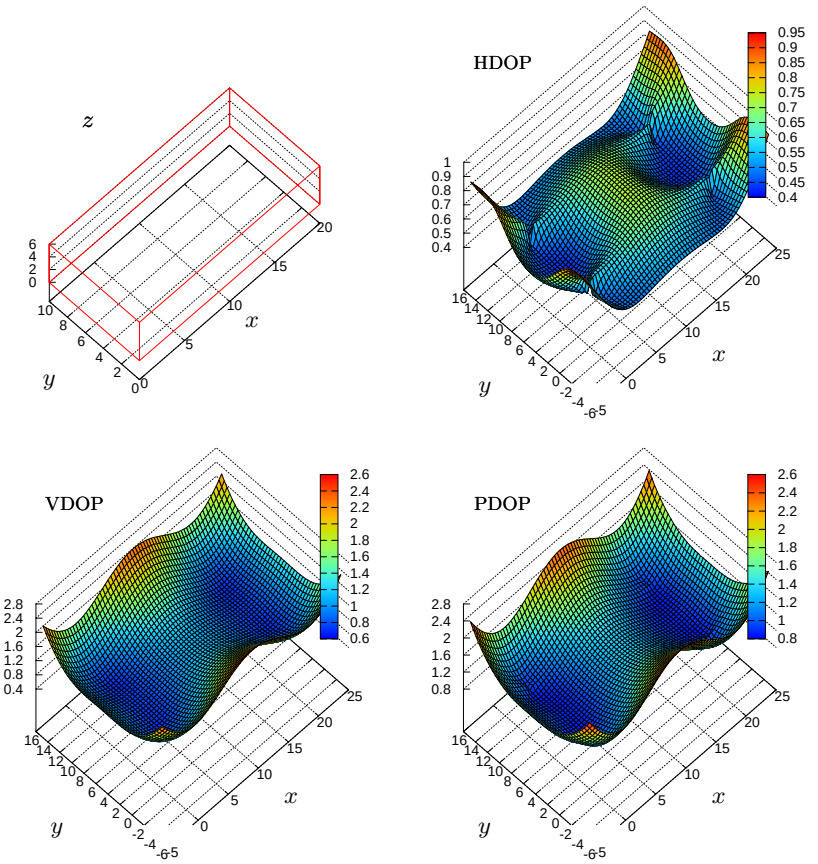

Figure 2.4.: Dilution of precision in x- and y-direction for $z = 0$

2.3. Relative localization

Relative localization is also commonly known as *dead reckoning*. With the formulations of Sec. 2.1 the basic idea of dead reckoning is to start with a known state $\boldsymbol{s}_1$ and iteratively evolve the state over time by solving Eq. (2.3).

2.3.1. Basic concepts

Historically, this kind of navigation was developed for marine navigation where the navigator used his knowledge about course and speed to develop an estimation for the actual position. Fig. 2.5 shows an old instrument for measuring the speed of the ship [27].

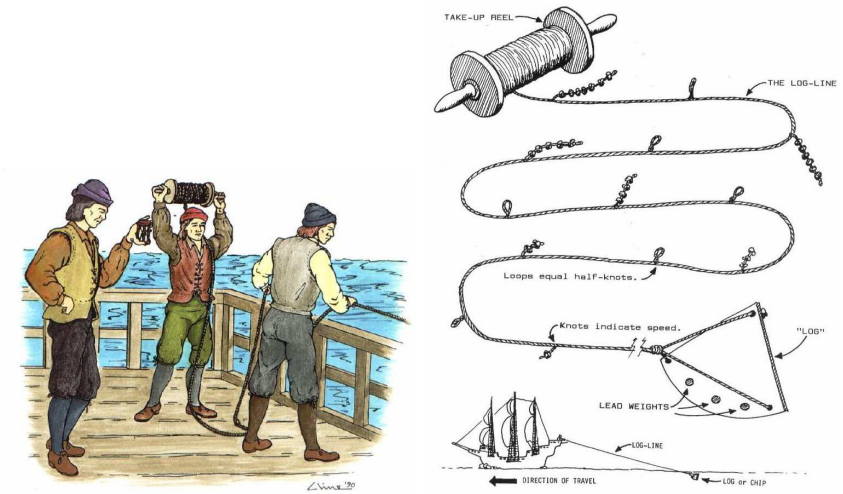

Figure 2.5.: Log line for measuring speed in knots

It consisted of a flat piece of wood, the so called log, which was weighted at the bottom to enable it to float upright in the water. A long rope was attached to the log, the so called log line that was wound on a spool so that it could be reeled out after the log was thrown into the water at the rear of the ship. The friction of the water held the log in place as the ship sailed away from it. On the log line, knots were tied at intervals of 7 fathoms which equates to about 12.8 meters. As the ship sailed away from the log, the sailors taking the reading would count the number of knots that passed over the rail in a period of half-a-minute that gave them their approximate speed in knots. From a mathematical

point of view, dead reckoning can be seen as a numeric integration of Eq. (2.3) over time by using control information. Since this technique tends to drift over time, updates of the real position are needed once and a while.

2.3.2. Inertial navigation

An exceptional position in this context is surely been taken by *Inertial Navigation Systems* (INS). These systems combine a sensor class called *Inertial Measurement Unit* (IMU) that measure accelerations and turn rates with an integration algorithm. Probably the greatest challenge for inertial navigation is based on the so called *equivalence principle*. It states the "complete physical equivalence of a gravitational field and a corresponding acceleration of the reference system" (Einstein 1907). The gravitational force of the earth $\boldsymbol{g}(\boldsymbol{x})$ which is a function of the position and accelerations with respect to a predefined coordinate system can therefore not be distinguished. Depending on the reference frame, gravitational and Coriolis effects on earth have to be taken care of. However, the importance of inertial navigation is justified by the unique proprioceptive character. An INS works without infrastructure, can't be jammed, and has a unique behavior concerning errors. Inertial navigation systems can be roughly subdivided into two classes.

Gimbaled systems

In gimbaled systems three linear accelerometers are placed on a gimbaled gyro stabilized platform. The gimbals are a set of three rings, each with a pair of bearings initially at right angles as depicted in Fig. 2.6.

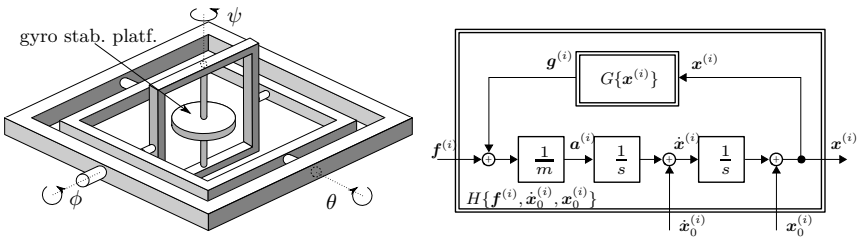

Figure 2.6.: Gimbal platform and integration scheme

This way the target may twist about any rotational axis while the platform keeps the same orientation with respect to an inertial frame (*i-frame*) which is stationary with respect to the fixed stars. Within this frame, the specific time dependent force $\boldsymbol{f}^{(i)} \triangleq \boldsymbol{f}^{(i)}(t)$ only has to be corrected by the gravitational force

vector $\boldsymbol{g}(\boldsymbol{x}^{(i)})$ that is dependent on the actual position of the accelerometers $\boldsymbol{x}^{(i)} \triangleq \boldsymbol{x}^{(i)}(t)$ before they directly can be integrated two times to obtain the actual location $\boldsymbol{x}^{(i)}$. The integration scheme is illustrated in Fig. 2.6 in form of a block diagram after performing a Laplace transformation.

The required starting position and the starting velocity is denoted by $\boldsymbol{x}_0^{(i)}$ respectively $\dot{\boldsymbol{x}}_0^{(i)}$. Applied on the framework of a dynamic system, the state vector of such an integration then takes the form

$$\boldsymbol{s}(t) = \begin{bmatrix} \boldsymbol{x}^{(i)}(t) \\ \dot{\boldsymbol{x}}^{(i)}(t) \end{bmatrix}. \tag{2.17}$$

For local geographic frames, like reference coordinate systems on the earth that are in movement due to earth rotation much more effort has to be taken [28]. One of the main disadvantages of the gimbaled scheme is that it usually uses many expensive precision mechanical parts that are moving, can wear out or jam and is vulnerable to the so called *gimbal lock*.

Strapdown systems

The second class of inertial system are strapdown systems. As the name already suggests, the system is strapped to the target as illustrated in Fig. 2.7.

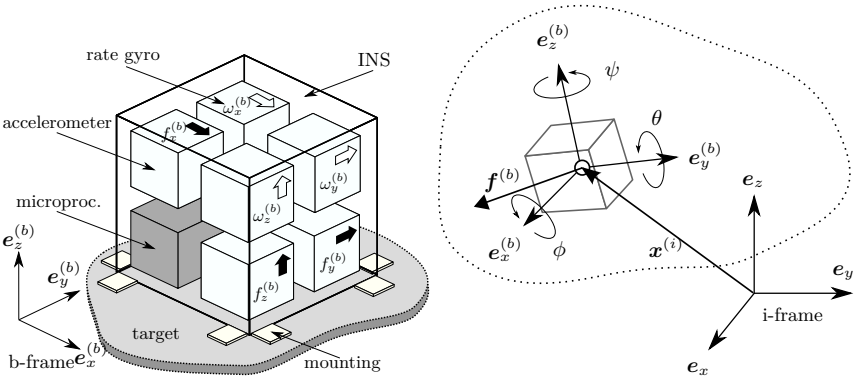

Figure 2.7.: Strapdown system and inertial frame

This reduces the cost, eliminates the gimbal lock, removes the need for some calibrations and increases the reliability by eliminating some of the moving parts. Instead of a gimbal, angular rate sensors called *rate gyros* measure the angular velocity of the targets changes in the body fixed frame (*b-frame*) that

moves along with the target as depicted in Fig. 2.7. Here $\boldsymbol{f}^{(b)}$ denotes the specific force measured by the accelerometers in a body fixed axis set and $\boldsymbol{x}^{(i)} = \left[x^{(i)}\ y^{(i)}\ z^{(i)}\right]^{\mathrm{T}}$ the actual position of the target in the i-frame as depicted in Fig. 2.7. Following the common denotation in the field of aviation (DIN 9300-2) [29] the orientation $\boldsymbol{\Theta}^{(i)} = [\phi\ \theta\ \psi]^{\mathrm{T}}$ is defined as the orientation of the object with respective to the i-frame [28]. The angles ϕ, θ and ψ are also known as *roll*, *pitch* and *yaw* and denote a rotation around the x-, y- and z-axis. The connection between the inertial and the body fixed frame can be described by the following orthogonal matrix

$$\boldsymbol{R}_i^b = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}}_{R_x} \underbrace{\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}}_{R_y} \underbrace{\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_z}.$$

With $\boldsymbol{R}_i^b$, the specific force vector measured by the accelerometers $f^{(b)}$ can be transformed to the inertial system

$$\boldsymbol{f}^{(i)} = \boldsymbol{R}_b^i \cdot \boldsymbol{f}^{(b)} = (\boldsymbol{R}_i^b)^T \cdot \boldsymbol{f}^{(b)}. \tag{2.18}$$

The actual orientation vector $\boldsymbol{\Theta}^{(i)}$, needed to generate the orientation matrix $\boldsymbol{R}_i^b$, is unknown and has to be evolved by the rate gyros.

Let

$$\boldsymbol{\omega}_{ib}^{(b)} = \left[\omega_x^{(b)}\ \omega_y^{(b)}\ \omega_z^{(b)}\right]^{\mathrm{T}}$$

denote the measured turn rate with respect to the i-frame in the b-frame. With an initial estimate for $\boldsymbol{\Theta}_0^{(i)}$ the orientation matrix $\boldsymbol{R}_b^i$ can be evolved by applying the following differential matrix equation

$$\dot{\boldsymbol{R}}_b^i = \boldsymbol{R}_b^i \cdot \boldsymbol{\Omega}_{ib}^b. \tag{2.19}$$

$\boldsymbol{\Omega}_{ib}^b$ denotes the *skew symmetric matrix*

$$\boldsymbol{\Omega}_{ib}^b = \begin{bmatrix} 0 & -\omega_z^{(b)} & \omega_y^{(b)} \\ \omega_z^{(b)} & 0 & -\omega_x^{(b)} \\ -\omega_y^{(b)} & \omega_x^{(b)} & 0 \end{bmatrix}.$$

Depending on the representation of the orientation as Euler Angles or quaternions, different evolving schemes for solving Eq. (2.19) can be derived. O. J. Woodman proposes to use the matrix exponential [30]

$$\boldsymbol{R}_b^i(t) = \boldsymbol{R}_b^i(0) \cdot \exp\left(\int_0^t \boldsymbol{\Omega}_{ib}^b(t)\mathrm{d}t\right). \tag{2.20}$$

Using the rectangle method for integration,

$$\boldsymbol{B} \triangleq \boldsymbol{\Omega}_{ib}^{b}\delta t \approx \int_{0}^{t} \boldsymbol{\Omega}_{ib}^{b}(t)\mathrm{d}t \tag{2.21}$$

and defining $\sigma \triangleq |\omega_{ib}^{b}|\delta t$ one can find the following update equation for the orientation matrix

$$\boldsymbol{R}_{b}^{i}(t+\delta t) = \boldsymbol{R}_{b}^{i}(t)(\boldsymbol{I} + \frac{\sin(\sigma)}{\sigma}\boldsymbol{B} + \frac{1-\cos(\sigma)}{\sigma^{2}}\boldsymbol{B}^{2}) \tag{2.22}$$

by substituting Eq. (2.21) into Eq. (2.20) and performing the Taylor expansion of the exponential term.

As depicted in Fig. 2.8, two consecutive integrations are needed for strapdown integration.

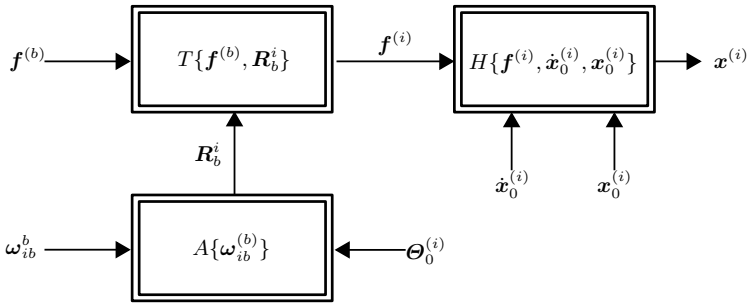

Figure 2.8.: Integration scheme for strapdown systems

Here $T\{\boldsymbol{f}^{(b)}, \boldsymbol{R}_b^i\}$ covers the transformation of Eq. (2.18) and $A\{\boldsymbol{\omega}_{ib}^{b}\}$ the integration of the orientation matrix as shown in Eq. (2.22). $H\{\boldsymbol{f}^{(i)}, \dot{\boldsymbol{x}}_0^{(i)}, \boldsymbol{x}_0^{(i)}\}$ denotes the same integration process developed for gimbaled systems as illustrated in Fig. 2.6. The state for strapdown integration consists of the position, velocity and orientation such that

$$\boldsymbol{s}(t) = \begin{bmatrix} \boldsymbol{x}^{(i)}(t) \\ \dot{\boldsymbol{x}}^{(i)}(t) \\ \boldsymbol{\Theta}^{(i)}(t) \end{bmatrix}. \tag{2.23}$$

As before, for local geographic frames like reference coordinate systems on earth a more sophisticated integration scheme is necessary.

Gyro technology

Strapdown integration requires three consecutive integrations as depicted in Fig. 2.8. As the integration starts with the angular rates a bias and general errors within the angular rate sensors transform into a positional error with $O(t^3)$. The limiting factor for strapdown systems thus is the quality of the gyros.

Basically there are two main principles for measuring angular rates. The first one is based on an optical feature called *Sagnac effect* [31, 32] that measures interferences of light beams. Two light beams are fed into opposing paths. Once the sensor is undergoing rotation then the beam travelling in the direction of rotation experiences a longer path to the exit than the other beam travelling against the rotation. When the beams exit they are combined again and interfere due to the phase shift caused by the rotation. The main types of this kind are *Ring Laser Gyro's* (RLG) and *Interferometric Fibre-Optic Gyro's* (IFOG). As this technology is quite elaborate the price tag for this kind of sensors is comparatively high.

The second principle is based on the Coriolis effect [33, 34] on vibrating masses. Both measuring principles are sketched in Fig. 2.9.

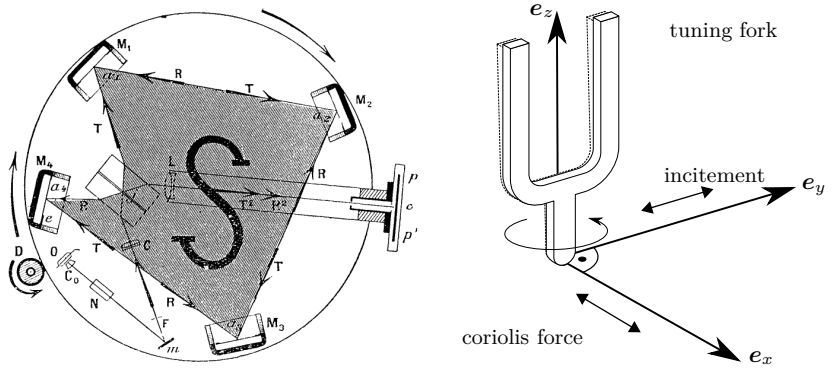

Figure 2.9.: Original sketch from Georges Sagnac and tuning fork

Within the picture a *tuning fork* vibrating gyro is artificially incited in y-direction. Every rotation around the z-axis causes a Coriolis force that can be measured in x-direction.

Conclusions

For strapdown systems the angular rates have to be integrated first to obtain the targets attitude before the integration process of the accelerometers can take

begin. Because of this additional error source, strapdown systems are considered to cause an inherently higher error compared to gimbaled systems. The largest error source for strapdown systems are bias instabilities that are measured in $[\frac{^\circ}{h}]$ and scale factor stability which is usually measured in parts per million (ppm) of the sensed inertial quantity. Fig. 2.10, taken from [35], gives a rough overview of gyro errors of assorted technologies and a rough estimate on the price in dollars for the corresponding accuracy.

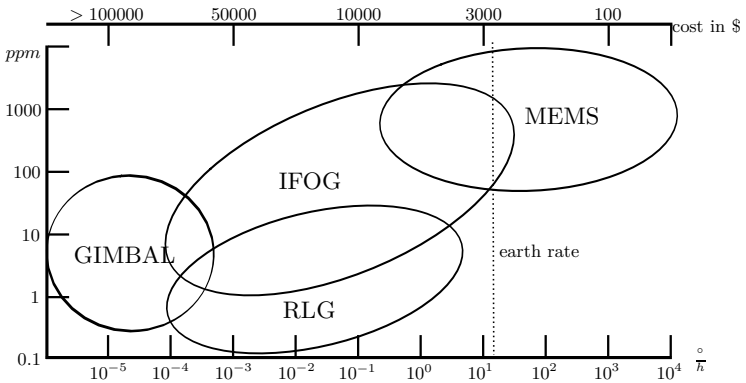

Figure 2.10.: Bias stability and scale factor stability of assorted technologies

Assumed a gyro within the price class of about 10000$ and a bias instability of $b = 0.1\frac{^\circ}{h}$ is applied. Then the error caused in the angle after the time t of integration is $\epsilon_\phi = b \cdot t$. Consequently, a wrong compensation of the gravitational force g causes an error in the accelerations of about $\epsilon_a \approx g \cdot \sin(\epsilon_\phi) \approx g \cdot b \cdot t$. This causes a positional error after two further integrations of about

$$\epsilon_x \approx g \cdot b \cdot \frac{t^3}{6}.$$

After 5 minutes of strapdown integration the estimated position error is about 20 meters. By inspecting Fig. 2.10, this accuracy can not yet be achieved by the relatively cheap *Micro-Electro-Mechanical Systems* (MEMS)-gyros that base on the principle of vibrating masses. For most indoor applications this scenario seems to be infeasible. However, their popularity bases on a lot of other profitable properties like small size and price tag [30]. Some authors expect these sensors to replace many of the current optical systems in near future [35].

2.4. Hybrid navigation and sensor data fusion

For relative navigation an initial solution $\boldsymbol{s}_0$ at time t_0 is required. In practice, a starting solution may be hard to obtain. Furthermore, relative navigation causes small errors for small time intervals and great errors for large time intervals. In contrast, most absolute navigation systems comprehend a source of constant error as already introduced for range based localization systems in Sec. 2.2.3. A combination of absolute navigation methods with relative navigation methods therefore is straightforward. This is especially true for inertial navigation systems.

2.4.1. Prerequisites and Notation

To use relative navigation techniques an appropriate state vector description is needed that represents the dynamic model from Eq. (2.3). For inertial navigation systems two different state vectors, Eq. (2.17) for gimbaled systems and Eq. (2.23) for strapdown systems, were introduced. To keep the notation as simple as possible, let $\boldsymbol{s}_i \triangleq \boldsymbol{s}(t_i)$ denote an appropriate state vector of the target at time t_i. The estimated position $\boldsymbol{x}_i \triangleq \boldsymbol{x}(t_i)$ can finally be reconstructed from the state vector.

Let

$$\boldsymbol{s}^{\langle n \rangle} \triangleq \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\}, \quad \boldsymbol{s}_i \triangleq \boldsymbol{s}(t_i)$$

be the set of all state vectors attained so far and analogously

$$\boldsymbol{u}^{\langle n \rangle} \triangleq \{\boldsymbol{u}_2, ..., \boldsymbol{u}_n\}, \quad \boldsymbol{u}_i \triangleq \boldsymbol{u}(t_i)$$

be the set of all control vectors. In the case of INS systems, $\boldsymbol{u}^{\langle n \rangle}$ consist of accelerations and turn rates. Furthermore let

$$\boldsymbol{z}^{\langle n \rangle} \triangleq \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}, \quad \boldsymbol{z}_i \triangleq \boldsymbol{z}(t_i)$$

denote the set of all observations. Note, in the case of range based localization $\boldsymbol{z}^{\langle n \rangle}$ consists of range measurements. For simplicity it is assumed that the state, the controls and the observations are synchronized. This implies that at each time step t_i with the state $\boldsymbol{s}_i$ there is exactly one control vector $\boldsymbol{u}_i$ and one observation $\boldsymbol{z}_i$ available. As addressed before, in any realistic scenario neither the control data nor the observation can be processed without errors. Sensor errors or inadequate modeling lead to a stochastic element in evaluating the dynamic model Eq. (2.3) and the observation model Eq. (2.4). Due to this, the sequence of the state variables evolving over time $\boldsymbol{s}^{\langle n \rangle}$ can also be interpreted as a *stochastic process*. This interpretation plays a crucial role in the following chapters.

2.4.2. Bayes filter derivation

In this section, all available information in form of controls $\boldsymbol{u}^{\langle n \rangle}$ and observations $\boldsymbol{z}^{\langle n \rangle}$ are fused to obtain a navigation solution. To this end, consider the *Maximum A Posteriori* (MAP) estimator for the state vector at time step t_n which formally reads

$$[\boldsymbol{s}_n]_{\mathrm{MAP}} = \arg\max_{\boldsymbol{s}_n} \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n \rangle}, \boldsymbol{u}^{\langle n \rangle}\right). \tag{2.24}$$

Applying Bayes rule the probability density function from Eq. (2.24) becomes [18, p. 694]

$$\mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \frac{\mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right)}{\mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right)}. \tag{2.25}$$

As the denominator of Eq. (2.25) is independent on $\boldsymbol{s}^{\langle n \rangle}$ it can be treated as a normalizing constant

$$\eta_n^{-1} \triangleq \mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right).$$

Usually it can be assumed that the actual observation is only dependent on the actual state and control such that

$$\mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{u}_n\right).$$

The sequence of states can be interpreted as a stochastic process. Assumed, this process fulfills the *Markov condition* which reads as

$$\mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{u}_n\right), \tag{2.26}$$

the *Chapman-Kolmogorov* equation [25] for

$$\mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \int \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) \mathrm{p}\left(\boldsymbol{s}_{n-1} | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n-1 \rangle}\right) \mathrm{d}\boldsymbol{s}_{n-1}$$

becomes

$$\mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \int \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{u}_n\right) \mathrm{p}\left(\boldsymbol{s}_{n-1} | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n-1 \rangle}\right) \mathrm{d}\boldsymbol{s}_{n-1}.$$

This simplifies the posterior probability density function from Eq. (2.24) to

$$\begin{aligned} &\mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{z}^{\langle n \rangle}, \boldsymbol{u}^{\langle n \rangle}\right) = \\ &\eta_n^{-1} \cdot \mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{u}_n\right) \int \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{u}_n\right) \mathrm{p}\left(\boldsymbol{s}_{n-1} | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n-1 \rangle}\right) \mathrm{d}\boldsymbol{s}_{n-1}. \end{aligned} \tag{2.27}$$

This is an update equation for the maximum a posteriori estimator. The components needed are the former probability density function $\mathrm{p}\left(\boldsymbol{s}_{n-1} | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n-1 \rangle}\right)$

for the previous state, a probability density function that describes the probability for the state change from $\boldsymbol{s}_{n-1}$ to $\boldsymbol{s}_n$ by known actual controls $\mathrm{p}(\boldsymbol{s}_n|\boldsymbol{s}_{n-1}, \boldsymbol{u}_n)$ and a probability density function for the probability of observing $\boldsymbol{z}_n$ with a known state $\boldsymbol{s}_n$ namely $\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{s}_n)$. By the assumption of the *Markov condition* from Eq. (2.26) and knowing the node positions $\boldsymbol{p}_1, \boldsymbol{p}_2$, the complete process can be visualized by a *Dynamic Bayes Network* as depicted in Fig. 2.11 [36]. This network represents the probabilistic relationships between all random variables and their conditional independencies via a directed acyclic graph. Unknown quantities are marked by dotted lines.

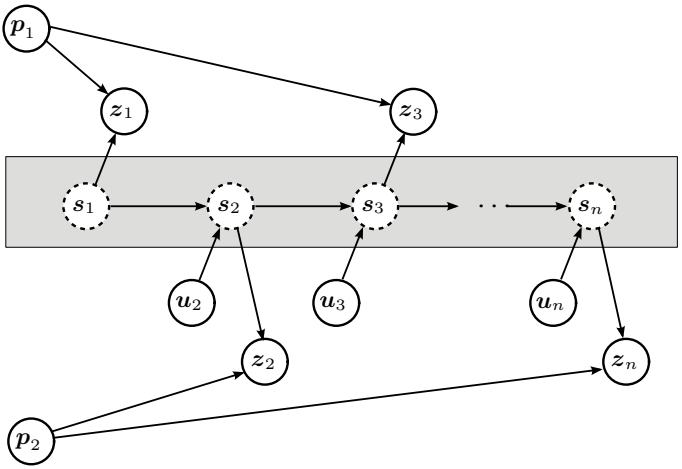

Figure 2.11.: Integrated navigation as a dynamic Bayesian network

Unfortunately, the update Eq. (2.27) cannot be evaluated in a closed form. The implementation of a solution would require the storage of the probability density functions which only may be representable by an infinity number of parameters.

2.4.3. Suboptimal filters

Sec. 2.1 formulates the navigation problem as a dynamic system. Most iterative solutions to the Bayesian update equation Eq. (2.27) assume that such a formulation is available. Enhanced by additive noise $\boldsymbol{\nu}_n$ that is modeling uncertainties in the dynamic model and $\boldsymbol{\eta}_n$, modeling the noise of the observation, the system

can be written as

$$\boldsymbol{s}_n = \boldsymbol{f}_{n-1}(\boldsymbol{s}_{n-1}, \boldsymbol{u}_n) + \boldsymbol{\nu}_n \quad (2.28)$$

$$\boldsymbol{z}_n = \boldsymbol{h}_n(\boldsymbol{s}_n, \boldsymbol{u}_n) + \boldsymbol{\eta}_n. \quad (2.29)$$

In analogy, the probability density $\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{s}_n, \boldsymbol{u}_n)$ can be obtained from the observation model described by Eq. (2.29). In many cases such a model is available in the specification of the respective sensor.

Once such a state space representation of the problem is known, the Bayesian update equation may be approximately evaluated by the *Extended Kalman Filter* (EKF). This approach is widely known and can be found in many books [37, 38, 28, 39, 20, 40]. The basic idea of the EKF is to model all of the required probability density functions from the Bayesian update equation by Gaussian probability density functions

$$\begin{aligned} \boldsymbol{s}_1 &\sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{s}_1}, \sigma^2_{\boldsymbol{s}_1}) \\ \boldsymbol{\nu}_i &\sim \mathcal{N}(0, \sigma^2_{\boldsymbol{\nu}_i}) \\ \boldsymbol{\eta}_i &\sim \mathcal{N}(0, \sigma^2_{\boldsymbol{\eta}_i}), \end{aligned}$$

such that the whole estimation process can be performed by matrix-matrix and matrix-vector multiplications. For linear models in Eq. (2.28) and Eq. (2.29) as well as normally distributed random variables $\boldsymbol{\nu}_n$ and $\boldsymbol{\eta}_n$, this approach can be shown to be optimal. However, as non-linear transformations of normally distributed random variables, as described by the state space representation at hand, do not have to remain normally distributed, the EKF additionally requires to linearize the system model and the observation model for inference of the covariance matrix. Nevertheless, this is the most applied approach for hybrid navigation and sensor data fusion because of its simplicity.

It is worth to note that there exist more refined methods that mainly differ in the treatment of the non linearities. A filter that is using the unscented transform for covariance update is referred to as the *Unscented Kalman Filter* (UKF) [41, 42]. As the EKF, the UKF models all random variables by Gaussian distributions. But unlike the EKF, the UKF does not linearize $\boldsymbol{f}_{n-1}(\boldsymbol{s}_{n-1}, \boldsymbol{u}_n)$ and $\boldsymbol{h}_n(\boldsymbol{s}_n, \boldsymbol{u}_n)$ for covariance update. Instead, it directly uses Eq. (2.28) and Eq. (2.29) to approximate the posterior $\mathrm{p}(\boldsymbol{s}_n|\boldsymbol{z}_n)$ by a Gaussian probability density using a set of deterministically chosen sample points. When propagated through a nonlinear transform, the sample points capture the true mean and covariance up to the second order of non-linearity with errors introduced in the third and higher orders [39].

A completely different procedure is performed by the *Particle Filter* (PF) [43, 44]. This method dates back to the 1950s and since then was continuously explored sporadically during the 1960s and 1970s [39]. The reason for the actual attraction of the PF within the navigation community is the advanced

computational power available today. The basic idea of the particle filter is *Sequential Monte Carlo* (SMC) estimation [43, 39] of probability densities. Within this technique a set of random samples with associated weights represent the probability densities. The associated weights are obtained by a technique called *importance sampling* [43, 39]. As every particle is propagated separately by the dynamic and the observation model, the more particles used for the filter, the higher the computational burden. However, this representation avoids the quadratic growth of elements within the covariance matrix for the EKF or UKF when the state vector becomes very big. Therefore, this filter can be advantageous for problems where the random variables can not be properly represented by Gaussian distributions or where the state vector becomes very big. For an increasing number of samples this method can be shown to converge to the true solution of Eq. (2.27) [39].

3. Motivation and problem definition

Based on the problem class of range based localization an augmented problem class is introduced in this chapter that motivates this thesis. A mobile target, unknown beacon positions and unsynchronized range measurements render this problem to be ill-conditioned.

Sec. 3.1 introduces the problem. It is shown, that the non linear least squares estimator from Sec. 2.2.2 cannot be successfully applied. This section reveals the basic problem with the new setting.

Sec. 3.2 presents related work and emphasizes the need for more research work.

3.1. Range only localization and mapping

The problem class presented here is referred to as *Range Only Localization And Mapping* (ROLAM) problem throughout this thesis.

3.1.1. Definition of the ROLAM problem

A set of k anchor nodes at the positions $\boldsymbol{p}_j \in \mathbb{R}^3$ with $1 \leq j \leq k$ form the infrastructure. Within the network range a vehicle moves along the trajectory

$$\boldsymbol{x} : \mathbb{R} \to \mathbb{R}^3, \quad t \mapsto \boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

and gathers range measurements $r_i \in \mathbb{R}$ at random time instants $t_1 \leq ... \leq t_n$ to randomly chosen static beacons. To identify the specific beacons the measurements correspond to the map

$$a : \{1, ..., n\} \to \{1, ..., k\}, \quad i \mapsto a_i,$$

from Sec. 2.2.2 is utilized such that

$$r_i \triangleq ||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| + \eta_i \tag{3.1}$$

with the abbreviation $\boldsymbol{x}_i \triangleq \boldsymbol{x}(t_i)$. In practice, measurements are erroneous. This is modeled by the additive measurement noise $\eta_i \in \mathbb{R}$. Furthermore, the random η_i variable is supposed to be mutually independent and the probability

density function $\mathrm{p}(\eta_i)$ is assumed to be known. With these notions the *ROLAM* problem can be defined as follows:

Definition 1. *(ROLAM) Let* $r^{\langle n \rangle} = \{r_1, ..., r_n\}$ *be a set of range observations measured at times* $t^{\langle n \rangle} = \{t_1, ..., t_n\}$ *and* $a^{\langle n \rangle} = \{a_1, ..., a_n\}$ *be the set of the respective data associations. Then the* ROLAM-Problem *is defined to determine the landmark positions* $\boldsymbol{p}^{\langle k \rangle} = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\}$ *while simultaneously tracking the path* $\boldsymbol{x}(t)$ *of the vehicle.*

Fig. 3.1 depicts the problem as a function block. The input is a set of range measurements with timestamp and data association and the output is an estimate for the trajectory and the node positions.

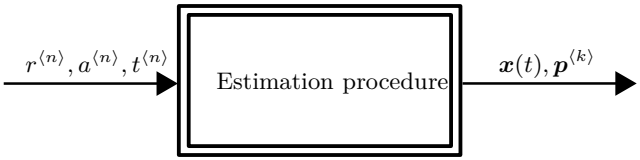

Figure 3.1.: Block representation of the ROLAM estimation problem

3.1.2. Definition of a Coordinate system

As the problem definition does not induce a coordinate system, there is an additional degree of freedom within this problem definition. Apparently, the whole scenario can be translated or rotated to any place and any orientation without changing the observations. Thus the degree of freedom is 6. Nevertheless, a solution with respect to a self defined local coordinate system can be obtained. One way to do so is sketched in Fig. 3.2.

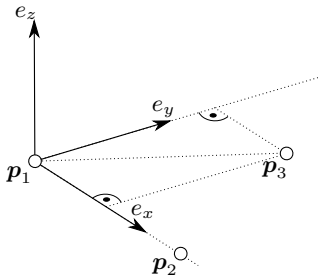

Figure 3.2.: Definition of a coordinate system for the ROLAM problem

If not all anchor nodes are collinear, the following right hand coordinate system can be defined. Without loss of generality it is assumed that the anchor nodes are enumerated in such a way that the first three nodes are not collinear. Then the origin of the new coordinate system can be chosen in such a way that the position of the first anchor node becomes the origin. The second anchor node lies on the x-axis of the new coordinate system and anchor node 3 on the first quadrant of the xy-plane. The z-axis is defined such that the coordinate system becomes an orthogonal right hand coordinate system. Once a global reference position and orientation is known the local solution can be translated from the local to the global coordinate system.

3.1.3. The ROLAM problem is ill-posed

The full Cartesian relationship between two positions $\boldsymbol{x}(t_i)$ and $\boldsymbol{x}(t_{i+1})$ is not observable when only ranges r_i and r_{i+1} are given. Moreover, without the knowledge on the beacon positions $\boldsymbol{p}^{\langle k \rangle}$ the problem can be considered to be ill-posed.

This is reflected by applying a ML estimator for the set of discrete position $\boldsymbol{x}^{\langle n \rangle} \triangleq \{\boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_n)\}$ of the mobile node and the positions of the static beacons $\boldsymbol{p}^{\langle k \rangle}$.

$$\left[\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{MAP}} = \arg \max_{\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}} \mathrm{p}\left(\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle} | r^{\langle n \rangle}, a^{\langle n \rangle}\right).$$

Applying Bayes Rule yields

$$\mathrm{p}\left(\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle} | r^{\langle n \rangle}, a^{\langle n \rangle}\right) = \mathrm{p}\left(r^{\langle n \rangle} | \boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}, a^{\langle n \rangle}\right) \frac{\mathrm{p}\left(\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle} | a^{\langle n \rangle}\right)}{\mathrm{p}\left(r^{\langle n \rangle} | a^{\langle n \rangle}\right)}. \quad (3.2)$$

The denominator of Eq. (3.2) $\mathrm{p}\left(r^{\langle n \rangle} | a^{\langle n \rangle}\right)$ can be considered as normalizing constant as it does not depend on $\boldsymbol{x}^{\langle n \rangle}$ and $\boldsymbol{p}^{\langle k \rangle}$. Furthermore, the formulation for the ROLAM problem (see Def. 1) does not give any information on the prior probability $\mathrm{p}\left(\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle} | a^{\langle n \rangle}\right)$. Removing these two terms and maximizing the remaining propability density function results in the ML Estimator

$$\left[\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{ML}} = \arg \max_{\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}} \mathrm{p}\left(r^{\langle n \rangle} | \boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}, a^{\langle n \rangle}\right).$$

The measurement error is assumed to be mutually independent. With Eq. (2.7) a change in random variables gives way to the simplification

$$\left[\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{ML}} = \arg \max_{\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}} \prod_{i=1}^{n} \mathrm{p}\left(\eta_i\right).$$

In analogy to Eq. (2.9), for normally distributed noise

$$p(\eta_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| - r_i)^2}{2\sigma_i^2}}, \quad 1 \leq i \leq n, \tag{3.3}$$

the negative log-likelihood becomes

$$\ell\big(\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\big) \triangleq \frac{1}{2} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} (||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| - r_i)^2 + \frac{1}{2} \sum_{i=1}^{n} \log(2\pi\sigma_i^2).$$

Therefore, the ML-estimator reads

$$\big[\boldsymbol{x}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\big]_{\mathrm{ML}} = \arg \min_{\boldsymbol{x}^n, \boldsymbol{p}^k} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} (||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| - r_i)^2. \tag{3.4}$$

This is almost identical to the non linear least squares approach from Sec. 2.2.2 that was optimal with respect to variance minimization. But minimization of Eq. (3.4) does not yield a unique solution for the ROLAM-problem. For any set of measurements $r^{\langle n \rangle}$, $a^{\langle n \rangle}$, there is an infinite number of beacon positions $\boldsymbol{p}^{\langle k \rangle}$ and vehicle positions $\boldsymbol{x}^{\langle n \rangle}$ such that

$$\sum_{i=1}^{n} \frac{1}{\sigma_i^2} (||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| - r_i)^2 = 0.$$

For every new measurement r_i the position $\boldsymbol{x}_i$ has to be estimated and thus adds 3 more unknowns compared to one additional known variable. Furthermore, the positions of the beacons $\boldsymbol{p}^{\langle k \rangle}$ are unknown and form another set of unknown parameters. Fig. 3.3 illustrates two different solutions that exactly comply to the same dataset for a two-dimensional ROLAM-problem. In order to find an accurate estimate for the unknown parameters $\boldsymbol{x}^{\langle n \rangle}$ and $\boldsymbol{p}^{\langle k \rangle}$ additional constraints are needed.

3.2. State of the art

This section classifies the ROLAM problem and discusses the applicability of existing methods. The ROLAM problem is situated between the problem of simultaneous localization and mapping and the problem of localization in self adjusting cell networks.

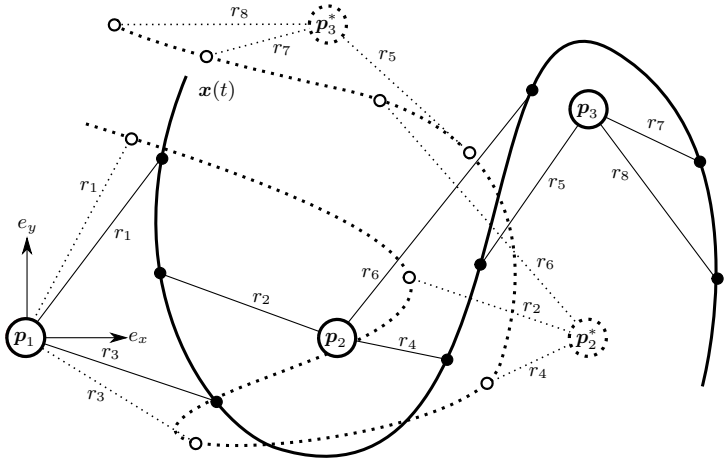

Figure 3.3.: Two optimal solutions to one data set

3.2.1. Simultaneous localization and mapping

When dealing with hybrid navigation techniques as described in Sec. 2.4 the update measurements $\boldsymbol{z}_i$ require knowledge on the infrastructure. For the case of range measurements, the beacon positions $\boldsymbol{p}^{\langle k \rangle}$ usually have to be known. A problem class that describes problems with unknown beacon positions is commonly known as *Simultaneous Localization And Mapping* (SLAM) [45]. Within this approach, unknown landmarks are spotted, identified and localized relative to the mobile unit. During this process a map of the environment is generated and the mobile unit is localized within the map at the same time. The main difference to classic hybrid localization systems is the unavailability of infrastructural information. In general the same assumptions are made as in Sec. 2.4.2. Consequently the problem can be described as a Dynamic Bayes Network as depicted in Fig. 3.4.

The dotted lines indicate unknown quantities. As before, the declared goal is to obtain the maximum a posteriori estimator

$$\left[\boldsymbol{s}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{MAP}} = \arg \max_{\boldsymbol{s}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle}} \mathrm{p}\left(\boldsymbol{s}^{\langle n \rangle}, \boldsymbol{p}^{\langle k \rangle} | \boldsymbol{z}^{\langle n \rangle}, \boldsymbol{u}^{\langle n \rangle}, a^{\langle n \rangle}\right)$$

for the unknown variables. Here, the set of positions $\boldsymbol{x}^{\langle n \rangle}$ was replaced with the more generic denotation $\boldsymbol{s}^{\langle n \rangle}$ to cover the dynamic state of the mobile unit. The general formulation for the SLAM problem even assumes that the sensor

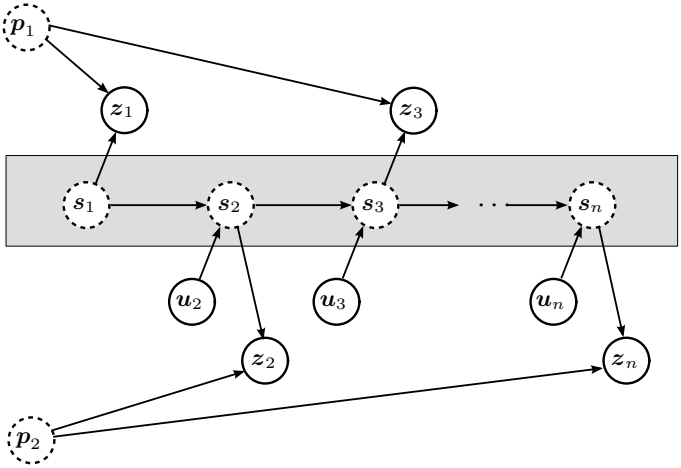

Figure 3.4.: SLAM as a dynamic Bayesian network

to data associations $a^{\langle n \rangle}$ are unknown. This is not the case for the ROLAM problem as formulated in Def. 1.

After performing the same transformations as in Sec. 2.4.2 one finally obtains the update equation [46]

$$\begin{aligned} \mathrm{p}\left(\boldsymbol{s}_n, \boldsymbol{p}^{\langle k \rangle} | \boldsymbol{z}^{\langle n \rangle}, \boldsymbol{u}^{\langle n \rangle}, a^{\langle n \rangle}\right) &= \mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{s}_n, \boldsymbol{p}^{\langle k \rangle}, a^{\langle n \rangle}\right) \cdot \\ &\cdot \int \mathrm{p}\left(\boldsymbol{s}_n | \boldsymbol{s}_{n-1}, \boldsymbol{u}_n\right) \mathrm{p}\left(\boldsymbol{s}^{\langle n-1 \rangle} | \boldsymbol{z}^{\langle n-1 \rangle}, \boldsymbol{u}^{\langle n-1 \rangle}, a^{\langle n-1 \rangle}\right) \mathrm{d}\boldsymbol{s}_{n-1}. \end{aligned} \tag{3.5}$$

With this augmented number of unknowns, the same approaches as for classic hybrid navigation can be used.

History and state of development

A seminal work in SLAM was the research of R.C. Smith and P. Cheeseman [45], that developed the theoretic foundations of this problem. In their work, the test scenario comprises a robot that observes features of unknown positions. To work with these unknowns the authors introduce a representation for spatial information of the beacon positions they call the *stochastic map*. As the robot is aware of his own control, it is able to develop a relative estimate of its own position. Furthermore, the probability density functions representing the targets position and the beacon positions are assumed to be normal. Under these conditions the paper shows that the estimate for the landmarks are necessarily

correlated with each other. To merge the approximate transformations between target and beacon position over time, R.C. Smith and P. Cheeseman [47] propose an Extended Kalman Filter *EKF* and fuse the observations with the relative positioning of the robot. This way, the beacons can be localized in form of the stochastic map. One of the first implementations using this approach was already realized in 1989 by Moutarlier and Chatila [48]. Further pioneering work in this field was conducted by the research group of Hugh F. Durrant-Whyte in the early 1990's [49, 50] who showed for the first time that the stochastic map converges monotonically to a relative map with zero uncertainty. The authors use an EKF with an enhanced state vector consisting of robot position and landmark position which was the dominant approach to the SLAM problem at that time. The authors further point out the increased complexity of the algorithm compared to classical hybrid navigation. As the state variable comprises of the targets state and the unknown feature positions, the number of elements of the covariance matrix grows quadratically with an increasing number of landmarks. Leonhard and Durrant-Whyte propose to reduce the filter to a series of decoupled beacon to vehicle filters as described in [49]. The latest and probably one of the most promising derivates of SLAM algorithms so far, is based on the work of Montemerlo and Thrun [51] omitting the former mentioned drawbacks. In the case of their *FastSLAM* algorithm, the authors propose a particle filter to estimate the trajectory and multiple EKF's to estimate the beacon positions [51]. This is achieved by a factored posterior representation

$$\mathrm{p}\left(\boldsymbol{s}^{\langle n\rangle}, \boldsymbol{p}^{\langle k\rangle} | \boldsymbol{z}^{\langle n\rangle}, \boldsymbol{u}^{\langle n\rangle}, a^{\langle n\rangle}\right) = \\ = \underbrace{\mathrm{p}\left(\boldsymbol{s}^{\langle n\rangle} | \boldsymbol{z}^{\langle n\rangle}, \boldsymbol{u}^{\langle n\rangle}, a^{\langle n\rangle}\right)}_{\text{path posterior}} \underbrace{\prod_{j=1}^{k} \mathrm{p}\left(\boldsymbol{p}_j | \boldsymbol{s}^{\langle n\rangle}, \boldsymbol{z}^{\langle n\rangle}, \boldsymbol{u}^{\langle n\rangle}, a^{\langle n\rangle}\right)}_{\text{landmark estimators}}$$

as presented in their paper.

Assumed, the observation $\boldsymbol{z}_i$ covers position information between $\boldsymbol{s}_i$ and $\boldsymbol{p}_{a_i}$ by using stereo cameras this approach is already approved to perform well with $log(n)$ complexity for the upper sketched SLAM problem. SLAM with fragmentary information though is a vibrant field of research at the moment. Täubig and Schröder adress the problem of fusing direction information offered by a monocular camera with control data of the target [52]. In their paper the authors propose an extended version of the FastSLAM algorithm to solve the problem in realtime. However, the authors also identify the initialization as the crucial point of their implemention. One of the most recent developments within this field was proposed in 2007 by Davison et al. [53]. In this paper, an algorithm called *MonoSLAM* is introduced that only uses information gathered

from a monocular camera. One of the key features is a constant velocity and constant angular velocity motion model for smooth camera movement replacing the control information. Of course such a model imposes a certain smoothness of the camera movement which can be guaranteed for most applications.

Adaptability to the ROLAM problem

As the beacons and their unknown positions $\boldsymbol{p}^{\langle k \rangle}$ can be considered to be an abstract map, the ROLAM problem can be interpreted as a SLAM problem. In some way, the ROLAM problem can be considered to be easier to solve than the classical SLAM problem as the data associations $a^{\langle n \rangle}$ are assumed to be known within the problem definition. The ROLAM problem, as it was introduced in Sec. 3.1, does not include the *correspondence problem*. Nevertheless, the sparse amount of information makes the problem hard to handle. Recall the characteristics of the ROLAM problem:

- No control data $\boldsymbol{u}^{\langle n \rangle}$ is available.
- The measurements $r^{\langle n \rangle}$ contain only range information instead of full relative position.
- The measurements are not aligned in time: $t_i \neq t_{i+1}$.

A straightforward implementation of an existing method is therefore not possible.

3.2.2. Self adjusting cell networks

Another field of work related to the ROLAM problem is the field of self calibrating cell networks. Usually, such networks use information based on ranges (RSS,TOA) or range differences (TDOA) to calculated the geometry of the network. The authors of [54, 55, 56] and [57] for example propose solutions to the problem of geometry estimation of a cell network based on range data. These methods solve the mapping problem for a static scenario with pairwise range distance information available in between the beacons. As the ROLAM problem only offers range measurements to the target, these methods are not applicable.

Grabowski and Khosla investigate a dynamic scenario with a set of small robots that pairwise measure ranges to other robots [58]. At each timestep all measures are assumed to be synchronized. To improve the accuracy concerning the geometrical setup and outlier detection the authors use a simple statistical motion model based on control information for rotation and translation. Once a measurement seems to be implausible with regard to the mobility constraints of the mobile unit it is discarded. Although this method is not applicable to the ROLAM problem attention can be drawn to the motion model.

Similar to the ROLAM problem, some authors propose moving targets for network calibration. In [59], Galstyan et al. present two solutions using a mobile unit to calibrate a sensor network. The first scenario assumes a moving beacon with known position that measures ranges to the static beacons. In the second scenario the position of the moving target is unknown but the author emphasizes that, as long as there are enough nodes of known positions in the vicinity of a given node, the target can be localized and this information can be used to impose new constraints on the position of the target. Within the ROLAM problem these constraints are not met. In [60], Cevher and McClellan propose an EKF that estimates the position of a moving target and at the same time the geometry of the static cell network. In contrast to the ROLAM problem, the input consists of direction information in the form of AOA. Moreover, to obtain a feasible solution, a target with constant velocity is assumed which is not applicable to ROLAM.

The closest match to the ROLAM problem poses the work from Newman and Leonard [61]. Here an autonomous underwater vehicle is equipped with acoustic transceivers that measure ranges by TOA between vehicle and small subsea transponders of unknown position. These measurements are not assumed to be synchronized. The authors further presume a certain depth of the vehicle so that the problem can be considered to be two-dimensional. To obtain a unique solution, the authors propose a simple trajectory model of constant velocity. After discretization this leads to a trivial linear relationship between three consecutive poses of the form $\boldsymbol{x}_{i-1} - 2\boldsymbol{x}_i + \boldsymbol{x}_{i+1} = 0$. As a result, an equation system is presented that consists of the trajectory model as well as the measurement model that has to be solved in whole. Following Olson, Leonard and Teller [62], this approach suffers from "significant convergence problems". They propose an alternative method that works in two steps. At first an initial estimate on the beacon positions is obtained by finding pairwise intersections of a set of measurements. Within the approach, the association of the appropriate measurements is established by the vehicle's dead-reckoned position. Each of these intersections is gathered in an accumulator that is represented by a grid chosen in such a way that it matches the total uncertainty in the solution. The second step incorporates an EKF that uses the previously found solution as initialization. The incorporated motion model assumes no movement in between timesteps. As for the ROLAM problem, no initial estimate can be found without any additional information of the vehicles movement.

An exact division of all available methods into SLAM and self adjusting networks can not be performed. The following method, called LaSLAT can be positioned in between both methods and is close to what is needed to solve the ROLAM problem. Therefore the next section is dedicated to this method.

3.2.3. LaSLAT

In [63] TAYLOR ET AL. propose an algorithm that solves a special variant of the ROLAM problem the authors call the *Simultaneous Localization And Tracking* (SLAT) problem. In contrast to the ROLAM problem the range measurements are assumed to be synchronized in time. The presented algorithm is a Bayesian filter that provides estimates for both, the target location and the beacon positions as well as their uncertainties in form of covariance matrices. As the method extensively uses Laplaces method to approximate occurring covariance matrices the algorithm is named LaSLAT. This algorithm is sketched in more detail.

The approach

Let $\boldsymbol{q}_i=[\boldsymbol{p}_i \quad \theta_i]^{\mathrm{T}}$ represent unknown parameters from a beacon with index i. $\boldsymbol{p}_i \in \mathbb{R}^3$ denotes the beacon position and $\theta_i \in \mathbb{R}$ a calibration parameter denoting a constant bias for the range measurements. The vector $\boldsymbol{q} \triangleq [\boldsymbol{q}_1^{\mathrm{T}} \quad \cdots \quad \boldsymbol{q}_k^{\mathrm{T}}]^{\mathrm{T}}$ denotes the whole set of parameters for all sensors. Let furthermore $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$ denote the position of the moving target at time t_i. The range measurements are assumed to be corrupted by additive zero mean Gaussian noise such that

$$r_i = ||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| + \theta_{a_i} + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2), \tag{3.6}$$

with η_i denoting the noise. Moreover, r_i and a_i denote the range measurement at the time t_i and the measurement to data association, respectively. To increase the convergence speed, the authors propose to divide the sequence of the target positions $\boldsymbol{x}_i$ into batches, each of them containing a subsequence of m positions. Let

$$I^j : \mathbb{N} \to \mathbb{N}, \quad i \mapsto i + (j-1)m, \quad I_i^j \triangleq I^j(i),$$

denote a map used for indexing with $j \geq 1$. Then

$$\boldsymbol{y}_n \triangleq \left[\boldsymbol{x}_{I_1^n}^{\mathrm{T}} \quad \cdots \quad \boldsymbol{x}_{I_m^n}^{\mathrm{T}}\right]^{\mathrm{T}}$$

denotes the vector of the n th batch and

$$\boldsymbol{z}_n \triangleq \left[r_{I_1^n}^{\mathrm{T}} \quad \cdots \quad r_{I_m^n}^{\mathrm{T}}\right]^{\mathrm{T}}$$

the vector of the respective measurements. Moreover let

$$\boldsymbol{z}^{\langle n \rangle} \triangleq \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$$

denote the set of all measurement batches until batch n. Then, the goal of LaSLAT is to iteratively compute the posterior density for new batches, given by

$$\mathrm{p}\left(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n \rangle}\right) \propto \mathrm{p}\left(\boldsymbol{z}_n | \boldsymbol{y}_n, \boldsymbol{q}\right) \mathrm{p}\left(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n-1 \rangle}\right). \tag{3.7}$$

Eq. (3.7) can be obtained by applying *Bayes Rule*. This equation resembles to the SLAM formulation introduced in Sec. 3.2.1 and could be treated in the same way. Nevertheless, the authors emphasize that an iterative approach in form of an EKF for solving this equation may be inaccurate as the measurement model from Eq. (3.6) is not linear. Instead of using the update step of the EKF, LaSLAT approximates the true posterior $\mathrm{p}\left(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n \rangle}\right)$ with a Gaussian distribution $\mathrm{q}(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n \rangle})$ obtained from an optimization step that incorporates a *Newton-Raphson* optimization algorithm. The mean of $\mathrm{q}(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n \rangle})$ is the minimum of the optimization problem at step n that writes as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_n^{\mathrm{T}} & \boldsymbol{q}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \arg\min_{\boldsymbol{y}_n, \boldsymbol{q}} (\boldsymbol{q} - \bar{\boldsymbol{q}})^{\mathrm{T}} \bar{\boldsymbol{\Omega}} (\boldsymbol{q} - \bar{\boldsymbol{q}}) + \\ + \frac{1}{\sigma^2} \sum_{i=I_1^n}^{I_m^n} (||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| + \theta_{a_i} - r_i)^2, \end{aligned} \tag{3.8}$$

with the estimated prior distribution parameterized by

$$\bar{\boldsymbol{q}} = \mathrm{E}\left[\boldsymbol{q} | \boldsymbol{z}^{\langle n-1 \rangle}\right], \quad \text{and} \quad \bar{\boldsymbol{\Omega}} = \mathrm{Cov}^{-1}\left[\boldsymbol{q} | \boldsymbol{z}^{\langle n-1 \rangle}\right].$$

The covariance matrix of the distribution is obtained by *Laplace's method* as described in Sec. 5.1.2. The first term of Eq. (3.8) represents the negative logarithm of $\mathrm{p}\left(\boldsymbol{y}_n, \boldsymbol{q} | \boldsymbol{z}^{\langle n-1 \rangle}\right)$ and the second term of $\mathrm{p}(\boldsymbol{z}_n | \boldsymbol{y}_n, \boldsymbol{q})$. The predictions $\bar{\boldsymbol{q}}$ and $\bar{\boldsymbol{\Omega}}$ are obtained by marginalizing out $\boldsymbol{y}_{n-1}$ from the former result $\mathrm{q}(\boldsymbol{y}_{n-1}, \boldsymbol{q}_{n-1} | \boldsymbol{z}^{\langle n-1 \rangle})$ for the previous batch. With

$$\mathrm{Cov}^{-1}\left[\begin{bmatrix} \boldsymbol{y}_{n-1}^{\mathrm{T}} & \boldsymbol{q}_{n-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \middle| \boldsymbol{z}^{\langle n-1 \rangle} \right] = \begin{bmatrix} \boldsymbol{\Omega}_{\boldsymbol{y}_{n-1}} & \boldsymbol{\Omega}_{\boldsymbol{y}_{n-1}\boldsymbol{q}_{n-1}} \\ \boldsymbol{\Omega}_{\boldsymbol{q}_{n-1}\boldsymbol{y}_{n-1}} & \boldsymbol{\Omega}_{\boldsymbol{q}_{n-1}} \end{bmatrix}$$

the predictions become

$$\bar{\boldsymbol{q}} = \boldsymbol{q}_{n-1} \quad \text{and} \quad \bar{\boldsymbol{\Omega}} = \boldsymbol{\Omega}_{\boldsymbol{q}_{n-1}} - \boldsymbol{\Omega}_{\boldsymbol{y}_{n-1}\boldsymbol{q}_{n-1}} \boldsymbol{\Omega}_{\boldsymbol{y}_{n-1}}^{-1} \boldsymbol{\Omega}_{\boldsymbol{q}_{n-1}\boldsymbol{y}_{n-1}}.$$

As can be seen by Eq. (3.8), the authors do not propose a dynamic model or prior knowledge on the targets position within LaSLAT. However, they propose a way to extend the existing framework to incorporate prior knowledge on the movement. Without any prior knowledge, the initial probability density $\mathrm{q}(\boldsymbol{y}_0, \boldsymbol{q})$ is initialized by using a diagonal covariance matrix with great diagonal elements. No further comment on initializing the mean is made.

The presented results for LaSLAT were obtained by using the *Cricket* ranging system [64]. Without batching, the standard EKF applied on Eq. (3.7) is the fastest of all tested algorithms. However, the results show the advantage of the optimization step compared to a standard EKF by an increased rate of convergence, even for the LaSLAT implementation with a batch size of one. Moreover,

the results show that an increased batch size greatly improves the speed of convergence. The size of the batch also influences the clock cycle obtaining actual results. For every new step, the algorithm has to wait for a new batch of measurements before starting the iteration step. Concerning the quality, the best result for this approach is achieved by treating all available measurements as one single batch. Without any knowledge on $\mathrm{q}(\boldsymbol{y}_0, \boldsymbol{q})$, the minimization problem from Eq. (3.8) becomes a least squares problem. If one applies LaSLAT on the ROLAM problem by ignoring the bias parameters θ_i and introducing a variable variance σ_i for the measurements, one ends up with the ML estimator already introduced in Sec. 3.1.3. This shows that LaSLAT can not be applied on the ROLAM problem without preprocessing the data. LaSLAT requires a greater number of measurements than the number of unknowns in every optimization step.

Alignment in time domain

For the ML estimator from Eq. (3.4) and LaSLAT feasibility can be assured by an alignment of the range data in time domain. Note, this can be interpreted as a simple path model that only allows a change in position at a few discrete time instances.

Let

$$T^{\langle l \rangle} \triangleq \{T_1, ..., T_l\}, \quad T_{i+1} - T_i = \Delta T, \quad l \leq n \tag{3.9}$$

denote a discrete time domain and

$$\boldsymbol{X}^{\langle l \rangle} \triangleq \{\boldsymbol{X}_1, ..., \boldsymbol{X}_l\}$$

a corresponding alignment for all attained positions. The alignments of $t^{\langle n \rangle}$ and $\boldsymbol{x}^{\langle n \rangle}$ to a new set of time instances and acquired positions

$$\check{t}^{\langle n \rangle} = \{\check{t}_1, ..., \check{t}_n\}, \quad \check{\boldsymbol{x}}^{\langle n \rangle} = \{\check{\boldsymbol{x}}_1, ..., \check{\boldsymbol{x}}_n\}$$

can then be performed by

$$\check{t}_i = \arg \min_{\tau \in T^{\langle l \rangle}} |t_i - \tau| \quad 1 \leq i \leq n.$$

Performing this, a map

$$s : \{1, ..., n\} \to \{1, ..., l\} \quad i \mapsto j$$

can be constructed such that

$$\check{t}_i = T_{s(i)} \quad \check{\boldsymbol{x}}_i = \boldsymbol{X}_{s(i)}.$$

With this modification of the data, the ML-estimator from Sec. 3.1.3 results in

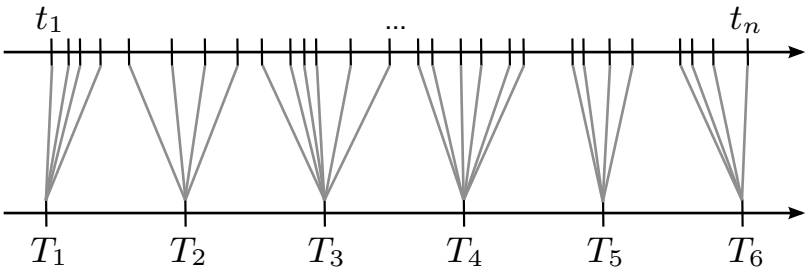

Figure 3.5.: Alignment map for $l \triangleq 6$ from $\boldsymbol{t}^n$ to $\boldsymbol{T}^l$

$$\left[\boldsymbol{x}^{\langle n\rangle}, \boldsymbol{p}^{\langle k\rangle}\right]_{\mathrm{ML}} = \underset{\check{\boldsymbol{x}}^n, \boldsymbol{p}^k}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (||\check{\boldsymbol{x}}_i - \boldsymbol{p}_{a_i}|| - r_i)^2, \quad \check{\boldsymbol{x}}_i \in \boldsymbol{X}^{\langle l\rangle}. \tag{3.10}$$

Apparently, this procedure greatly reduces the number of unknown variables as $\check{\boldsymbol{x}}_i = \check{\boldsymbol{x}}_j$ for $s(i) = s(j)$ but still remains a problem of the same class. However, a unique solution can only be assured if the number of unknown parameters is smaller than the number of measurements.

$$3l + 3k - 6 \leq n. \tag{3.11}$$

Furthermore, at least four available range measurements must be associated to each timestep T_i, $1 \leq i \leq l$ to uniquely define the appropriate position $\boldsymbol{X}_i$.

A small l therefore increases the chance to obtain a unique solution. However, the result becomes more grainy because of the alignment error. If l is chosen too large, measurement errors strongly influence the solution and render a noisy result until the point where Eq. (3.11) gets infringed.

3.2.4. Conclusions

The ROLAM problem as defined in this thesis is an ill-posed problem and therefore can not be resolved without additional assumptions. It can be classified as a special occurrence of a SLAM problem. By the cell based setup for ROLAM there is also a strong overlap of ROLAM with self adjusting cell-networks. However, existing algorithms for SLAM or self adjusting cell networks cannot be applied directly due to the sparse amount of information within the ROLAM problem. Only very few methods deal with ill-posed localization problems. Within these methods very basic dynamic models that are customized for the particular application are implemented.

Motivated by the ill-posed ROLAM problem this thesis contributes with a novel and generic dynamic model for ill-posed localization problems. This model

allows to compensate missing information for localization problems and to quantify a minimal amount of data required for the specific application.

4. Kernel methods for localization

This chapter establishes a generic dynamic model to solve ill-posed localization problems. As it is impossible to get something for nothing, the popular adage "*There Ain't No Such Thing As A Free Lunch*" from a science fiction novel [65] also holds for ill-posed localization problems. Without inserting additional information into an infeasible problem formulation and thus changing the problem at hand, it stays infeasible.

The basic idea in this thesis is to represent the unknown trajectories as a member of a vector space of functions and then to restrict the search space to compact sets. This technique is strongly related to *Tikhonov regularization* introduced by Tikhonov and Arsenin in 1977 [66]. The particular regularization technique of *Kernel Regression* is proposed to deal with the ROLAM problem. Within this technique the solution is embedded within a complete function space called *Reproducing Kernel Hilbert Space* (RKHS). Kernel Regression and the RKHS is introduced in Sec. 4.1.

For good performance the artificially inserted information needs to reflect the true situation. Within kernel regression the restriction is performed by a penalty term in form of a vector space norm of the RKHS that balances data compliance and smoothness. The critical design element of the penalty term is given by the form of the kernel function that spans the RKHS. The design of the kernel function for localization problems is addressed in Sec. 4.2. First, a physical motivated model allows to design a *Stochastic Process* that reflects the problem at hand. Then, the realizations of the stochastic process are embedded within a RKHS. From this perspective it is shown that solving the resulting minimization problem can also be interpreted as maximum a posteriori estimation.

The minimization problem for the ROLAM problem is explicitly formulated in Sec. 4.3.

Sec. 4.4 clarifies under which conditions kernel regression can be successful adapted to ill-posed localization problems. The more restrictive the solution space the more solutions are filtered out and the more stable the approach. On the other hand reasonable solutions should not be excluded from the set of possible solutions. Thus, the kernel design plays a major role within this chapter.

4.1. Kernel regression

In statistics, regression analysis is a technique to investigate relationships between variables. During the regression procedure the investigator gathers observations on some underlying parameters of interest and employs a fit to estimate the effect of the data on the parameters. Most commonly, the estimate consists of a conditional expectation of the unknown parameters given some erroneous data.

Let

$$f : \mathrm{T} \to \mathrm{F}, \quad t \mapsto f(t, \boldsymbol{\theta})$$

denote the *regression function* that is defined by a finite parameter vector $\boldsymbol{\theta} \in \Theta$. Let's further assume the availability of n observations $z_i \in \mathrm{Z}$, $1 \leq i \leq n$ that are connected to $f(t, \boldsymbol{\theta})$ via an observation model. Then, the optimal parameter vector is obtained by minimizing a rating functional of the form

$$\boldsymbol{\theta}_{\mathrm{opt}} = \arg\min_{\boldsymbol{\theta} \in \Theta} R(f(t, \boldsymbol{\theta}), z_1, ..., z_n).$$

Therefore, two design aspects can be considered as the key features of accurate regression methods:

1. An appropriate parameter set Θ that defines the solution space.
2. An adequate rating function $R(f(t, \boldsymbol{\theta}), z_1, ..., z_n)$ that describes the underlying relationship of the variables.

4.1.1. Empirical risk minimization and consistency

For kernel regression, the rating functional can be motivated by the concept of *Expected Risk Minimization* [67, p. 66].

Let Θ be an appropriate parameter set such that $f(t, \boldsymbol{\theta})$ is well defined for any $t \in \mathrm{T}$ and $\boldsymbol{\theta} \in \Theta$. The solution space is denoted by

$$\mathcal{F} = \{ f(., \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta \}.$$

In analogy to Chapter 2 let

$$h : \mathrm{F} \to \mathrm{Z}, \quad f(t, \boldsymbol{\theta}) \mapsto z$$

denote a known observation model that describes the connection of observations and parameters.

Definition 2. (Loss Function)
Denote by $(t, z, \bar{z}) \in \mathrm{T} \times \mathrm{Z} \times \mathrm{Z}$ the triplet consisting of a pattern t, an observation z and a prediction $\bar{z} = h(f(t, \boldsymbol{\theta}))$. Then the map $c : \mathrm{T} \times \mathrm{Z} \times \mathrm{Z} \to [0, \infty)$ with the property $c(t, z, z) = 0$ for all $t \in \mathrm{T}$ and $z \in \mathrm{Z}$ is called a loss function.

By definition, the loss function describes the quality of the fit of an element of $f \in \mathcal{F}$ with respect to one available observation. While a perfect fit is designed to cause no loss, the loss is increasing for a bad fit.

The concept of expected risk minimization is to minimize the functional

$$R_{\exp}[f] \triangleq \mathrm{E}[c(t,z,\bar{z})] = \int_{\mathrm{Z}} \int_{\mathrm{T}} c(t,z,\bar{z}) \mathrm{p}(t,z)\, \mathrm{d}t\mathrm{d}z, \quad \bar{z} \triangleq h(f(t,\boldsymbol{\theta})), \tag{4.1}$$

with respect to $\boldsymbol{\theta} \in \Theta$ which is called the *expected risk*. Here, $\mathrm{p}(t,z)$ can be thought of the probability of the observation z at time t. In general, this density function is unknown. Instead, a finite number of observations $z_1, ..., z_n$ can give a hint of the real density function. $\mathrm{p}(t,z)$ is therefore often replaced by the *empirical density*

$$\mathrm{p}_{\mathrm{emp}}(t,z) \triangleq \frac{1}{n} \sum_{i=1}^{n} \delta_{t_i}(t) \delta_{z_i}(z)$$

with $\delta_a(t) \triangleq \delta(a-t)$ and δ as the *Dirac delta distribution* [18, p. 591]. Fig. 4.1 sketches the true probability density of the observations and some discrete samples.

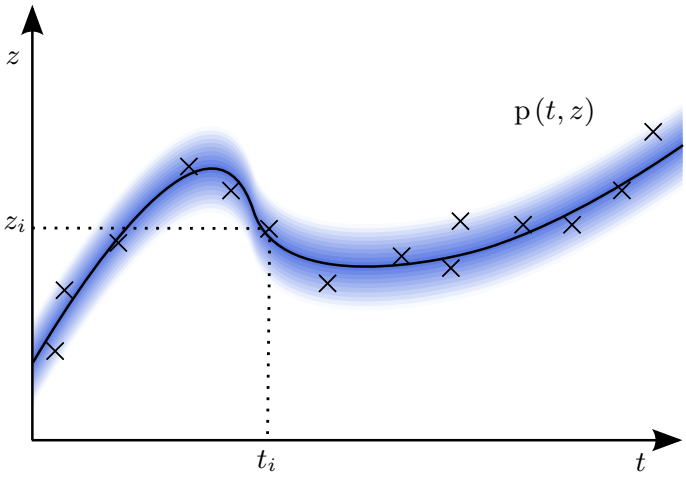

Figure 4.1.: Probability density for the observation and samples

Inserted into Eq. (4.1) this replacement leads to the rating functional

$$R_{\mathrm{emp}}[f] \triangleq \int_{\mathrm{Z}} \int_{\mathrm{T}} c(t, z, \bar{z}) \mathrm{p}_{\mathrm{emp}}(t, z) \, \mathrm{d}t \mathrm{d}z = \frac{1}{n} \sum_{i=1}^{n} c(t_i, z_i, \bar{z}_i) \tag{4.2}$$

also called the *empirical risk*. Consequently, minimization of Eq. (4.2) is called *empirical risk minimization*. In many cases, the loss function is designed to result in a maximum likelihood estimator.

For example let z_i denote the observation at time t_i that is subject to additive and i.i.d. Gaussian noise

$$z_i = f(t_i, \boldsymbol{\theta}) + \eta_i, \quad \mathrm{p}\left(z_i | \boldsymbol{\theta}, t_i\right) = \mathcal{N}(f(t_i, \boldsymbol{\theta}), \sigma_i^2).$$

Then, the negative log-likelihood becomes

$$\ell(\boldsymbol{\theta}) = -\ln \prod_{i=1}^{n} \mathrm{p}\left(z_i | \boldsymbol{\theta}, t_i\right) = \sum_{i=1}^{n} \left(\frac{(z_i - f(t_i, \boldsymbol{\theta}))^2}{2\sigma_i^2} - \ln \frac{1}{\sqrt{2\pi}\sigma_i} \right).$$

Let now

$$c(t_i, z_i, \bar{z}_i) \triangleq \frac{(z_i - \bar{z}_i)^2}{\sigma_i^2}$$

define a cost function. Then minimizing the sum of all cost functions

$$[\boldsymbol{\theta}]_{\mathrm{ML}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} c(t_i, z_i, \bar{z}_i) = \arg\min_{\boldsymbol{\theta} \in \Theta} R_{\mathrm{emp}}[f]$$

results in the ML estimator.

From this perspective, the ML estimator from Eq. (3.4) can be interpreted as a minimizer for the empirical risk. With this notion it is possible to use results from the literature that clarify when minimization of the empirical risk Eq. (4.2) can be successful. In 1998 Vapnik and Chervonenkis, one of the main contributors to the theory of *Support Vector Machines* (SVN), stated that the basic concept is to assure consistency.

Theorem 1 (Vapnik & Chervonenkis [68])**.** *One-sided uniform convergence in probability,*

$$\lim_{n \to \infty} \mathrm{P}\{\sup_{f \in \mathcal{F}} (|R[f] - R_{\mathrm{emp}}[f]|) > \epsilon\} = 0$$

for all $\epsilon > 0$, is a necessary and sufficient condition for nontrivial consistency of empirical risk minimization.

For consistency, the difference between the empirical risk and the expected risk has to converge to zero in probability as the number of samples is increased. As mentioned before, there is a connection between the capacity of the solution space $\mathcal{F}$ and consistency. If the function space is very capacious then there is a higher risk of choosing an infeasible solution. In fact there is more than one loose bound available in the literature stating the latter [68, 69, 70]. All these methods differ in the way they measure capacity. The problem formulation from Eq. (3.4) did not assure consistency because the solution space was too capacious.

4.1.2. Reproducing Kernel Hilbert Spaces

A popular way to reduce the solution space is the assumption of linearity which is covered by linear regression techniques. Although such a model could be directly applied to the ROLAM problem it is clear that a linear movement only is not representative enough for most cases. Without the concrete knowledge of an appropriate solution space modelled by the space of parameters Θ one has to deal with contradictory goals. A small number of parameters makes the problem behave well conditioned but also restricts the diversity of the solution space as for the linear case. With an increasing number of a priori unknown parameters $\boldsymbol{\theta} \in \Theta$ the probability increases that the desired solution can be represented by the parameters. A great amount of parameters may however lead to a very bad conditioned problem that might be very hard or even impossible to solve. Underfitting and overfitting are the terms describing this dilemma within the machine learning community.

Definition of the RKHS

For kernel regression, the solution space is always represented by a reproducing kernel Hilbert space. As the name of the method presumes, kernel regression is based on a certain function class called kernel functions. These functions correspond to an inner product in some feature space $\mathcal{H}$. The following statements remain unproven within this thesis but can be found in the book of Schölkopf and Smola [67].

Let ϕ denote a map

$$\phi : \mathrm{T} \to \mathcal{H}, \quad t \mapsto \phi(t), \tag{4.3}$$

from a compact subspace $\mathrm{T} \subset \mathbb{R}$ to an inner product space $\mathcal{H}$, also called the feature space. Then the kernel function can be defined via the inner product defined within $\mathcal{H}$

$$k(t, t') \triangleq \left\langle \phi(t), \phi(t') \right\rangle. \tag{4.4}$$

By defining different maps, multiple kernel functions can be derived. Thus, by using the kernel function, the inner product of a possibly high dimensional

feature space can be evaluated without explicitly performing the feature map itself. In fact, there are approaches that completely base on inner products. For these methods the kernel function is relevant while the explicit knowledge of the feature map is unessential. But how to design a kernel function without dealing with the map from Eq. (4.3)? Moreover, which functions allow a representation of the form Eq. (4.4)?

Definition 3. *(Gram Matrix)*
Given a function $k : \mathrm{T}^2 \to \mathbb{R}$ *and patterns* $t_1, ..., t_n \in \mathrm{T}$*, the* $n \times n$ *matrix* $\boldsymbol{K}$ *with elements*

$$[\boldsymbol{K}]_{ij} \triangleq k(t_i, t_j)$$

is called the Gram matrix *(or* kernel matrix*) of* k *with respect to* $t_1, ..., t_n$*.*

The function k is called positive definite if and only if for any sampling $t_1, ..., t_n$ all eigenvalues of the corresponding Gram matrix are positive. Basically, it is the class of positive definite functions that allow a representation as in Eq. (4.4). Originally revealed in 1909 by the British mathematician *James Mercer*, the so called *Mercer's theorem* even allows to explicitly construct the map from Eq. (4.3) for any positive definite function [71, p. 230]. For simplicity an explicit construction of the RKHS is taken from [67, p. 32].

Let ϕ denote the map depicted in Fig. 4.2.

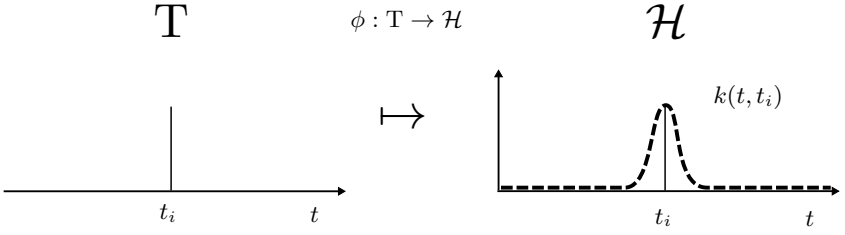

Figure 4.2.: The kernel map

Furthermore let $t_1, ..., t_n$ and $t'_1, ..., t'_m$ be sample times and

$$f(t) = \sum_{i=1}^{n} \alpha_i k(t, t_i), \quad g(t) = \sum_{j=1}^{m} \beta_j k(t, t'_j) \tag{4.5}$$

be two elements of a function space $\mathcal{H}$ that is spanned by a positive definite k. With an inner product that is defined by

$$\langle f, g \rangle_{\mathcal{H}} \triangleq \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(t_i, t_j).$$

this becomes a pre-Hilbert space with the characteristic

$$\langle f, k(.,t)\rangle_{\mathcal{H}} = f(t) \tag{4.6}$$

that is called reproducing property of the kernel function. The proof of Eq. (4.6) is trivial and can be obtained by evaluation. In particular, this definition constructs a space where Eq. (4.4)

$$\left\langle k(.,t), k(.,t')\right\rangle_{\mathcal{H}} = k(t',t) = k(t,t') = \left\langle \phi(t), \phi(t')\right\rangle_{\mathcal{H}}$$

is fulfilled. For any positive definite kernel k such a mapping can be constructed. The completion of all elements that can be represented by Eq. (4.5) forms the Hilbert space $\mathcal{H}$ that is also called reproducing Hilbert space.

Definition 4. *(Reproducing Kernel Hilbert Space [67, p. 36])*
Let T *be a nonempty set (often called the index set) and* $\mathcal{H}$ *a Hilbert space of functions* $f : \mathrm{T} \to \mathbb{R}$*. Then* $\mathcal{H}$ *is called a reproducing kernel Hilbert space endowed with the dot product* $\langle ., .\rangle_{\mathcal{H}}$ *(and the norm* $||f||_{\mathcal{H}} := \sqrt{\langle f, f\rangle_{\mathcal{H}}}$*) if there exists a function* $k : \mathrm{T} \times \mathrm{T} \to \mathbb{R}$ *with the following properties.*
1. k *has the reproducing property* [1]

$$\langle f, k(.,t)\rangle_{\mathcal{H}} = f(t) \quad \forall f \in \mathcal{H},$$

in particular,

$$\left\langle k(.,t), k(.,t')\right\rangle_{\mathcal{H}} = k(t,t').$$

2. k *spans* $\mathcal{H}$*, i.e.* $\mathcal{H} = \overline{span\{k(.,t)|t \in \mathrm{T}\}}$*, where* $\overline{X}$ *denotes the completion of the set* X*.*

Fig. 4.3 shows an element of a reproducing kernel Hilbert space spanned by the Gaussian kernel

$$k(t,t') \triangleq e^{-\frac{1}{8}(t-t')^2},$$

with parameters as defined in Tab. 4.1.

i	1	2	3	4	5
t_i	2.0	3.0	4.5	4.1	5.3
α_i	-0.5	1.6	0.2	0.15	0.31

Table 4.1.: Index set $t_1, ..., t_5$ and coefficients $\alpha_1, ..., \alpha_5$

[1] Note that this implies that each $f \in \mathcal{H}$ is actually a single function whose values at any $t \in \mathrm{T}$ are well-defined. In contrast, $\mathbf{L}^2$ Hilbert spaces usually do not have this property. The elements of these spaces are equivalence classes of functions that disagree only on sets of measure 0; see footnote 15 in Section B.3 [67].

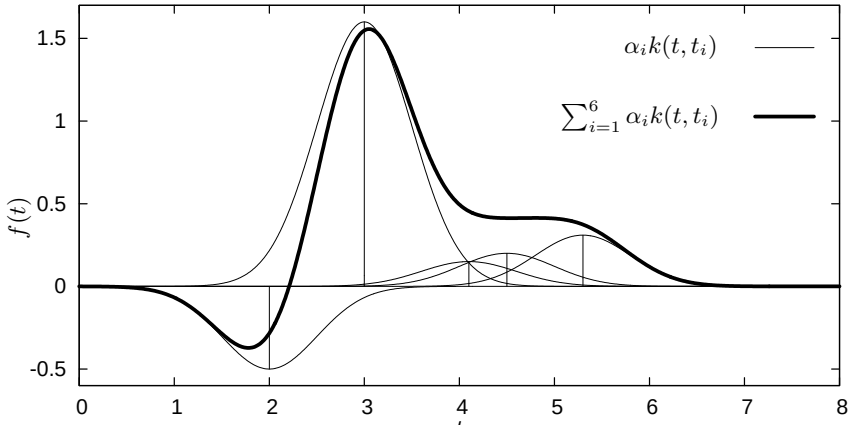

Figure 4.3.: Element of a RKHS

Representing all elements of the RKHS

So far, the elements of $\mathcal{H}$ were written as a finite sum of weighted kernel functions. However, there are elements in $\mathcal{H}$ where this is not possible. A more generic approach of expressing the elements of $\mathcal{H}$ can be found in [72].

Let $\mathcal{G}$ be the space of all functions that can be constructed by

$$\mathcal{G} \triangleq \left\{ f(t) \triangleq \int_{\mathrm{T}} k(t,\tau)\alpha(\tau)\mathrm{d}\tau \right\}. \tag{4.7}$$

Then $\mathcal{G}$ is dense in $\mathcal{H}$ if $\alpha \in \mathbf{L}^1(\mathrm{T}, d\tau) \cup \mathcal{M}_D$. Here, $\mathbf{L}^1(\mathrm{T}, d\tau)$ denotes the space space of integrable functions,

$$\mathbf{L}^1(\mathrm{T}) = \left\{ \alpha : \mathrm{T} \to \mathbb{R} : \int_{\mathrm{T}} |\alpha(\tau)|\mathrm{d}\tau < \infty \right\}$$

and $\mathcal{M}_D$ the space of discrete measures

$$\mathcal{M}_D \triangleq \left\{ \alpha = \sum_i c_i \delta_{t_i} : \{c_i\} \subset \mathbb{R}, \{t_i\} \subset \mathrm{T}, \sum_i |c_i| < \infty \right\} \tag{4.8}$$

for a compact set T. δ_t denotes the Dirac delta supported at $t \in \mathrm{T}$ in Eq. (4.8). Note, the sum in Eq. (4.8) may be infinite. With this notation, the norm of a

function within the RKHS becomes

$$||f||^2_{\mathcal{H}} = \int_{\mathrm{T}} \int_{\mathrm{T}} k(\tau_1, \tau_2)\alpha(\tau_1)\alpha(\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2. \tag{4.9}$$

Translation invariant kernels

For the class of translation invariant kernel functions with the property

$$k(t_i, t_j) = k'(t_i - t_j),$$

the norm from Eq. (4.9) may also be formulated within the frequency domain. Let $k(t_i, t_j)$ denote a translation invariant kernel, then by Eq. (4.7) the element of the RKHS spanned by k can be expressed by the convolution integral

$$f(t) \triangleq \int_{\mathrm{T}} k(t, \tau)\alpha(\tau)\mathrm{d}\tau = \int_{\mathrm{T}} k'(t - \tau)\alpha(\tau)\mathrm{d}\tau. \tag{4.10}$$

Let further

$$F(\omega) \triangleq \mathfrak{F}\{f\}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-\mathrm{i}\omega t}\mathrm{d}t,$$

$K(\omega) \triangleq \mathfrak{F}\{k'(t)\}(\omega)$ and $A(\omega) \triangleq \mathfrak{F}\{\alpha\}(\omega)$ denote the Fourier transformations of $f(t)$, $k'(t)$ and $\alpha(t)$. Then, Eq. (4.10) corresponds to

$$F(\omega) = K(\omega) \cdot A(\omega)$$

after transformation into the frequency domain [18, p. 679]. Disregarding the constant factor 2π, the Fourier transformation is an isometric isomorphism within the space of square integrable functions [71, p. 171], i.e.

$$\langle f, \alpha \rangle_{\mathbf{L}^2} \triangleq \int_{\mathrm{T}} f(\tau)\alpha(\tau)\mathrm{d}\tau = \frac{1}{2\pi} \int_{\Omega} F(\omega)\overline{A(\omega)}\mathrm{d}\omega.$$

The norm within the RKHS can therefore be written as

$$||f||^2_{\mathcal{H}} = \int_{\mathrm{T}} \int_{\mathrm{T}} k(\tau_1, \tau_2)\alpha(\tau_1)\alpha(\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 = \frac{1}{2\pi} \int_{\Omega} \frac{F(\omega)\overline{F(\omega)}}{K(\omega)}\mathrm{d}\omega. \tag{4.11}$$

In frequency domain, the Fourier transformation of the kernel function displays the filtering properties of the kernel. If $f \in \mathcal{H}$ contains frequencies that are only sparsely contained within the kernel function, the norm of the element becomes very big. This filtering property of the norm is used to restrict the space of functions in the following chapters.

4.1.3. Regularization

For kernel regression, the solution space is always represented by a reproducing kernel Hilbert space. This space is usually very large and does not define a fixed set of parameters to optimize. However, the basic idea in kernel regression is to restrict the search space to a compact set and then apply the following theorem.

Theorem 2. *(Operator Inversion Lemma [67, p. 88])*
Let X be a compact set and let the map $f : X \to Y$ be continuous. Then there exists an inverse map $f^{-1} : f(X) \to X$ that is also continuous.

Instead of directly choosing a compact subset of $\mathcal{H}$ the formulation of the minimization problem is added by a regularization term $\Omega[f]$ that leads to an improved conditioning of the problem

$$R_{\text{reg}}[f] \triangleq R_{\text{emp}}[f] + \mu\Omega[f]. \tag{4.12}$$

The resulting functional is called *regularized risk functional*. This is motivated by the idea that the regularization term causes consistency by penalizing functions that aren't smooth. $\Omega[f]$ denotes a term controlling the capacity of the function space and μ is a modeling parameter balancing the capacity versus the cost function $R_{\text{emp}}[f]$. This implies additional information on the solution since the regularizing term quantifies the non smooth content of the function. Therefore, the general assumption here is that all possible solutions $f(t)$ exhibit smoothness.

Recall, the solution space $f \in \mathcal{H}$ is constructed to be a RKHS. The basic idea of kernel regression is to express the regularization term $\Omega[f]$ by means of the norm in the RKHS. If the assumed smoothness property of the problem can be expressed by means of a norm within the RKHS, then the minimizer of Eq. (4.12) can be expressed by a finite dimensional vector.

Theorem 3. *(Representer Theorem [73])*
Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function, by T a set, and by $c : (\mathrm{T} \times \mathbb{R}^2)^n \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer $f \in \mathcal{H}$ of the regularized risk

$$c\left((t_1, z_1, \bar{z}_1), ..., (t_n, z_n, \bar{z}_n)\right) + \Omega(||f||_{\mathcal{H}}) \tag{4.13}$$

admits a representation of the form

$$f(t) = \sum_{i=1}^{n} \alpha_i k(t_i, t).$$

Although the solution space $\mathcal{H}$ is an infinite dimensional space of functions, a regularizer defined as in Eq. (4.13) allows to reduce the search space to a finite dimensional parameter space denoted by $\boldsymbol{\theta} \triangleq \boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_n]^{\mathrm{T}} \in \mathbb{R}^n$. Consequently, the squared norm $||f||^2_{\mathcal{H}}$ can be simplified to

$$||f||^2_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} \alpha_i k(t_i, .), \sum_{j=1}^{n} \alpha_j k(t_j, .) \right\rangle_{\mathcal{H}} =$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle k(t_i, .), k(t_j, .) \rangle_{\mathcal{H}} = \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha},$$

with $\boldsymbol{K}$ denoting the Gramian matrix. With the definition $\Omega : \mathbb{R} \to \mathbb{R}, \quad \xi \mapsto \xi^2$, the final parameterized minimization problem then becomes

$$R_{\mathrm{reg}}[f] \triangleq R_{\mathrm{emp}}[f] + \mu \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}. \tag{4.14}$$

Eq. (4.14) can now be minimized by standard optimization algorithms. However note, the kernel function has to be chosen properly for good performance.

4.2. Kernel design

The regularizing term $\mu \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}$, respectively the kernel function $k(.,.)$, is the essential design element for kernel regression. Therefore, the filtering property of the kernel function needs to reflect true information. In this section, a kernel function is designed with respect to the ROLAM problem.

4.2.1. Physical motivation

Let

$$\boldsymbol{x} : \mathbb{R} \to \mathbb{R}^3, \quad t \mapsto \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

denote a three dimensional trajectory describing the movement of the target. As before the denotation $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$ is used for simplification.

Any trajectory is a solution of a differential equation

In Chapter 2.1 the localization problem was already formulated in form of a dynamic system that implies a system of differential equations. A justification for this can be found in the fact that $\boldsymbol{x}$ is representing a curve describing a movement. In the general case, the set of differential equations describing the movement of a complex mechanical system can be found by using *Hamilton's*

principle. [74]. For a single particle modelled as a point mass, the movement can directly be described by Newton's law's of motion

$$m\ddot{\boldsymbol{x}}(t) = \boldsymbol{f}_a(t).$$

In this notation $\boldsymbol{f}_a(t) \in \mathbb{R}^3$ covers all relevant forces like control forces and dissipative forces and $m \in \mathbb{R}$ the mass of the moving node. For the special case of a moving particle subject to friction

$$\boldsymbol{f}_a(t) \triangleq \boldsymbol{f}(t) - \beta\dot{\boldsymbol{x}}(t)$$

the dynamic model

$$m\dot{\boldsymbol{v}}(t) = -\beta\boldsymbol{v}(t) + \boldsymbol{f}(t), \quad \boldsymbol{v}(t) \triangleq \dot{\boldsymbol{x}}(t) \tag{4.15}$$

can be obtained. Here $\boldsymbol{f}(t)$ denotes the external driving forces and $\beta\boldsymbol{v}(t)$ covers the friction directly proportional to the velocity. *Langevin* used this equation to describe the Brownian motion of a particle moving within a fluid [75]. Although other types of friction exist, this thesis sticks to this model for simplification as the resulting differential equation can be described by a linear operator. Following this presumption, the external force $\boldsymbol{f}(t)$ is the origin for all possible motions. As a consequence, the trajectory has to be continuous and differentiable. Eq. (4.15) is a well known studied initial value problem. Provided, the driving force is known and integrable, the differential equation can be solved. For simplification let the driving forces be controlled independently in each direction. Note, this corresponds to forces that are directly controlling the target in the inertial frame. Then, Eq. (4.15) can be decoupled in each dimension and can be solved separately. This one-dimensional treatment is kept until the end of this chapter. Reduced to one dimension, Eq. (4.15) becomes

$$m\dot{v}(t) = -\beta v(t) + f(t). \tag{4.16}$$

The corresponding, one-dimensional motion model is depicted in Fig. 4.4. The

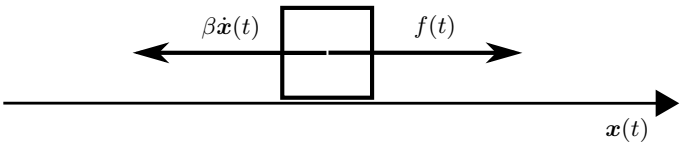

Figure 4.4.: Simple one-dimensional motion model

solution can be expressed as the sum of the homogeneous solution and a particular solution

$$v(t) = v^p(t) + v^h(t).$$

The homogeneous solution can be calculated by the roots of the characteristic polynomial

$$\mathrm{CP}(\lambda) = m\lambda + \beta,$$

which results in

$$v^h(t) = c_0 \cdot e^{-\frac{\beta t}{m}}, \quad c_0 \in \mathbb{R}.$$

The particular solution is calculated using a *Greens function*. Let $\mathcal{L}$ be defined as the linear differential operator

$$\mathcal{L} : \mathcal{V} \to \mathcal{F}, \quad v(t) \mapsto f(t) = m\dot{v}(t) + \beta v(t) \tag{4.17}$$

mapping from the space of velocities $\mathcal{V}$ to the space of forces $\mathcal{F}$. Then the Greens function can be obtained by applying the Fourier transformation. Rewritten in frequency domain, Eq. (4.15) becomes

$$V(\omega) = \frac{1}{m\mathrm{i}\omega + \beta} F(\omega),$$

with

$$F(\omega) \triangleq \mathfrak{F}\{f(t)\} \quad \text{and} \quad V(\omega) \triangleq \mathfrak{F}\{v(t)\}.$$

After defining the step function

$$H(t) \triangleq \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases}$$

the Green's function $g_{\mathcal{L}}(t)$ of $\mathcal{L}$ can be explicitly written as

$$g_{\mathcal{L}}(t) = \sqrt{2\pi} \cdot \mathfrak{F}^{-1}\left\{\frac{1}{m\mathrm{i}\omega + \beta}\right\} = e^{-\frac{\beta t}{m}} \frac{H(t)}{m}.$$

The particular solution can be obtained by the convolution integral

$$v^p(t) = \int_{-\infty}^{\infty} g_{\mathcal{L}}(t-\tau) f(\tau) \mathrm{d}\tau. \tag{4.18}$$

Added to the homogeneous solution, the final solution for the differential equation becomes

$$v(t) = c_0 \cdot e^{-\frac{\beta t}{m}} + \int_{-\infty}^{\infty} g_{\mathcal{L}}(t-\tau) f(\tau) \mathrm{d}\tau.$$

With an initial value for the starting position $x_0 \triangleq x(0)$ and the starting velocity $v_0 \triangleq v(0)$ the trajectory can be obtained by integration of the velocity function

$$x(t) = x_0 + \int_0^t v(\tau)\mathrm{d}\tau. \tag{4.19}$$

As mentioned before the trajectory becomes a continuous and differentiable function.

An unknown force as the driver

From the former paragraph it follows that the trajectory can be uniquely determined by Eq. (4.19) if the force function $f(t)$ is known and if starting conditions for $x(0)$ and $v(0)$ are available. Unfortunately, no information on the controlling forces is available for the problem at hand. However, for all practical applications, the forces are subject to physical processes and therefore limited in power.

Bounding the power of $f(t)$ is modelled by postulating a *Power Spectral Density* (PSD) for the force function. To this end, let $f(t)$ be an element of the space of square integrable functions,

$$\mathbf{L}^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{R} : \int_{-\infty}^{\infty} |f(\tau)|^2 \mathrm{d}\tau < \infty \right\}.$$

Then, a Fourier transformation of $F(\omega) = \mathfrak{F}\{f\}(w)$ and hence the PSD $\boldsymbol{S}_f(\omega) = F(\omega)\overline{F(\omega)}$ exists. The typical assumption is, that high frequencies are less represented than low frequencies. However, this is a matter of design and may be chosen differently for different applications. Within this thesis the parameterized model

$$\boldsymbol{S}_f(\omega) \triangleq a^2 e^{-\frac{\omega^2}{b^2}} \tag{4.20}$$

is chosen. For more generality, the model parameters $a, b \in \mathbb{R}$ are introduced. In Fig. 4.5, the PSD is illustrated for $a \triangleq 1$ and $b \triangleq 2$.

With this limitation for the force functions and with the limitation coming from the differential Eq. (4.16) the search space for trajectories can be reduced. Trajectories that come into consideration are solutions to Eq. (4.16) driven by an unknown force function $f(t)$ with a PSD close to Eq. (4.20). Because the force function is unknown the whole scenario becomes a system with a stochastic input. Consequently, the next section deals with the stochastic properties of forces, velocities and trajectories.

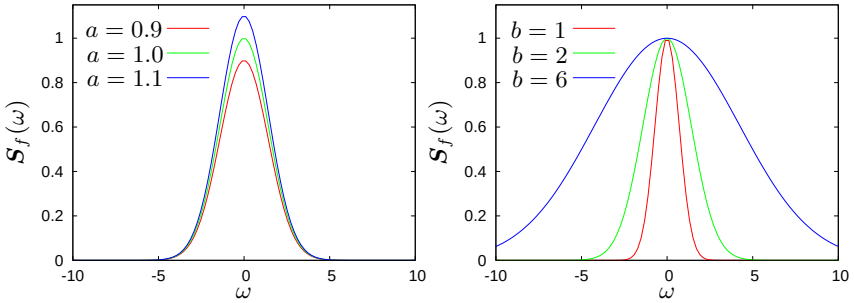

Figure 4.5.: power spectrum model for $f(t)$ with different parameters

4.2.2. Stochastic approach

The derivations from the former section are now used to construct stochastic processes for forces, velocities and trajectories such that they exhibit the properties postulated in Sec. 4.2.1. All processes can be constructed to be Gaussian processes. This way, the stochastic properties of any sampling vector of the trajectories can be described as a multivariate Gaussian distribution. This serves as a prior distribution for the localization problem.

Gaussian Processes

Definition 5. *A continuous random process $y(t, \varsigma)$ with $t \in \mathbb{R}$ and ς specifying the sample space is said to be Gaussian if for every sequence of time instants $t_1, ..., t_n$ and $n \in \mathbb{N}$, the random variables $y(t_1), ..., y(t_n)$ are jointly normally distributed.*

By definition, a Gaussian process $y(t, \varsigma)$ is completely specified by its mean function and covariance function that is denoted as follows

$$\boldsymbol{\mu}_y(t) = \mathrm{E}[y(t,\varsigma)]\,, \tag{4.21}$$

$$\boldsymbol{r}_y(t_i, t_j) = \mathrm{E}\big[(y(t_i,\varsigma) - \boldsymbol{\mu}_y(t_i))(y(t_j,\varsigma) - \boldsymbol{\mu}_y(t_j))\big]\,. \tag{4.22}$$

The probability density function is well defined and can be consistently extended to an infinite number of variables as stated in the *Kolmogorov extension theorem* [76]. Thus, for every sequence of time instants $t_1, ..., t_n$ the probability density function of the vector $\boldsymbol{y} = [y(t_1) \cdots y(t_n)]^{\mathrm{T}}$ can be explicitly written as

$$\mathrm{p}(\boldsymbol{y}) = \frac{1}{(\sqrt{2\pi})^n |\boldsymbol{\Sigma}_y|^{0.5}} e^{-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}} \boldsymbol{\Sigma}_y{}^{-1} \boldsymbol{y}} \tag{4.23}$$

with the covariance matrix $[\boldsymbol{\Sigma}_y]_{ij} \triangleq \boldsymbol{r}_y(t_i, t_j)$.

Definition 6. *A continuous random process $y(t,\varsigma)$ is called strict-sense stationary (*SSS*) if its statistical properties are invariant to a shift of the origin. The processes $y(t,\varsigma)$ and $y(t+\tau,\varsigma)$ with $\tau > 0$ then have the same statistics. A continuous random process $y(t,\varsigma)$ is called wide-sense stationary (*WSS*) if the first two moments are invariant to a shift. The processes $y(t,\varsigma)$ and $y(t+\tau,\varsigma)$ with $\tau > 0$ then have the same mean and covariance.*

Note, not all Gaussian processes are stationary. At least, the following remark is worth a note.

Remark 1. *All Gaussian processes that are WSS also are SSS as the whole statistics is covered through mean and covariance.*

Because all processes observed within this thesis are Gaussian processes no further distinction is made between SSS and WSS. In most cases the processes are either called stationary or not stationary. For stationary Gaussian processes the autocorrelation function allows a representation of the form

$$\boldsymbol{r}_y(t_i,t_j) = \boldsymbol{r}'_y(t_i - t_j).$$

Inference of Gaussian processes

In Sec. 4.2.1 the solution for the trajectory was obtained in two steps. The velocity $v^p(t)$ was calculated by convolving the driving force $f(t)$ with the Greens function $g_{\mathcal{L}}(t)$ and the trajectory $x^p(t)$ by integration of $v^p(t)$. This is repeated now for the stochastic processes $f(t,\varsigma)$ and $v^p(t,\varsigma)$ to obtain $x^p(t,\varsigma)$.

Consider a stochastic process $f(t,\varsigma)$ evolved by applying a linear operator $\mathcal{T}$ on another stochastic process $y(t,\varsigma)$ that only operates on the variable t treating ς as a parameter

$$f(t,\varsigma) = \mathcal{T}y(t,\varsigma).$$

Such a system is called *deterministic*. Furthermore, if the input process is a Gaussian process the output is also a Gaussian process [77, p. 191]. Note, the property of stationarity can not be conserved for all kind of linear operators. However, let $\mathcal{T}$ denote a *Linear Time Invariant* (LTI) system that can explicitly be written as a convolution

$$\mathcal{T}y \triangleq y * g \triangleq \int_{-\infty}^{\infty} y(t-\tau,\varsigma)g(\tau)\mathrm{d}\tau. \tag{4.24}$$

If the input of such an operator is a stationary Gaussian process then the resulting process is a stationary Gaussian process [25, p. 398]. Thus, by starting with an initial Gaussian process, other Gaussian processes can be constructed by

applying linear operators of this kind. The expectation of the resulting process $f(t,\varsigma)$ can be determined as follows:

$$\boldsymbol{\mu}_f(t) \triangleq \mathrm{E}[f(t,\varsigma)] = \mathrm{E}[\mathcal{T}y(t,\varsigma)] = \mathcal{T}\mathrm{E}[y(t,\varsigma)] = \mathcal{T}\boldsymbol{\mu}_y(t)$$

for a linear operator $\mathcal{T}$ [25, p. 398]. The autocorrelation function of the resulting process can by obtained by the *Wiener-Khinchin theorem* which states that the PSD of a WSS process $f(t,\varsigma)$, real or complex, is the Fourier transformation of its autocorrelation function [37, 25]

$$\boldsymbol{S}_f(\omega) = \mathfrak{F}\left\{\boldsymbol{r}'_f(t)\right\} = \int_{-\infty}^{\infty} \boldsymbol{r}'_f(t)e^{-\mathrm{i}\omega t}\mathrm{d}t.$$

With $F(\omega) \triangleq \mathfrak{F}\{f(t,\varsigma)\}$, $Y(\omega) \triangleq \mathfrak{F}\{y(t,\varsigma)\}$ and $G(\omega) \triangleq \mathfrak{F}\{g(t)\}$, the convolution in Eq. (4.24) becomes a product

$$F(\omega) = G(\omega) \cdot Y(\omega)$$

in frequency domain. The PSD of the output becomes

$$\boldsymbol{S}_f(\omega) = \mathrm{E}\left[F(\omega)\overline{F(\omega)}\right] = \mathrm{E}\left[G(\omega)Y(\omega)\overline{G(\omega)Y(\omega)}\right] = \boldsymbol{S}_y(\omega)G(\omega)\overline{G(\omega)}$$

and therefore

$$\boldsymbol{r}_y(t_i,t_j) = \mathfrak{F}^{-1}\left\{\boldsymbol{S}_y(\omega)G(\omega)\overline{G(\omega)}\right\}. \tag{4.25}$$

Integrability and Continuity of stochastic processes

The trajectory can be evolved from the velocity by integration.

Definition 7. *(integrability) [25, p. 428]*
A process $y(t,\varsigma)$ is integrable in the mean square sense if the limit

$$\int_0^t y(\tau,\varsigma)\mathrm{d}\tau = \lim_{\Delta t_i \to 0} \sum_i y(t_i,\varsigma)\Delta t_i$$

exists in the mean square sense.

The integrability of the operator can be verified using the following theorem.

Theorem 4. *[25, p. 428]*
The process $y(t,\varsigma)$ is integrable in the mean square sense if

$$\int_0^t \int_0^t |\boldsymbol{r}_y(t_1,t_2)|\mathrm{d}t_1\mathrm{d}t_2 < \infty$$

with $\boldsymbol{r}_y(t_1,t_2) \triangleq \mathrm{E}\left[(y(t_1,\varsigma) - \boldsymbol{\mu}_y(t_1))(y(t_2,\varsigma) - \boldsymbol{\mu}_y(t_2))\right]$ denoting the autocorrelation function of the stochastic process.

Also, continuity can be defined for stochastic processes.

Definition 8. *(continuity) [25, p. 426]*
A process $y(t,\varsigma)$ is continuous in the mean square sense if

$$\mathrm{E}\left[(y(t+\epsilon,\varsigma)-y(t,\varsigma))^2\right] \overset{\epsilon\to 0}{\longrightarrow} 0$$

exists in the mean square sense.

This is fulfilled as long as the autocorrelation function of the process is continuous.

The force process

By the assumptions from Sec. 4.2.1 the target is driven by a force with a PSD specified by Eq. (4.20). Although the true force is unknown its statistical properties can be described by a stationary stochastic process. Let $n(t,\varsigma)$ be the Gaussian white noise process with its statistical properties

$$\boldsymbol{\mu}_n(t) = 0 \tag{4.26}$$

$$\boldsymbol{r}_n(t_i,t_j) = \frac{1}{\nu}\delta(t_i-t_j) \tag{4.27}$$

and $\nu > 0$ modeling the "intensity" of the noise.[1] Let further $\mathcal{T}$ define the linear operator

$$\mathcal{T}:\mathcal{N}\to\mathcal{F}, \quad n(t,\varsigma)\mapsto(n*h)(t,\varsigma) \tag{4.28}$$

with

$$h(t) \triangleq \frac{ab\sqrt{\nu}}{\sqrt{2\pi}} e^{-\frac{t^2b^2}{2}}$$

and "$*$" denoting the convolution as defined in Eq. (4.24). Then the output of the linear operator $\mathcal{T}$ is a stochastic process $f(t,\varsigma)$ with the requested PSD from Eq. (4.20). This procedure, called *prewhitening*, is depicted in Fig. 4.6 [38].

The PSD of the result can be verified as follows. The operator defined in Eq. (4.28) is of the form from Eq. (4.24) defined in the former section. Therefore, the output of such a system is a stationary Gaussian process and fully described by its expectation and covariance. The expectation of the resulting process $f(t,\varsigma)$ can be determined to be

$$\boldsymbol{\mu}_f(t) = \mathrm{E}[f(t,\varsigma)] = \mathrm{E}[\mathcal{T}n(t,\varsigma)] = \mathcal{T}\mathrm{E}[n(t,\varsigma)] = 0.$$

[1] Following [76] (page 21) there does not exist any "reasonable" stationary stochastic process with continuous paths satisfying Eq. (4.26). Nevertheless it is possible to represent $n(t,\varsigma)$ as a generalized stochastic process that can be constructed as a probability measure on a larger space than $\mathbb{R}^{[0,\infty)}$.

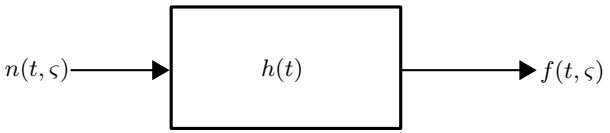

Figure 4.6.: Shaping filter with white noise $n(t,\varsigma)$ as input

The covariance and accordingly the autocorrelation function of $f(t,\varsigma)$ needs to satisfy Eq. (4.25). The postulated PSD in Eq. (4.20) therefore is determined by

$$\boldsymbol{r}_f(t_i,t_j) = \mathfrak{F}^{-1}\left\{a^2 e^{-\frac{\omega^2}{b^2}}\right\} = \mathfrak{F}^{-1}\left\{\boldsymbol{S}_n(\omega)H(\omega)\overline{H(\omega)}\right\}$$

with

$$H(\omega) = \mathfrak{F}\left\{h(t)\right\} = a\sqrt{\nu}e^{-\frac{\omega^2}{2b^2}}.$$

As $\boldsymbol{S}_n(\omega) = \nu^{-1}$, this complies with Eq. (4.25)

$$a^2 e^{-\frac{\omega^2}{b^2}} = \frac{1}{\nu}H(\omega)\overline{H(\omega)}.$$

By close inspection, the design parameters a and $\sqrt{\nu}$ can be considered to be reciprocal and as such can be reduced to one parameter. Without loss of generality, ν is defined as $\nu \triangleq 1$ and a is kept as parameter. As the PSD defined in Eq. (4.20) is square integrable,

$$\int_{\Omega} \boldsymbol{S}_f(\omega)^2 \mathrm{d}\omega < \infty,$$

the inverse Fourier transformation of the PSD is well defined. The autocorrelation function of $f(t,\varsigma)$ then finally becomes

$$\boldsymbol{r}_f(t_i,t_j) = \mathfrak{F}^{-1}\left\{a^2 e^{-\frac{\omega^2}{b^2}}\right\} = \frac{a^2 b}{2\sqrt{\pi}}e^{-\frac{(t_i-t_j)^2 b^2}{4}}.$$

As the input process $n(t,\varsigma)$ is stationary the output process $f(t,\varsigma)$ is stationary as well. Moreover, the resulting process is continuous in mean square

$$\mathrm{E}\left[\left(f(t+\epsilon,\varsigma) - f(t,\varsigma)\right)^2\right] = 2\boldsymbol{r}_f(0) - 2\boldsymbol{r}_f(\epsilon) \overset{\epsilon\to 0}{\longrightarrow} 0$$

as the autocorrelation function is continuous and integrable.

The velocity process

After defining the stochastic process $f(t,\varsigma)$ representing the force, the next step is to find a solution for the differential Eq. (4.16) with $f(t,\varsigma)$ as the driving force. By the stochastic input, Eq. (4.16) becomes a *Stochastic Differential Equation* (SDE). The solution of this equation is again a stochastic process.

For the special parameterization

$$a \triangleq 1, \quad b^2 \triangleq \frac{2}{\epsilon} \quad \text{with} \quad \epsilon \to 0, \tag{4.29}$$

which renders the force process to be white Gaussian noise, the SDE becomes the famous *Ornstein-Uhlenbeck equation* or also called *Langevin equation* [76, 75, 38]

$$dv^p(t,\varsigma) = -\frac{\beta}{m} v^p(t,\varsigma) + \frac{1}{m} n(t,\varsigma).$$

The solution for this SDG is a stochastic process called *Ornstein-Uhlenbeck process*.

For an arbitrary choice of a and b the statistical properties of the velocity process can be obtained as follows. The particular solution for the velocity can be calculated by applying the linear operator $\mathcal{L}$ from Eq. (4.17). With $f(t,\varsigma)$ as an integrable process, Eq. (4.18) can also be written as a convolution

$$\mathcal{L} : \mathcal{F} \to \mathcal{V}, \quad f(t,\varsigma) \mapsto (g_{\mathcal{L}} * f)(t,\omega).$$

With Eq. (4.28) and the associativity and commutativity property of convolutions

$$v^p = (g_{\mathcal{L}} * h) * n \triangleq w * n$$

a new operator

$$\mathcal{W} : \mathcal{N} \to \mathcal{V}, \quad n(t,\varsigma) \mapsto (w * n)(t,\varsigma)$$

can be defined. The transfer function w becomes[1]

$$w \triangleq g_{\mathcal{L}} * h = \frac{a}{2m} e^{-\frac{\beta(2b^2 tm - \beta)}{2m^2 b^2}} \left(1 + \operatorname{erf}\left(\frac{mb^2 t - \beta}{\sqrt{2}mb}\right)\right)$$

with the error function

$$\operatorname{erf}(t) \triangleq \frac{2}{\sqrt{\pi}} \int_0^t e^{-\tau^2} \mathrm{d}\tau.$$

The transfer function $w(t)$ is plotted in Fig. 4.7 for $a = 1$, $b = 2$, $m = 10$ and $\beta = 6$. $\mathcal{W}$ is a linear operator describing a deterministic system and can be

[1] The calculations that lead to this results can be found in Sec. A.1.1.

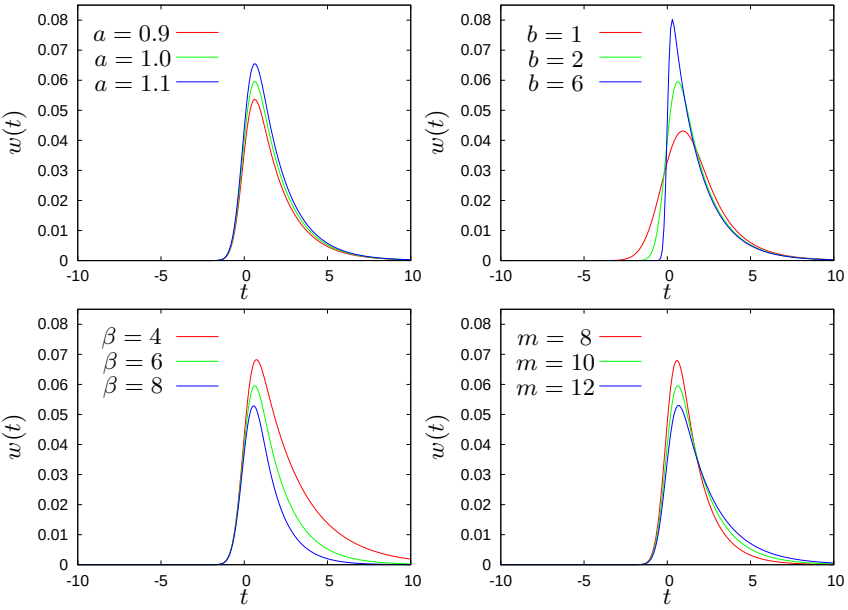

Figure 4.7.: Influence of the parameters a, b, β and m on $w(t)$

described by a convolution integral. The resulting stochastic process therefore is also Gaussian and can be completely described by its mean and covariance.

$$\boldsymbol{\mu}_{v^p}(t) = \mathrm{E}[v^p(t,\varsigma)] = \mathrm{E}[\mathcal{W}n(t,\varsigma)] = \mathcal{W}\mathrm{E}[n(t,\varsigma)] = 0.$$

Analogously to the former section, the autocorrelation function of $v^p(t,\varsigma)$ can be calculated using the Fourier transformation. With

$$W(\omega) = G(\omega)H(\omega) = \frac{a e^{-\frac{\omega^2}{2b^2}}}{m\mathrm{i}\omega + \beta}, \quad G(\omega) \triangleq \mathfrak{F}\{g_{\mathcal{L}}\}, \tag{4.30}$$

the autocorrelation function becomes

$$\boldsymbol{r}_{v^p}(t_i,t_j) = \mathfrak{F}^{-1}\left\{\boldsymbol{S}_n(\omega)H(\omega)\overline{H(\omega)}G(\omega)\overline{G(\omega)}\right\} = \mathfrak{F}^{-1}\left\{\frac{a^2 e^{-\frac{\omega^2}{b^2}}}{m^2\omega^2+\beta^2}\right\}$$

that finally results in[2]

$$\boldsymbol{r}_{v^p}(t_i,t_j) = c_0\left(e^{\frac{\beta\Delta t}{m}} f_1(\Delta t) + e^{\frac{-\beta\Delta t}{m}} f_2(\Delta t)\right),$$

[2]This result was found by using a computer algebra system.

with

$$\Delta t \triangleq t_i - t_j, \quad c_0 \triangleq \frac{a^2}{4\beta m} e^{\frac{\beta^2}{b^2 m^2}}$$

and

$$f_1(\Delta t) \triangleq \left(1 - \operatorname{erf}\left(\frac{b\Delta t}{2} + \frac{\beta}{bm}\right)\right), \quad f_2(\Delta t) \triangleq \left(1 + \operatorname{erf}\left(\frac{b\Delta t}{2} - \frac{\beta}{bm}\right)\right).$$

$v^p(t, \varsigma)$ is continuous in mean square as the autocorrelation function is continuous. Furthermore, $v^p(\tau, \varsigma)$ is integrable for $t \in [0, T]$ with $T < \infty$ because

$$\int_0^t \int_0^t |\boldsymbol{r}_{v^p}(t_1 - t_2)| \mathrm{d}t_1 \mathrm{d}t_2 < \infty.$$

In Fig. 4.8 the autocorrelation function is plotted for different data sets. A basic set, drawn in green color with the parameters $a = 1, b = 2, \beta = 6, m = 10$ was slightly modified to show the influence for every single parameter.

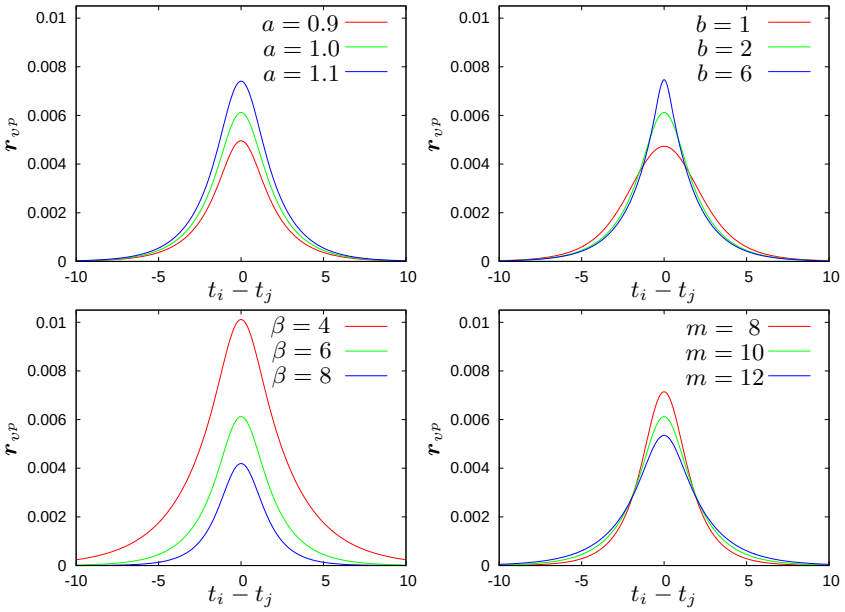

Figure 4.8.: Influence of a, b, β and m on the autocovariance

It's worth to note the connection to the former mentioned Ornstein Uhlenbeck process. For the parameterization proposed in Eq. (4.29), the resulting autocorrelation function consistently becomes the asymptotic and stationary part of the autocorrelation function for the Ornstein Uhlenbeck process [25, p. 448]

$$\boldsymbol{r}_{OU}(t_i, t_j) = \frac{1}{2\beta m} e^{-\left|\frac{\beta \Delta t}{m}\right|}. \tag{4.31}$$

The trajectory process

Eq. (4.19) shows the dependency of the trajectory on the velocity and the initial conditions. With the contruction of a stochastic process for the velocities that is integrable in the mean square sense the trajectory can be written as a stochastic integral

$$x(t, \varsigma) = x_0 + \int_0^t c_0 \cdot e^{-\frac{\beta \tau}{m}} \mathrm{d}\tau + \int_0^t v^p(\tau, \varsigma) \mathrm{d}\tau.$$

Disregarding the initial conditions the integral can be described by the linear operator

$$\mathcal{I} : \mathcal{V} \to \mathcal{X}, \quad v^p(t, \varsigma) \mapsto \int_0^t v^p(\tau, \varsigma) \mathrm{d}\tau.$$

However note, integration is a linear but not a time invariant operation. The integrated Gaussian process therefore is a non stationary Gaussian process representing a trajectory. The mean of the process $x^p(t, \varsigma)$ can be obtained by

$$\boldsymbol{\mu}_{x^p}(t) = \mathrm{E}[x^p(t, \varsigma)] = \mathrm{E}[\mathcal{I} v^p(t, \varsigma)] = \mathcal{I}\mathrm{E}[v^p(t, \varsigma)] = 0.$$

The covariance of the process can be found to be[1]

$$\begin{aligned} \boldsymbol{r}_{x^p}(t_1, t_2) \triangleq \mathrm{E}[x(t_1, \varsigma) x(t_2, \varsigma)] = \int_0^{t_1} \int_0^{t_2} \boldsymbol{r}_{v^p}(\tau_2, \tau_1) \mathrm{d}\tau_1 \mathrm{d}\tau_2 = \\ = R(t_1) + R(t_2) - R(t_2 - t_1) \end{aligned} \tag{4.32}$$

with the definition

$$R(t) \triangleq \int_0^t \kappa(\tau_1) \mathrm{d}\tau_1 = \int_0^t \int_0^{\tau_1} \boldsymbol{r}'_{v^p}(\tau_2) \mathrm{d}\tau_2 \mathrm{d}\tau_1 = R(-t). \tag{4.33}$$

[1] The calculations that lead to this results can be found in Sec. A.1.2.

By Eq. (4.32), $\boldsymbol{r}_{x^p}(t_1, t_2) \neq \boldsymbol{r}_{x^p}(t_1 + \mu, t_2 + \mu)$ with $\mu > 0$ and thus the process is not stationary. This can be seen by comparing the covariance functions for $v^p(t, \varsigma)$ and $x^p(t, \varsigma)$. For the parameter set $a = 1, b = 2, \beta = 6$ and $m = 10$ these are plotted in Fig. 4.9.

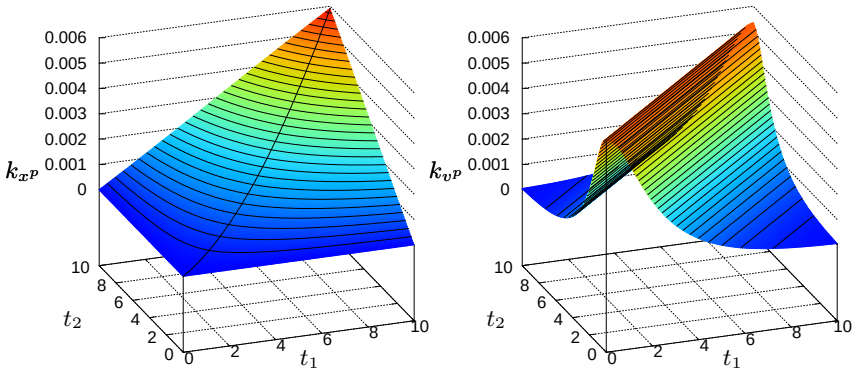

Figure 4.9.: Covariance matrices of trajectory and velocity for $0 \leq t_1, t_2 \leq 10$

The following plots from Fig. 4.10 show the influence of the single model parameters on the autocovariance function. Here, only the diagonal elements of the covariance matrix are plotted. Again, the default parameters are chosen to be $a = 1, b = 2, \beta = 6$ and $m = 10$.

With this model for a trajectory in form of a Gaussian stochastic process the probability density function of any sampling sampling vector is well defined and offers a prior distribution.

4.2.3. Embedding the solution into a RKHS

In Sec. 4.2.2 a probabilistic model for arbitrary sampling vectors of trajectories was generated. This model allows to distinguish between plausible and implausible trajectories and thus serves as a prior. In this section an appropriate search space for trajectories is defined. It turns out that reproducing kernel Hilbert spaces suit perfectly for this purpose.

A RKHS of velocities

As the objective is to determine a trajectory it is straightforward to search for a solution space that contains all relevant realizations from $x^p(t, \varsigma)$. Because $x^p(t, \varsigma)$ is not stationary and only defined for $t \geq 0$ this is very inconvenient.

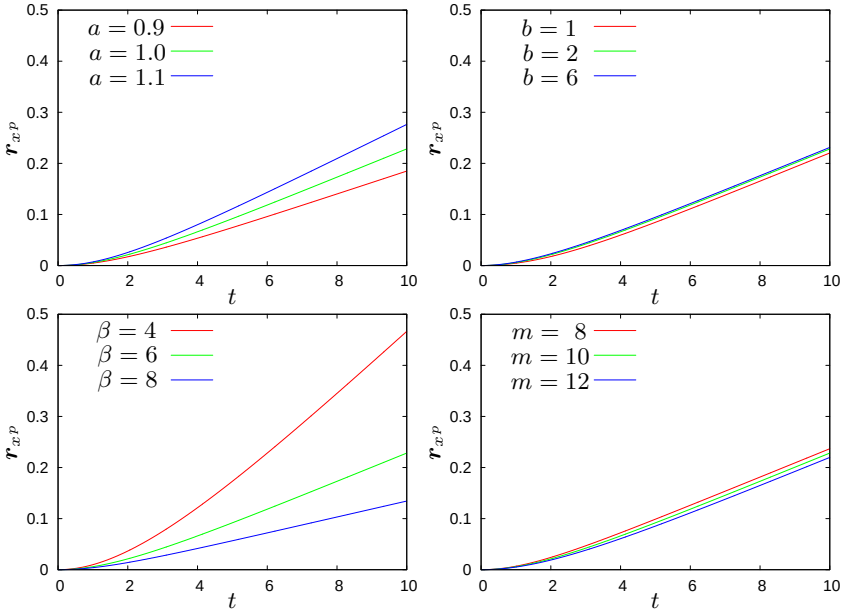

Figure 4.10.: Influence of a, b, β and m on the autocovariance

However, if one disregards the initial condition, there exists an one to one mapping between trajectories and velocities. Therefore it is equivalent to search within the space of velocities and then integrate the solution to obtain the trajectory.

Consider the Gaussian process $v^p(t, \varsigma)$ as defined in Sec. 4.2.2 with the autocorrelation function $\boldsymbol{r}_{v^p}(t_i, t_j)$. Then for any sampling $t_1, ..., t_n$ the covariance matrix $\boldsymbol{\Sigma}_{v^p}$ with $[\boldsymbol{\Sigma}_{v^p}]_{ij} = \boldsymbol{r}_{v^p}(t_i, t_j)$ is positive definite. $\boldsymbol{r}_{v^p}(t_i, t_j)$ can therefore be used as a kernel function.

Let

$$k(t_i, t_j) \triangleq \boldsymbol{r}_{v^p}(t_i, t_j)$$

define a kernel function and

$$\mathcal{H} = \overline{span\{k(t, .)|x \in \mathrm{T}\}}$$

denote the reproducing kernel Hilbert space spanned by k. By definition the

elements of $\mathcal{H}$ can be written in the form

$$v(t) = \int_{\mathrm{T}} k(t,\tau)\alpha(\tau)\mathrm{d}\tau, \quad v \in \mathcal{H}, \tag{4.34}$$

and the inner product within $\mathcal{H}$

$$\left\langle v, v' \right\rangle_{\mathcal{H}} = \int_{\mathrm{T}} \int_{\mathrm{T}} k(\tau_1,\tau_2)\alpha(\tau_1)\alpha'(\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2, \quad v, v' \in \mathcal{H}.$$

In particular

$$||v||^2_{\mathcal{H}} = \int_{\mathrm{T}} \int_{\mathrm{T}} k(\tau_1,\tau_2)\alpha(\tau_1)\alpha(\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2,$$

with $\alpha, \alpha' \in \Lambda \triangleq \mathbf{L}^1(\mathrm{T}, d\tau) \cup \mathcal{M}_D$ as defined in Sec. 4.1.2.

The connection between the RKHS $\mathcal{H}$ and the stochastic process $v^p(t,\varsigma)$

Is the RKHS $\mathcal{H}$ covering all realizations of the stochastic process $v^p(t,\varsigma)$? Two cases have to be distinguished.

Let $\mathcal{A} : \Lambda \to \mathcal{H}$ denote the operator that performs the mapping from Eq. (4.34). If $\mathcal{A}$ has a finite number of nonzero eigenvalues, then the sample functions of $v^p(t,\varsigma)$ are in $\mathcal{H}$ with probability 1. For the case of an infinite number of nonzero eigenvalues however, the sample functions of $v^p(t,\varsigma)$ are not in $\mathcal{H}$ with probability 1 [78]. The basic problem here is that the expected value of the RKHS norm of a sample function of $v^p(t,\varsigma)$ can be shown to be infinite [77]. This is also the case when using the constructed kernel function k from the former section. However, there is a way to construct a RKHS $\mathcal{H}_r$ with kernel function r that is bigger than $\mathcal{H} = \mathcal{H}_k$ that contains the sample paths almost surely.

Theorem 5. *[79]*
Let k and r be two reproducing kernels. Assume that the reproducing kernel Hilbert space $\mathcal{H}_r$ is separable. A necessary and sufficient condition for the existence of a Gaussian process with covariance k and mean $m \in \mathcal{H}_r$ with trajectories in $\mathcal{H}_r$ with probability 1 is that $r \gg k$.

In this context $\gg$ is considered to indicate nuclear dominance. Let's first define the dominance of kernel functions.

Definition 9. *(dominance of kernel functions) [72]*
Given two kernel functions r and k, r dominates k if $\mathcal{H}_k \subseteq \mathcal{H}_r$.

Here, $\mathcal{H}_k$ denotes the RKHS spanned by k and accordingly $\mathcal{H}_r$ the RKHS spanned by r. The connection of both spaces can be expressed by a linear operator.

Theorem 6. *[72]*
Let r dominate k. Then

$$||g||_{\mathcal{H}_r} \leq ||g||_{\mathcal{H}_k}, \quad \forall g \in \mathcal{H}_k,$$

and there exists a unique linear operator $\mathcal{L} : \mathcal{H}_r \to \mathcal{H}_r$ whose range is contained in $\mathcal{H}_k$, such that

$$\langle f, g \rangle_{\mathcal{H}_r} = \langle \mathcal{L}f, g \rangle_{\mathcal{H}_k}, \quad \forall f \in \mathcal{H}_r, \forall g \in \mathcal{H}_k.$$

In particular

$$\mathcal{L}r(., t) = k(., t).$$

As an operator into $\mathcal{H}_r$, $\mathcal{L}$ is bounded, symmetric and positive.

Controversly, let $\mathcal{L} : \mathcal{H}_r \to \mathcal{H}_r$ be a positive, continuous, self-adjoint operator then

$$k(t, t') = \left\langle \mathcal{L}r(., t), r(., t') \right\rangle_{\mathcal{H}_r}$$

defines a reproducing kernel.

$\mathcal{L}$ is called the dominance operator of $\mathcal{H}_r$ over $\mathcal{H}_k$. This dominance is called nuclear if $\mathcal{L}$ is a trace class operator and denoted by $r \gg k$. The trace class operator assures that the norm of the sample function will be finite. It is easy to construct a smoothing operator for which a trace may be defined.

Proposition 1. *[80, p. 114]*
Let g be a continuous function on $[a, b]^2$. Then the integral operator $\mathcal{L}$ on $\mathbf{L}^2([a, b])$ defined by

$$\mathcal{L}f(t) = \int_a^b g(t, \tau) f(\tau) \mathrm{d}\tau \tag{4.35}$$

is trace class, with

$$\mathrm{tr}\mathcal{L} = \int_a^b g(\tau, \tau) \mathrm{d}\tau.$$

Furthermore let

$$g(t_i, t_j) \triangleq \delta_\epsilon(t_i - t_j) = \frac{1}{\sqrt{2\pi\epsilon}} e^{-\frac{(t_i - t_j)^2}{2\epsilon}}$$

define a continuous function that converges to the Dirac delta distribution for $\epsilon \to \infty$. The operator defined by Eq. (4.35) using g is by definition a trace class operator. For a small $\epsilon > 0$, this operator does not change the input by a great margin in the interval $[a + \epsilon, b - \epsilon]$ and it turns out that r is almost identical to k. Therefore, the RKHS spanned by k can be considered to be sufficient to represent all relevant trajectories for most practical applications.

A normed space of trajectories

After the definition of a search space $\mathcal{H}$ for velocities the resulting space of trajectories,

$$\mathcal{X} = \left\{ x | x(t) = x_0 + \int_0^t v(\tau) \mathrm{d}\tau, \quad x_0 \in \mathbb{R}, \quad v \in \mathcal{H} \right\}, \tag{4.36}$$

becomes the range of an integral added to an initial condition. In particular for $v \in \mathcal{H}$

$$x(t) = x_0 + \int_0^t \int_{\mathrm{T}} k(\tau', \tau) \alpha(\tau) \mathrm{d}\tau \mathrm{d}\tau' = x_0 + \int_{\mathrm{T}} (\kappa(\tau) + \kappa(t - \tau)) \alpha(\tau) \mathrm{d}\tau, \tag{4.37}$$

with

$$\kappa(t) \triangleq \int_0^t k(\tau, 0) \mathrm{d}\tau. \tag{4.38}$$

To construct a normed space of trajectories the norm from the RKHS $\mathcal{H}$ may be reused such that

$$||x||_{\mathcal{X}}^2 \triangleq ||v||_{\mathcal{H}}^2. \tag{4.39}$$

Note, because the initial condition does not occur in Eq. (4.39), $(\mathcal{X}, ||.||_{\mathcal{X}})$ can only be shown to be a seminormed space. This is reflected by the fact that the starting position has no influence on the norm.

The norm of $\mathcal{H}$ as a measure for the posterior probability

In Sec. 4.2.2 stochastic processes are defined that embed characteristics motivated by physics proposed in Sec. 4.2.1. In Sec. 4.2.3 a reproducing kernel Hilbert space was proposed that contains all relevant realizations of the stochastic process for the velocity. This subsection introduces the norm of the RKHS as a primer for MAP estimation of the trajectory.

Let $v(t)$ be an element of $\mathcal{H}$ that can be written as a finite sum

$$v(t) = \sum_{i=1}^{n} \alpha_i k(t, t_i), \quad v \in \mathcal{H}.$$

At the sampling times $t_1, ..., t_n$ the evaluation of these elements are written in the vector such that the sampling from Eq. (4.40) can be described by $\boldsymbol{v} \triangleq [v(t_1) \cdots v(t_n)]^{\mathrm{T}}$. If this vector is interpreted as a sampling from a realization of the stochastic process $v^p(t, \varsigma)$ then, from Eq. (4.23), it is known that the probability density function is of the form

$$\mathrm{p}(\boldsymbol{v}) = \frac{1}{(\sqrt{2\pi})^n |\boldsymbol{\Sigma}_v|^{0.5}} e^{-\frac{1}{2}\boldsymbol{v}^{\mathrm{T}} \boldsymbol{\Sigma}_v^{-1} \boldsymbol{v}} \tag{4.40}$$

with $\boldsymbol{\Sigma}_v$ as the covariance matrix for this special sampling. By definition the Gramian matrix $[\boldsymbol{K}]_{ij} = k(t_i, t_j) = \boldsymbol{r}_{v^p}(t_i, t_j)$ is identical to the covariance matrix such that $\boldsymbol{K} = \boldsymbol{\Sigma}_v$. Moreover, the sampling of velocities within $\boldsymbol{v}$ can be described as a matrix vector equation

$$\boldsymbol{v} = \boldsymbol{K}\boldsymbol{\alpha}$$

with the coefficient vector $\boldsymbol{\alpha} \triangleq [\alpha(t_1) \cdots \alpha(t_n)]^{\mathrm{T}}$. As the covariance matrix has full rank by definition, the negative logarithm of the probability density can be written as

$$-\ln(\mathrm{p}(\boldsymbol{v})) = -\ln \frac{1}{(\sqrt{2\pi})^n |\boldsymbol{K}|^{0.5}} + \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} = c_0 + \frac{1}{2}||v||_{\mathcal{H}}$$

with the constant

$$c_0 \triangleq -\ln \frac{1}{(\sqrt{2\pi})^n |\boldsymbol{K}|^{0.5}}.$$

In other words, the norm of an element within the RKHS $\mathcal{H}$ can be interpreted as the negative logarithm of the prior distribution. Does this also hold for the trajectory?

In Sec. 4.23 the covariance function of the process deduced for trajectories was found to be

$$\boldsymbol{r}_{x^p}(t_i, t_j) = R(t_i) + R(t_j) - R(t_j - t_i).$$

Let $\boldsymbol{\Sigma}_x$ denote the covariance matrix with $[\boldsymbol{\Sigma}_x]_{ij} = \boldsymbol{r}_{x^p}(t_i, t_j)$. Then it can be shown that

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{\Sigma}_v^{-1} \boldsymbol{v} = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{\Sigma}_x^{-1} \boldsymbol{x},$$

for any element $v \in \mathcal{H}$ with coefficients $\alpha_1, ..., \alpha_n$ and the corresponding trajectory x obtained by integration. The proof mainly uses two equations

$$x(t) = \int_0^t v(\tau)\mathrm{d}\tau \quad \text{and} \quad \boldsymbol{r}_{x^p}(t_i, t_j) = \int_0^{t_i}\int_0^{t_j} k(\tau_1, \tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2.$$

The full proof can be found in Sec. A.1.3. From this perspective, the seminormed space $\mathcal{X}$ for trajectories becomes a space where the norm is representing a measure for the prior probability. Thus the norm within $\mathcal{X}$ allows to assign probabilities for any sampling vectors within this space. Note, disregarding the initial condition in defining a seminorm for $\mathcal{X}$ is reflecting the fact that the probability for any trajectory is independent from the starting position.

4.3. A solution to the ROLAM problem

Until now, all considerations dealt with a motion in a one-dimensional space. This section establishes a maximum a posteriori estimator for the trajectory based on the framework developed in Sec. 4.1 and Sec. 4.2.

4.3.1. Steering models

In Sec. 4.2.3 the norm of the RKHS spanned by k was introduced as a quantity that indicates prior probabilities for sampling vectors of a one-dimensional mechanical system. An inducing force was assumed to apply to a predefined PSD and the trajectory to a differential equation of motion. In practice however, the true physical layout may not be properly represented by such a model. Imagine a car that is controlled by a steering wheel, an accelerator pedal and brakes. When such a model is known it can be used to construct application specific stochastic processes for the input variables. At this point however, it is worth to recite the no free lunch theorem again. Although such a proceeding promises superior results for the application it comes at the cost of flexibility as it is only applicable for the respective problem. For generality and simplicity, this thesis sticks to a very general and simple steering model.

The extension to three dimensions is performed by treating three forces independently per axis. This corresponds to forces that are directly controlling the vehicle in an inertial frame. Fig. 4.4 depicts such a steering model.

By construction, the three elements of the trajectory can be treated independently. Instead of searching a finite number of positions $\boldsymbol{x}_i$ with $1 \leq i \leq n$ as developed in Sec. 3.1.3 the search space for every dimension of the trajectory is replaced by the normed space $\mathcal{X}$ as defined in Eq. (4.36). Subsequently, the

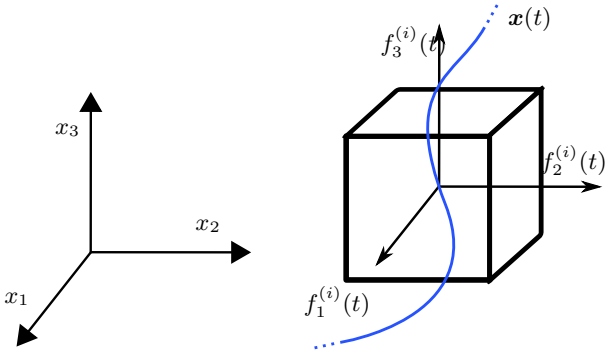

Figure 4.11.: Simple 3-dimensional steering model

three-dimensional trajectories can be represented by

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_0^t \boldsymbol{v}(t)\mathrm{d}t \tag{4.41}$$

with

$$\boldsymbol{x}(t) \triangleq \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \in \mathcal{X}^3, \quad \boldsymbol{x}_0 \in \mathbb{R}^3 \quad \text{and} \quad \boldsymbol{v}(t) \triangleq \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \in \mathcal{H}^3.$$

4.3.2. Maximum a posteriori estimation

Let $\boldsymbol{x}_j = [x_{j,1} \cdots x_{j,n}]^{\mathrm{T}}$ with $j \in \{1, 2, 3\}$ denote sampling vectors that cover the targets positions for arbitrary sampling times $t_1, ..., t_n$, such that $x_{j,i} = x_j(t_i)$. Moreover, let $\boldsymbol{x} \triangleq \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} & \boldsymbol{x}_2^{\mathrm{T}} & \boldsymbol{x}_3^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ denote the vector that contains all positional data and $\boldsymbol{p}^{\langle k \rangle}$ the set of all beacon positions.

The MAP estimator is defined by

$$\left[\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{MAP}} = \arg \max_{\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}} \mathrm{p}\left(\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle} | r^{\langle n \rangle}, a^{\langle n \rangle}\right).$$

By applying Bayes Rule this can be reduced to

$$\left[\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{MAP}} = \arg \max_{\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}} \mathrm{p}\left(r^{\langle n \rangle} | \boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}, a^{\langle n \rangle}\right) \mathrm{p}\left(\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle} | a^{\langle n \rangle}\right).$$

For most applications, the positions of the beacons can be assumed to be independent from the trajectory and the data associations as long as the target is

within the range of the beacons. Therefore, with $\rho > 0$ denoting the range of the beacons,

$$\mathrm{p}\left(\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle} | \boldsymbol{a}^n\right) = \mathrm{p}\left(\boldsymbol{x}\right) \prod_{i=1}^{n} \Lambda(||\boldsymbol{x}(t_i) - \boldsymbol{p}_{a_i}||, \rho)$$

with

$$\Lambda(r, \rho) = \begin{cases} 1, & \text{if } r \leq \rho \\ 0, & \text{if } r > \rho. \end{cases}$$

For simplification the range is assumed to be large enough such that

$$\mathrm{p}\left(\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle} | a^{\langle n \rangle}\right) \propto \mathrm{p}\left(\boldsymbol{x}\right).$$

After taking the negative logarithm and eliminating constant terms, the minimization problem finally writes as

$$\left[\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}\right]_{\mathrm{MAP}} = \arg \min_{\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}} \left[- \ln \left(\mathrm{p}\left(r^{\langle n \rangle} | \boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}, a^{\langle n \rangle}\right)\right) - \ln(\mathrm{p}\left(\boldsymbol{x}\right))\right].$$

The first term covers a maximum likelihood estimator and the second term covers a prior distribution for the trajectories.

The cost function

The cost function is defined in such a way that the minimum of the regularized risk from Eq. (4.12) becomes the maximum a posteriori estimator.

Let $\boldsymbol{x}_{*,i} \triangleq [x_{1,i} \quad x_{2,i} \quad x_{3,i}]^{\mathrm{T}}$ denote the vector of the targets position at time t_i and r_i denote a random variable covering the range measurement at time t_i to the beacon a_i at position $\boldsymbol{p}_{a_i}$. Furthermore let $\mathrm{p}\left(r_i | \boldsymbol{x}_{*,i}, \boldsymbol{p}_{a_i}, a_i\right)$ denote the known probability density function of the independent measurements. With

$$c(t_i, r_i, \bar{r}_i) \triangleq - \ln \left(\mathrm{p}\left(r_i | \boldsymbol{x}_{*,i}, \boldsymbol{p}_{a_i}, a_i\right)\right), \quad \bar{r}_i \triangleq ||\boldsymbol{x}_{*,i} - \boldsymbol{p}_{a_i}||$$

the minimum of the empirical risk function

$$R_{\mathrm{emp}}[\boldsymbol{x}] \triangleq \frac{1}{n} \sum_{i=1}^{n} c(t_i, r_i, \bar{r}_i),$$

as defined in Eq. (4.2), is equal to the minimum of the likelihood function. For the special case of normally distributed measurement errors

$$r_i \sim \mathcal{N}(||\boldsymbol{x}_{*,i} - \boldsymbol{p}_{a_i}||, \sigma_i^2),$$

the empirical risk becomes

$$\begin{aligned} R_{\text{emp}}[\boldsymbol{x}] &\triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}(||\boldsymbol{x}_{*,i}-\boldsymbol{p}_{a_i}||-r_i)^2 = \\ &= -\ln\left(\mathrm{p}\left(r^{\langle n\rangle}|\boldsymbol{x},\boldsymbol{p}^{\langle k\rangle},a^{\langle n\rangle}\right)\right)+c_0 \end{aligned} \tag{4.42}$$

with a constant $c_0 \in \mathbb{R}$. Note, different error distributions for the measurement error may be covered by different cost functions.

The prior

With the interpretation of $\boldsymbol{x}_j$, $j \in \{1,2,3\}$ as a realization of a stochastic process and the construction of the steering model from the former section the probability density function can be splitted up such that

$$\mathrm{p}(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3) = \mathrm{p}(\boldsymbol{x}_1)\,\mathrm{p}(\boldsymbol{x}_2)\,\mathrm{p}(\boldsymbol{x}_3)\,. \tag{4.43}$$

Note, the processes representing the positions are stochastic processes with unknown initial position $\boldsymbol{x}_0 = [x_{j,1} \quad x_{2,1} \quad x_{3,1}]^{\mathrm{T}}$. Instead, consider $\boldsymbol{x}_j^p = [0 \quad x_{j,2}-x_{j,1} \quad \cdots \quad x_{j,n}-x_{j,1}]^{\mathrm{T}}$ which can be interpreted as a realization of the process representing the trajectories as defined in Sec. 4.2.2. If the initial position $\boldsymbol{x}_0$ can be assumed to be equally distributed, then Eq. (4.43) becomes

$$\mathrm{p}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3) = \mathrm{p}(\boldsymbol{x}_0)\,\mathrm{p}(\boldsymbol{x}_1^p)\,\mathrm{p}(\boldsymbol{x}_2^p)\,\mathrm{p}(\boldsymbol{x}_3^p) \propto \mathrm{p}(\boldsymbol{x}_1^p)\,\mathrm{p}(\boldsymbol{x}_2^p)\,\mathrm{p}(\boldsymbol{x}_3^p)\,. \tag{4.44}$$

In Sec. 4.2.3 the trajectories with the property $x_j(0)=0$ were embedded into the seminormed space $(\mathcal{X},||.||_{\mathcal{X}})$ such that the negative logarithm of Eq. (4.44) can be expressed using the norm within $\mathcal{X}$. Choose $x_1(.),x_2(.),x_3(.) \in \mathcal{X}$ such that $x_j(t_i) = x_{j,i}^p$, then

$$-\ln(\mathrm{p}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)) = ||x_1(.)||_{\mathcal{X}}^2 + ||x_2(.)||_{\mathcal{X}}^2 + ||x_3(.)||_{\mathcal{X}}^2 + c_1 \tag{4.45}$$

with c_1 as a constant. Note, the independent treatment of the axes becomes apparent by summing up the norms.

A set of assorted two-dimensional trajectories $\boldsymbol{x}_1 \triangleq [x_{1,1} \quad x_{2,1}]^{\mathrm{T}}, \boldsymbol{x}_2 \triangleq [x_{1,2} \quad x_{2,2}]^{\mathrm{T}}, \boldsymbol{x}_3 \triangleq [x_{1,3} \quad x_{2,3}]^{\mathrm{T}} \in \mathcal{X}\times\mathcal{X}$ starting at $\boldsymbol{x}_0 \triangleq [0 \quad 0]^{\mathrm{T}}$ are plotted in Fig. 4.12. The parameters of the kernel function were chosen to be $a=10.0$, $b=1$, $\beta=5$ and $m=5$. The sum of their norms $||x_1||_{\mathcal{X}}+||y_1||_{\mathcal{X}}$, $||x_2||_{\mathcal{X}}+||y_2||_{\mathcal{X}}$ and $||x_3||_{\mathcal{X}}+||y_3||_{\mathcal{X}}$ are listed within the plot. As a consequence, $\mathrm{p}(\boldsymbol{x}_1) > \mathrm{p}(\boldsymbol{x}_2) > \mathrm{p}(\boldsymbol{x}_3)$.

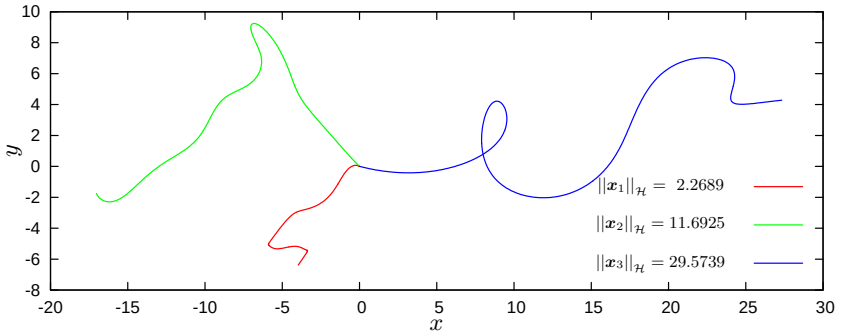

Figure 4.12.: Three trajectories and their norms

The regularized risk functional

A combination of the prior from Eq. (4.45) and the empirical risk from Eq. (4.42) delivers the regularized risk functional

$$R_{\text{reg}}[\boldsymbol{x}, \boldsymbol{p}^k] \triangleq R_{\text{emp}}[\boldsymbol{x}] + \mu_1 ||x_1(.)||^2_{\mathcal{X}} + \mu_2 ||x_2(.)||^2_{\mathcal{X}} + \mu_3 ||x_3(.)||^2_{\mathcal{X}}. \tag{4.46}$$

Note, the regularizing term is representing a norm. Eq. (4.46) is therefore of the form from Eq. (4.13) and thus covered by Theorem 3 introduced in Sec. 4.1.3.

Let $\boldsymbol{\alpha}_j = [\alpha_{j,1} \cdots \alpha_{j,n}]^{\mathrm{T}}$, $1 \leq j \leq 3$ denote the vector of coefficients for the sampling times $t_1, ..., t_n$ of the measurements. Then the final representation of the regularized risk functional becomes

$$R_{\text{reg}}[\boldsymbol{x}, \boldsymbol{p}^k] \triangleq R_{\text{emp}}[\boldsymbol{x}] + \mu_1 \boldsymbol{\alpha}_1^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}_1 + \mu_2 \boldsymbol{\alpha}_2^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}_2 + \mu_3 \boldsymbol{\alpha}_3^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}_3 \tag{4.47}$$

with the additional regularization parameters $\mu_1, \mu_2, \mu_3 \in \mathbb{R}$. The final optimization problem becomes

$$\left[\boldsymbol{x}, \boldsymbol{p}^k\right]_{\text{MAP}} = \arg\min_{\boldsymbol{x}, \boldsymbol{p}^k} R_{\text{reg}}[\boldsymbol{x}, \boldsymbol{p}^k]. \tag{4.48}$$

These regularization parameters can be interpreted as follows. Assumed, the vectors $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_3$ minimize the regularized risk as defined in Eq. (4.47) with Gram matrix $\boldsymbol{K}$ and regularization parameters μ_1, μ_2 and μ_3. Then the coefficient vectors $\hat{\boldsymbol{\alpha}}_1 = \frac{1}{2}\boldsymbol{\alpha}_1$, $\hat{\boldsymbol{\alpha}}_2 = \frac{1}{2}\boldsymbol{\alpha}_2$, $\hat{\boldsymbol{\alpha}}_3 = \frac{1}{2}\boldsymbol{\alpha}_3$ are minimizing the regularized risk functional for a different kernel function with $\hat{\boldsymbol{K}} = 2\boldsymbol{K}$ and regularization parameters $\hat{\mu}_1 = 2\mu_1, \hat{\mu}_2 = 2\mu_2$ and $\hat{\mu}_3 = 2\mu_3$. Thus, scaling the regularization parameters μ_1, μ_2 and μ_3 has the same scope as scaling the kernel functions by the design parameter a. Moreover a was already shown to have the same scope as the model noise ν from Eq. (4.26). Therefore, the regularizing parameters μ_1,

μ_2 and μ_3 can also be interpreted as a scaling parameters indirectly proportional to the amplitude of the input model noise. By applying different kernel functions for each dimension these parameters could be replaced by different kernel parameters a_1, a_2 and a_3. However, for simplicity it can be beneficial to use a as a single parameter scaling the kernel function and keep the regularization parameters.

Unfortunately, Eq. (4.48) describes a non convex optimization problem as illustrated by Fig. 4.13.

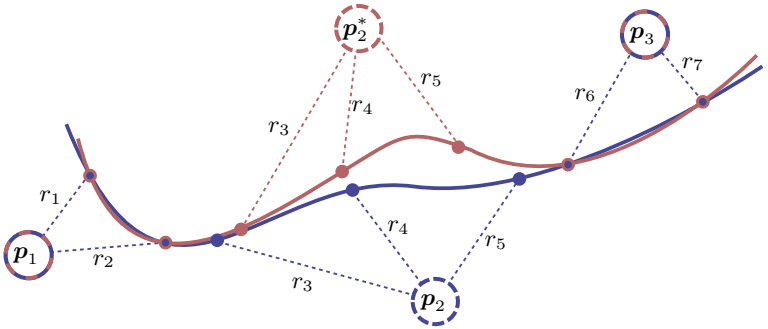

Figure 4.13.: Global and local minimum for the same set of range data

Assumed the blue curve from Fig. 4.13 represents the global minimum of the regularized risk from Eq. (4.48) and the red curve an initial value for a numerical optimization. As indicated by the sketch, both curves induce exactly the same value for $R_{\text{emp}}[\boldsymbol{x}]$ and by their similarity almost the same values for the regularizing terms. However, the initial value for $\boldsymbol{p}_2^*$ greatly differs from the optimal position $\boldsymbol{p}_2$. By only descending the objective function the optimal parameter set can not be reached using these initial values. The regularized risk functional has a local minimum. Although, this phenomenon can also be observed for the three-dimensional case, the tests performed in the next chapter show a good convergence behaviour for the investigated scenarios with randomly chosen initial values. Problems concerning this issue are not investigated further within this thesis.

4.4. Consistency

So far, a reasonable minimization problem was derived whose solution is a finite set of parameters $\boldsymbol{\alpha}_i$ which represent the solution. Close inspection of Eq. (4.48) reveals however that the number of unknowns grows with the number of obser-

vations. Although the new problem formulation with an additional regularizing term adds information, the number of unknown positions $\boldsymbol{x}$ still equals the number of the unknown coefficients. The problem can therefore still be considered to be ill-posed. In Sec. 4.1.1 one sided uniform convergence in probability in Theorem 1 was introduced as the crucial condition such that a regression method can be successful. For the ROLAM problem this requires additional conditions that are addressed in this section. Besides consistency, the results also allows to quantify when minimization of Eq. (4.48) can be successful.

4.4.1. Definition of the search space $\mathcal{V}$

Consider empirical risk minimization where the solution $\check{f}$ is an element of a finite solution space

$$\check{f} = \arg\min_{f} R_{\text{emp}}[f], \quad f \in \mathcal{F} \triangleq \{f_1, ..., f_m\}.$$

Consistency in the sense of Theorem 1 can then be shown by applying the *Chernoff bound* in combination with the *Union Bound* which is an application of the law of large numbers [67, p. 130,p. 135].

A finite representation of the solution space

The general idea is therefore to represent the solution space by a finite number of samples such that the error induced by the sampling remains small.

Definition 10. *(ϵ-cover)*
Let M and E denote sets such that $M \subset E$. Then an ϵ-cover *is defined as a set of points in E such that the union of all ϵ-balls around these points contains M.*

Let $\mathcal{H}$ denote the RKHS spanned by the kernel function k and d denote the metric induced by the values of $v_1, v_2 \in \mathcal{H}$ on some data $T = [t_1, t_n]$ such that

$$d(v_1, v_2) \triangleq \max_{t \in T} |v_1(t) - v_2(t)|.$$

Assumed, an ϵ-cover for $\mathcal{H}$ with m samples exists then $\mathcal{H}^3$ and $\mathcal{X}^3$ can be covered by m^3 samples disregarding the initial position of the target. Note, the metric d is defined in $\mathcal{H}$. This implies the assumption that with an ϵ-cover for the velocities the space of trajectories can be sufficiently represented. However, this restriction is not investigated further.

The remainder of this section is dedicated to find an appropriate search space $\mathcal{V}$ and an ϵ-cover for $\mathcal{V}$. The number of samples required to cover $\mathcal{V}$ indicates the capacity of the constructed solution space. In Fig. 4.14 an *ϵ-cover* is sketched for the search space $\mathcal{V} \subset \mathcal{H}$.

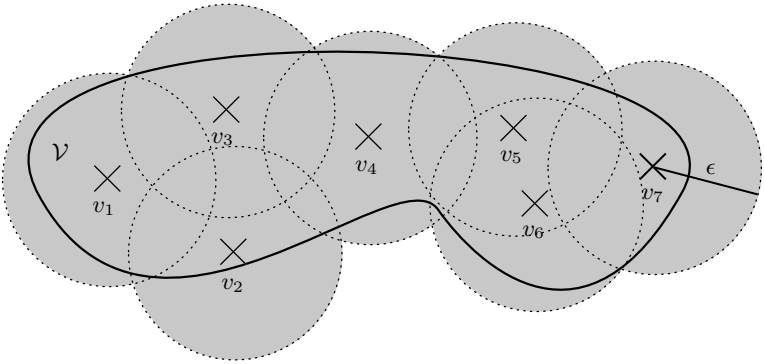

Figure 4.14.: ϵ-cover for the function space $\mathcal{V}$

Restriction on probable velocities

Let $T \triangleq [t_1, t_n]$ denote the region of interest and $v \in \mathcal{H}$. Then, following Sec. 4.3.2, the norm $||v||_{\mathcal{H}}$ is intended to restrict the search space in such a way that a unique solution can be calculated. To this end, the space is reduced to a subspace of probable velocity functions.

Let

$$\mathcal{H}_\Lambda \triangleq \{v \in \mathcal{H}, ||v||_{\mathcal{H}} \leq \Lambda\} \subseteq \mathcal{H} \tag{4.49}$$

denote a ball of radius Λ in $\mathcal{H}$. By the interpretation of any element of $\mathcal{H}$ as a realization of the stochastic process $v(t, \varsigma)$ the ball separates likely velocity functions within the ball from unlikely velocity functions outside of the ball. Assumed, the stochastic model that was developed for the velocities is appropriate, then the radius Λ can be adjusted in such a way that only very unlikely velocity functions are not covered by $\mathcal{H}_\Lambda$. For this reason let $\mathcal{V} = \mathcal{H}_\Lambda$ denote the search space for the velocities for an appropriate $\Lambda \in \mathbb{R}$.

The basic structure of $\mathcal{V}$ can be investigated by applying Eq. (4.11) on the norm

$$||v||_{\mathcal{H}}^2 = \int_{-\infty}^{\infty} \frac{V(\omega)\overline{V(\omega)}}{K(\omega)} \mathrm{d}\omega \tag{4.50}$$

with $V(\omega) \triangleq \mathfrak{F}\{v\}(t)$ and $K(\omega) \triangleq \mathfrak{F}\{k(0,t)\}(\omega)$. If $v(t) \in \mathcal{V}$ contains frequencies with a high amplitude that are sparsely covered in the kernel function, the norm of the element has a great value. Thus, the elements within the ball $\mathcal{V}$ either are smooth with respect to the kernel function that spans $\mathcal{H}$ or they have a very small amplitude. For the designed kernel in Sec. 4.2

with $a = 1$, $b = 2$, $m = 10$ and $\beta = 6$, three elements $v_1, v_2, v_3 \in \mathcal{V}$ were plotted in Fig. 4.15. All elements reside on the surface of the ball with radius $||v_1||_{\mathcal{H}} = ||v_2||_{\mathcal{H}} = ||v_3||_{\mathcal{H}} = \Lambda \triangleq 1$.

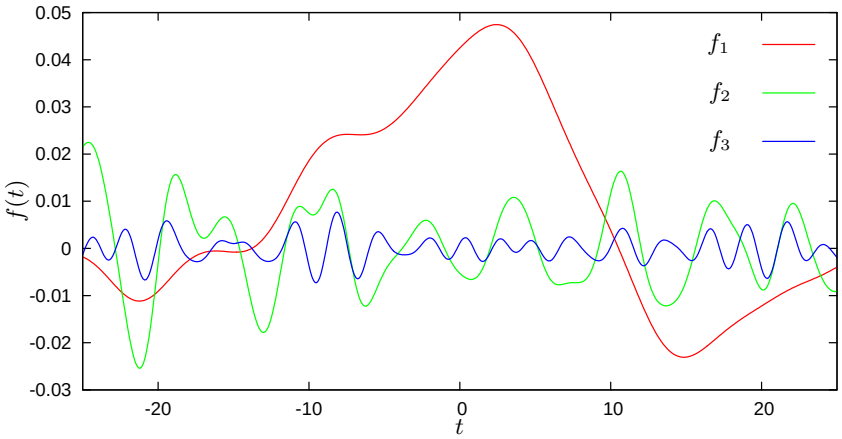

Figure 4.15.: Three elements of $\mathcal{V}$ with $||v_i||_{\mathcal{H}} = 1$

As mentioned before, the higher the frequency of the signal, the smaller the amplitude. $\mathcal{V}$ can therefore be considered to be limited in bandwith. If one intends to find an ϵ-cover for $\mathcal{V}$ in form of a number of samples, the *Nyquist-Shannon sampling theorem* states that the number is dependent on the bandwith of the signal [81, 82]. The more restrictive the kernel in frequency domain, the smaller the number of required samples to cover $\mathcal{V}$.

An ϵ-cover for $\mathcal{V}$

The following proposition shows, that there exists an ϵ-cover for $\mathcal{V}$. Restricting the search space to $\mathcal{V}$ allows to reduce the number of parameters from n to $m < n$ such that the ROLAM problem becomes feasible without introducing an error larger than ε for almost all time instants within the interval of interest.

Proposition 2. *Let $t_1, ..., t_n \in [0, T]$ be an arbitrary sampling and $v = \sum_{i=1}^{n} \alpha_i k(t, t_i) \in \mathcal{H}_\Lambda$. For any $\varepsilon > 0$ there exists a finite m, a set of orthonormal basis functions $\tilde{\psi}_1, ..., \tilde{\psi}_m \in \mathcal{H}$, $\tilde{\lambda}_1, ..., \tilde{\lambda}_m \in \mathbb{R}_+$, and parameters $\nu_1, ..., \nu_m$, such that*

$$\left| v(t) - \sum_{j=1}^{m} \nu_j \sqrt{\tilde{\lambda}_j} \tilde{\psi}_j(t) \right| \leq \varepsilon \tag{4.51}$$

for almost all $t \in [0, T]$.

Proof. By definition $\mathcal{H}$ is spanned by a symmetric real valued Mercer kernel. Following *Mercer's theorem* [67, p. 37], the eigenvalues $\tilde{\lambda}_i$ to the corresponding orthonormal eigenfunction $\tilde{\psi}_i$ of the operator

$$(\mathcal{T}\alpha)(t) \triangleq \int_0^T k(t, t')\alpha(\tau)dt' \tag{4.52}$$

satisfy $\sum_{i=1}^{\infty} |\tilde{\lambda}_i| < \infty$. Assuming the kernel function $k(t, t')$ is continuous, then $\mathcal{T}$ is additionally self adjoined, compact and a trace class operator [83]. Thus, for any $\tilde{\varepsilon}$ there exists a finite m (cf.[83] and [67, p. 38]), such that

$$\left| k(t, t') - \sum_{j=1}^{m} \tilde{\lambda}_j \tilde{\psi}_j(t)\tilde{\psi}_j(t') \right| < \tilde{\varepsilon} \tag{4.53}$$

for almost all $t, t' \in [0, T]$. Using the representation for the kernel function $k(t, t')$ from Eq. (4.53) and re-sorting yields

$$v(t) = \lim_{m\to\infty} \sum_{j=1}^{m} \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t) \underbrace{\sum_{i=1}^{n} \alpha_i \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t_i)}_{\nu_j} \tag{4.54}$$

$$= \sum_{j=1}^{\infty} \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t)\nu_j. \tag{4.55}$$

We insert this result into the LHS of Eq. (4.51) and apply Hölders inequality to obtain

$$\left| \sum_{j=m+1}^{\infty} \nu_j \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t) \right| \leq ||\boldsymbol{\nu}||_2 ||\boldsymbol{\phi}(t)||_2,$$

with $\boldsymbol{\nu} \triangleq \begin{bmatrix} \nu_1 & \nu_2 \cdots \end{bmatrix} \in \ell^2$ and

$$\boldsymbol{\phi}(t) \triangleq \begin{bmatrix} \sqrt{\tilde{\lambda}_{m+1}}\tilde{\psi}_{m+1}(t) & \sqrt{\tilde{\lambda}_{m+2}}\tilde{\psi}_{m+2}(t) \cdots \end{bmatrix} \in \ell^2.$$

Due to the definition of ν_j in Eq. (4.55), we obtain

$$\begin{aligned} ||\boldsymbol{\nu}||_2^2 &= \textstyle\sum_{j=1}^{\infty} \sum_{i=1}^{n} \alpha_i \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t_i) \sum_{i'=1}^{n} \alpha_{i'} \sqrt{\tilde{\lambda}_j}\tilde{\psi}_j(t_{i'}) \\ &= \textstyle\sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} \sum_{j=1}^{\infty} \tilde{\lambda}_j \tilde{\psi}_j(t_i)\tilde{\psi}_j(t_{i'}), \end{aligned}$$

and by applying Mercer's theorem and the assumption $v \in \mathcal{H}_\Lambda$,

$$\begin{aligned} ||\boldsymbol{\nu}||_2^2 &= \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} k(t_i, t_{i'}) \\ &= \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha} = ||v||_{\mathcal{H}}^2 \leq \Lambda^2. \end{aligned}$$

Eventually, setting $\tilde{\varepsilon} = \varepsilon \Lambda^{-1}$ and identifying $||\boldsymbol{\phi}(t)||_2$ with the LHS of Eq. (4.53) proves the proposition. □

This basically means that independent of any sampling and in particular any number of samplings n, a set of m parameters is enough to represent every element v within the ball $\mathcal{H}_\Lambda$, with respect to the given accuracy ε. Whenever $m < n$, the coefficients for the orthonormal basis can be uniquely determined. Alternatively, a choice of $m < n$ defines the level of accuracy ε that can be reached with n measurements. From this perspective the impact of Eq. (4.50) becomes apparent. The eigenvalues of the operator $\mathcal{T}$ from Eq. (4.52) are penalizing factors for the respective frequencies in $v(t) \in \mathcal{V}$. The decay in the eigenvalues therefore describes the filter properties of the kernel function. The steeper the eigenvalue decay, the more limited the bandwidth of the operator $\mathcal{T}$ and the less samples and respectively parameters are needed to reconstruct elements created by $\mathcal{T}$.

4.4.2. Eigenvalues and eigenvectors of the kernel map

The result from the former section was that an ϵ-cover can be found when regarding the kernel map in frequency domain. Elements with high frequencies are supposed to have a low impact on the velocity function. With respect to an ϵ-cover, these elements can be neglected such that $\mathcal{V}$ can be represented by a finite number of samples. Unfortunately, the eigenvalues and eigenfunctions of $\mathcal{T}$ are not easy to obtain. However, the eigenvectors and eigenvalues of the Gramian matrix $\boldsymbol{K}$ provide a good approximation.

Proposition 3. *Let $k(t, t')$ be a Mercer kernel and Eq. (4.52) define the corresponding integral operator with eigenvalues $\tilde{\lambda}_j$ and eigenfunctions $\tilde{\psi}_j$. Furthermore, let $t_1, ..., t_n$ be an i.i.d. sampling within $[0, T]$ and $\boldsymbol{K}$ the corresponding Gramian matrix created by $k(t, t')$.*
Then, for $n \to \infty$ the eigenvectors $\boldsymbol{\psi}_j$ of $\boldsymbol{K}$ converge to scaled discrete samplings of the eigenfunctions $\tilde{\psi}_j$, such that $\boldsymbol{\psi}_j \approx \frac{\sqrt{T}}{\sqrt{n}} \left[\tilde{\psi}_j(t_1) \cdots \tilde{\psi}_j(t_n)\right]^{\mathrm{T}}$ and $\lambda_j \approx \frac{n}{T} \tilde{\lambda}_j$. In particular, $\sqrt{\lambda_j} \left[\boldsymbol{\psi}_j\right]_i \approx \tilde{\lambda}_j \tilde{\psi}_j(t_i)$.

Proof. As $k(t,t')$ is quadratically integrable, *Mercer's theorem* [67, p. 37] states that there exists orthonormal eigenfunctions $\tilde{\psi}_j(t) \in \mathbf{L}^2(\mathbb{R})$ associated with eigenvalues $\tilde{\lambda}_1 > \tilde{\lambda}_2 > \cdots$ and $\lim_{j\to\infty} \tilde{\lambda}_j = 0$ such that $k(t,\tau) = \sum_{j=1}^{\infty} \tilde{\lambda}_j \psi_j(t)\psi_j(\tau)$, cf. Eq. (4.53).

If $k(t,\tau)\alpha(\tau)$ is *Riemann-integrable* in τ then for i.i.d. sampled $t_1, ..., t_n$ within $[0,T]$ the *Monte Carlo* (MC) estimate for the integral is known [39] to converge such that

$$(\mathcal{T}\alpha)(t) \triangleq \int_0^T k(t,t')\alpha(t')dt' \approx \frac{T}{n}\sum_{i=1}^{n} k(t,t_i)\alpha(t_i).$$

For sufficiently large n, a discrete sampling of any eigenfunction $\tilde{\psi}_j(t)$ with eigenvalue $\tilde{\lambda}_j$ of $\mathcal{T}$ and $(\mathcal{T}\tilde{\psi}_j)(t) = \tilde{\lambda}_j\tilde{\psi}_j(t)$ yields

$$\frac{T}{n}\sum_{i=1}^{n} k(t,t_i)\tilde{\psi}_j(t_i) \approx \tilde{\lambda}_j\tilde{\psi}_j(t). \tag{4.56}$$

By the discrete sampling $t_1, ..., t_n$ and introducing the normalization factor γ_n on both sides of Eq. (4.56), we obtain

$$\boldsymbol{K}\boldsymbol{\psi}_j = \lambda_j\boldsymbol{\psi}_j,$$

with $\lambda_j = \frac{n}{T}\tilde{\lambda}_j$, $\boldsymbol{\psi}_j = \gamma_n \left[\tilde{\psi}_j(t_1) \dots \tilde{\psi}_j(t_n)\right]^{\mathrm{T}}$ and γ_n such that

$$||\boldsymbol{\psi}_j||_2^2 = \gamma_n^2 \sum_{i=1}^{n} \tilde{\psi}_j(t_i)^2 \approx \frac{n\gamma_n^2}{T}||\tilde{\psi}_j(\tau)||_{\mathbf{L}^2}^2 = \frac{n\gamma_n^2}{T} = 1,$$

again exploiting the Monte Carlo technique. Resolving for γ_n yields $\gamma_n \approx \frac{\sqrt{T}}{\sqrt{n}}$. This concludes the proof. □

The eigenvalues $\tilde{\lambda}_j$ of the kernel map were plotted in Fig. 4.16 for the parameters $a = 1$, $b = 2$, $m = 10$ and $\beta = 6$. Variations in the parameters are denoted within the figure.

The steeper the descent of the eigenvalue decay the less elements are required for the epsilon cover. However note, the decay is dependent on the kernel parameters and the size of the region of interest T. For a fixed ϵ an expansion of T also requires to increase the amount of measurements which is rather clear from the application.

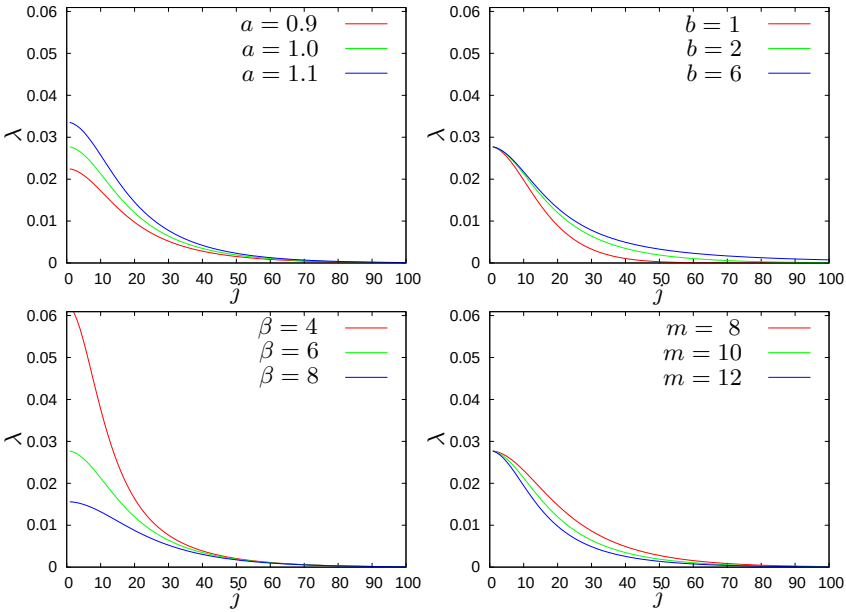

Figure 4.16.: Eigenvalue decay of $\mathcal{T}$ for different parameters

5. Minimization methods for solving ROLAM

After formulating an applicable minimization problem this chapter introduces methods for solving ROLAM. The proposed regularized risk functional from Eq. (4.47) is minimized in two steps.

An initial step is addressed in Sec. 5.1. Data, that was gathered over the first time period, is used to calculate a solution with the Levenberg-Marquardt algorithm. The optimization process is executed offline.

Sec. 5.2 introduces a sequential optimization process that incorporate new data into the solution in real time. The online estimation algorithm uses the Bayesian update step from Eq. (3.5). To this end, an EKF is proposed that is initialized by the offline estimate from Sec. 5.1.

5.1. Offline estimation

By the stochastic input in form of range measurements r_i two quantities are of interest for obtaining a solution. The minimum of the regularized risk functional from Eq. (4.48) and the significance of the result in form of a covariance matrix. This is backed by the *central limit theorem* which states that the probability density function of the mean of a random sampling is normally distributed [18, p. 714]. The resulting parameters $\boldsymbol{x}_0 \in \mathbb{R}^3$, $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3 \in \mathbb{R}^n$, $\boldsymbol{p}_1, ..., \boldsymbol{p}_k \in \mathbb{R}^3$ are therefore assumed to be jointly normally distributed.

5.1.1. Levenberg-Marquardt minimization

An adequate approach for minimizing Eq. (4.47) is the *Levenberg-Marquardt algorithm* [84, 85, p. 683, p. 47] also known as the damped least-squares method. This method combines the fast convergence of the *Gauss Newton* method and the stability of the *steepest descent* by balancing both methods by a damping parameter that is also called *Levenberg-Marquardt parameter*.

Let

$$\boldsymbol{F} : \mathbb{R}^M \to \mathbb{R}^N, \quad M < N, \quad \boldsymbol{\theta} \mapsto (F_1(\boldsymbol{\theta}) \cdots F_N(\boldsymbol{\theta}))^{\mathrm{T}}$$

be a multidimensional non linear function. Then the *Levenberg-Marquardt* method minimizes

$$\sum_{i=1}^{N} F_i^2(\boldsymbol{\theta}).$$

In order to comply with this method, the problem needs to be reformulated. An appropriate parameterization can be derived by the eigenvalue decomposition of the Gramian matrix $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}$. A change of the base vectors provides a representation of the trajectory in terms of new coefficients. For scaling reasons, the eigenvalues are incorporated within the new coefficient vector such that $\hat{\boldsymbol{\nu}}_j = [\hat{\nu}_{j,1} \cdots \hat{\nu}_{j,n}]^{\mathrm{T}} \triangleq \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\alpha}_j,\ j \in \{1,2,3\}$. Further reduction to a set of $m < n$ dominating coefficients with respect to the ϵ-cover from Sec. 4.4.1 delivers the final dataset. The reduced parameter vector becomes

$$\boldsymbol{\theta} \triangleq \begin{bmatrix} \boldsymbol{x}_0^{\mathrm{T}} & \boldsymbol{p}_1^{\mathrm{T}} & \cdots & \boldsymbol{p}_k^{\mathrm{T}} & \boldsymbol{\nu}_1^{\mathrm{T}} & \boldsymbol{\nu}_2^{\mathrm{T}} & \boldsymbol{\nu}_3^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

with $\boldsymbol{\nu}_j \triangleq [\hat{\nu}_{j,1} \cdots \hat{\nu}_{j,m}]^{\mathrm{T}}$, $\mathrm{Dim}(\boldsymbol{\theta}) = 3(1+k+m)$ and

$$F(\boldsymbol{\theta}) \triangleq \begin{bmatrix} F_1(\boldsymbol{\theta}) \\ \vdots \\ F_n(\boldsymbol{\theta}) \\ F_{n+1}(\boldsymbol{\theta}) \\ \vdots \\ F_{n+3m}(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \sigma_1^{-1}(||\boldsymbol{x}_{*,1} - \boldsymbol{p}_{a_1}|| - r_1) \\ \vdots \\ \sigma_n^{-1}(||\boldsymbol{x}_{*,n} - \boldsymbol{p}_{a_n}|| - r_n) \\ \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \boldsymbol{\nu}_3 \end{bmatrix}.$$

Note, after the base change the regularizing terms simplify to $\boldsymbol{\alpha}_j\boldsymbol{K}\boldsymbol{\alpha}_j = \boldsymbol{\nu}_j^{\mathrm{T}}\boldsymbol{\nu}_j$. Thus,

$$R_{\mathrm{reg}}[\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}] = \sum_{i=1}^{N} F_i^2(\boldsymbol{\theta}) \tag{5.1}$$

and the Levenberg-Marquardt algorithm can be applied to the ROLAM problem. Note, the Jacobian matrix that is necessary for this method can be obtained by differentiating Eq. (5.1) analytically or numerically.

5.1.2. Covariances

As the input range data is corrupted by noise, the minimum of Eq. (5.1) only provides an estimate for the parameter vector $\boldsymbol{\theta}$. The confidence in this estimate can be expressed in form of a covariance matrix for $\boldsymbol{\theta}$. Such an estimate can be obtained by *Laplace's method* [86].

Laplace's method

Let $\mathrm{p}(\boldsymbol{\theta})$ be a probability density function for the random vector $\boldsymbol{\theta}$ that can be written as

$$\mathrm{p}(\boldsymbol{\theta}) \triangleq ce^{-S(\boldsymbol{\theta})},$$

with $S(\boldsymbol{\theta})$ as a function in the parameter vector $\boldsymbol{\theta}$ and a constant $c \in \mathbb{R}$. Then the *Taylor series expansion* of $S(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_0$ yields

$$S(\boldsymbol{\theta}) = S(\boldsymbol{\theta}_0) + \nabla S(\boldsymbol{\theta}_0)^{\mathrm{T}}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^{\mathrm{T}} \boldsymbol{H}_{\boldsymbol{\theta}_0}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + O(\boldsymbol{\theta}^3),$$

with $\boldsymbol{H}_{\boldsymbol{\theta}_0}$ as the *Hessian* matrix of $S(\boldsymbol{\theta})$ evaluated at $\boldsymbol{\theta}_0$. Assumed, $\boldsymbol{\theta}_0$ is a maximum of $\mathrm{p}(\boldsymbol{\theta})$ and therefore a minimum of $S(\boldsymbol{\theta})$, respectively. Then $\nabla S(\boldsymbol{\theta}_0) = \mathbf{0}$ and thus

$$S(\boldsymbol{\theta}) \approx S(\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^{\mathrm{T}} \boldsymbol{H}_{\boldsymbol{\theta}_0}(\boldsymbol{\theta} - \boldsymbol{\theta}_0).$$

Consequently,

$$\mathrm{p}(\boldsymbol{\theta}) \approx c^* e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^{\mathrm{T}} \boldsymbol{H}_{\boldsymbol{\theta}_0}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)} \quad \text{with} \quad c^* \triangleq c e^{-S(\boldsymbol{\theta}_0)}.$$

This is an approximation for $\mathrm{p}(\boldsymbol{\theta})$ in form of a normal distribution. The estimate for the covariance matrix has the form of the Hessian matrix of $S(\boldsymbol{\theta})$

$$\Sigma_{\boldsymbol{\theta}}^{-1} \approx \boldsymbol{H}_{\boldsymbol{\theta}_0}.$$

By construction of Eq. (4.47), the probability density for the parameter set can be written as

$$\mathrm{p}(\boldsymbol{\theta}) = e^{\frac{1}{2} R_{\mathrm{reg}}[\boldsymbol{x}, \boldsymbol{p}^{\langle k \rangle}]}.$$

A definition of

$$2S(\boldsymbol{\theta}) \triangleq \Delta \boldsymbol{r}^{\mathrm{T}} \boldsymbol{R}^{-1} \Delta \boldsymbol{r} + \boldsymbol{\nu}^{\mathrm{T}} \boldsymbol{\nu} \tag{5.2}$$

with $\Delta \boldsymbol{r} \triangleq [r_1 - \bar{r}_1 \cdots r_n - \bar{r}_n]^{\mathrm{T}}$, $\bar{r}_i \triangleq ||\boldsymbol{x}_{*,i} - \boldsymbol{p}_{a_i}||$ and the diagonal matrix $\boldsymbol{R}$, as well as $[\boldsymbol{R}]_{ii} = \sigma_i^2$ therefore allows a direct appliance of Laplace's method.

5.2. Online estimation

In Sec. 5.1 the ROLAM problem is solved for a fixed time interval $[t_1, t_n]$. However, for most real world applications the solution is needed in real time. This section presents a solver for Eq. (4.48) that incorporates new range measurements in real time. With some additional assumptions the Bayesian update equation from Eq. (3.5) can be applied to the problem. The primary goal for this section is therefore to incorporate the approach of kernel regression into a state space description of the system.

5.2.1. Velocity estimation

In the former chapters kernel regression was used to interpolate between the time instances $t_1, ..., t_n$. However, the method can also be used to extrapolate into the future.

Bayesian inference

Let $\boldsymbol{v}_n \triangleq [v_1 \cdots v_n]^{\mathrm{T}}$ denote a vector sampled from the stochastic process $v_p(t, \varsigma)$ at time instances $\boldsymbol{t}_n \triangleq [t_1 \cdots t_n]^{\mathrm{T}}$. By definition, the vector $\boldsymbol{v}_n$ is jointly normally distributed with

$$\mathrm{E}[\boldsymbol{v}_n] = \boldsymbol{0}_n$$

and the covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_n} = \boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n), \quad [\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)]_{ij} \triangleq k(t_i, t_j), \quad t_i, t_j \in \{t_1, ...t_n\}.$$

An expanded vector $\boldsymbol{v}_{n+1} \triangleq \begin{bmatrix} \boldsymbol{v}_n^{\mathrm{T}} & v_{n+1} \end{bmatrix}^{\mathrm{T}}$, also sampled from the same stochastic process at time instances $\begin{bmatrix} \boldsymbol{t}_n^{\mathrm{T}} & t_{n+1} \end{bmatrix}^{\mathrm{T}}$, with v_{n+1} denoting the velocity at time t_{n+1} can be treated the same way. The expectation of the expanded vector is determined by

$$\mathrm{E}[\boldsymbol{v}_{n+1}] = \boldsymbol{0}_{n+1},$$

and the covariance matrix is known to be

$$\boldsymbol{\Sigma}_{\boldsymbol{v}_{n+1}} = \begin{bmatrix} \boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n) & \boldsymbol{K}(\boldsymbol{t}_n, t_{n+1}) \\ \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n) & \boldsymbol{K}(t_{n+1}, t_{n+1}) \end{bmatrix}.$$

If $\boldsymbol{v}_n$ is known and v_{n+1} is unknown, then the probability density function for v_{n+1} can be expressed by

$$\mathrm{p}\,(v_{n+1}|\boldsymbol{v}_n) = \frac{\mathrm{p}\,(\boldsymbol{v}_n, v_{n+1})}{\mathrm{p}\,(\boldsymbol{v}_n)}. \tag{5.3}$$

Since $\boldsymbol{v}_n$ and v_{n+1} are normally distributed random variables, Eq. (5.3) can be explicitly evaluated. The expectation becomes

$$\mathrm{E}[v_{n+1}|\boldsymbol{v}_n] = \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n)\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}\boldsymbol{v}_n, \tag{5.4}$$

and the variance is

$$\Sigma_{v_{n+1}} = \boldsymbol{K}(t_{n+1}, t_{n+1}) - \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n)\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}\boldsymbol{K}(\boldsymbol{t}_n, t_{n+1}). \tag{5.5}$$

Note, the resulting probability density for the prediction is also Gaussian [77, p. 15]. Fig. 5.1 shows a velocity vector $\boldsymbol{v}_n$ that was sampled from $v^p(t, \varsigma)$ at $t_1 = 0, t_2 = 0.1, ..., t_{201} = 20$ with an autocorrelation defined by the kernel function with parameters $a = 10$, $b = 2$, $m = 10$ and $\beta = 6$. Moreover, the first half of the sampled data from $t_1 = 0, ..., t_{101} = 10$, was used to predict the velocity at $t_{102} = 10 + 0.1, ..., t_{201} = 20$ using Eq. (5.4). The standard deviation for the prediction was evaluated by Eq. (5.5). The figure shows the sampled velocity, the prediction and the σ interval for the prediction. Note, by the knowledge of $\boldsymbol{v}_n$, the expectation of the future velocity does not need

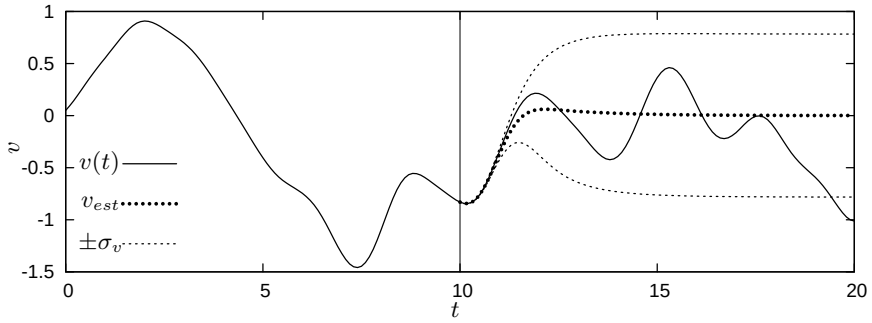

Figure 5.1.: Prediction of the velocity and standard deviation

to be zero and the covariance is greatly decreased for short-time predictions. Moreover, when retrieving the parameter vector $\boldsymbol{\alpha}_{n+1} \triangleq \boldsymbol{K}^{-1}(t_{n+1}, \boldsymbol{t}_{n+1})\boldsymbol{v}_{n+1}$ this implies

$$\mathrm{E}[\alpha_{n+1}|\boldsymbol{\alpha}_n] = 0, \tag{5.6}$$

which is consistent with Theorem 3. The coefficients are supposed to be nonzero only when there are measurements available. With Eq. (5.6) the extrapolated velocity in RKHS therefore has the continuous formulation

$$\mathrm{E}[v_{n+1}|\boldsymbol{v}_n] = \sum_{i=1}^{n} \alpha_i k(t_{n+1}, t_i), \quad \forall t_{n+1} \in \mathbb{R}. \tag{5.7}$$

5.2.2. Position estimation

Assumed, the velocity function $v(t)$ and an initial value for the position x_0 is well known the trajectory $x(t)$ can be calculated by integrating the velocity

$$x(t) = x_0 + \int_0^t v(\tau)\mathrm{d}\tau$$

or alternatively

$$x_{n+1} = x_n + \int_{t_n}^{t_{n+1}} v(\tau)\mathrm{d}\tau$$

with x_n denoting the position at time t_n. Adopting Eq. (5.7) and using Eq. (4.37) the expectation for the new position x_{n+1} at t_{n+1} becomes

$$\mathrm{E}[x_{n+1}|x_0, \boldsymbol{v}_n] = x_0 + \sum_{i=1}^{n} \alpha_i(\kappa(t_i) + \kappa(t_{n+1} - t_i)),$$

with $\kappa(t)$ defined as in Eq. (4.38). Furthermore, let

$$\boldsymbol{\kappa}_j \triangleq [\kappa(t_1) + \kappa(t_j - t_1) \cdots \kappa(t_n) + \kappa(t_j - t_n)]^{\mathrm{T}}, \tag{5.8}$$

then the estimator for the position at time t_{n+1} can be written in matrix vector denotation

$$\mathrm{E}[x_{n+1}|x_0, \boldsymbol{v}_n] = x_0 + \boldsymbol{\kappa}_{n+1}^{\mathrm{T}}\boldsymbol{\alpha}_n = x_0 + \boldsymbol{\kappa}_{n+1}^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}\boldsymbol{v}_n$$

or alternatively

$$\mathrm{E}[x_{n+1}|x_n, \boldsymbol{v}_n] = x_n + (\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n)^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}\boldsymbol{v}_n. \tag{5.9}$$

5.2.3. Dynamic Model for the target

Eq. (5.4) and Eq. (5.9) is now used to build up a discrete linear dynamic model.

State equation

Let $\boldsymbol{v}_n$ denote the velocity vector that is essential to predict $v_{n+1}^- \triangleq \mathrm{E}[v_{n+1}|\boldsymbol{v}_n]$ and $x_{n+1}^- \triangleq \mathrm{E}[x_{n+1}|x_n, \boldsymbol{v}_n]$. Moreover let $\boldsymbol{v}_{n+1}^- \triangleq \mathrm{E}[\boldsymbol{v}_{n+1}|\boldsymbol{v}_n]$ denote the expected velocity vector extended by v_{n+1}. Then, with Eq. (5.4) and Eq. (5.9) the dynamic model can be written as linear model

$$\begin{bmatrix} x_{n+1}^- \\ \boldsymbol{v}_{n+1}^- \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1} \\ 0 & \boldsymbol{I}_n \\ 0 & \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1} \end{bmatrix}}_{\boldsymbol{A}_{n+1}} \begin{bmatrix} x_n \\ \boldsymbol{v}_n \end{bmatrix}, \tag{5.10}$$

with $\boldsymbol{K}^{-1} \triangleq \boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}$ and $\boldsymbol{A}_{n+1}$ as the transition matrix. Note, the result is a vector of higher dimension than the input.

Covariance matrix

The covariance matrix for the state vector

$$\boldsymbol{W}_{n+1} \triangleq \mathrm{E}\left[\begin{bmatrix} x_{n+1}^{-} - x_{n+1} \\ \boldsymbol{v}_{n+1}^{-} - \boldsymbol{v}_{n+1} \end{bmatrix} \begin{bmatrix} x_{n+1}^{-} - x_{n+1} \\ \boldsymbol{v}_{n+1}^{-} - \boldsymbol{v}_{n+1} \end{bmatrix}^{\mathrm{T}}\right]$$

has only four non zero entries specified by

$$[\boldsymbol{W}_{n+1}]_{n+2,n+2} = \mathrm{E}\left[(v_{n+1}^{-} - v_{n+1})^2\right] = k_0 - \boldsymbol{k}_{n+1}^{\mathrm{T}} \boldsymbol{K}^{-1} \boldsymbol{k}_{n+1},$$

$$\begin{aligned}[\boldsymbol{W}_{n+1}]_{1,n+2} =& \mathrm{E}\left[(x_{n+1}^{-} - x_{n+1})(v_{n+1}^{-} - v_{n+1})\right] = \\ =& \kappa(t_{n+1} - t_n) - \boldsymbol{k}_{n+1}^{\mathrm{T}} \boldsymbol{K}^{-1} (\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n)\end{aligned}$$

and

$$\begin{aligned}[\boldsymbol{W}_{n+1}]_{1,1} =& \mathrm{E}\left[(x_{n+1}^{-} - x_{n+1})^2\right] = \\ =& 2R(t_{n+1} - t_n) - (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}}) \boldsymbol{K}^{-1} (\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n).\end{aligned}$$

with $R(t)$ defined as in *Eq.*(4.33). The calculations that lead to these results can be found in Sec. A.1.4.

For simplicity, the regularizing parameter μ was ignored in this section. However, in Sec. 4.3.2 the scaling parameter was identified to be indirectly proportional to the amplitude of the input model noise. Thus, the influence of μ on the dynamic model can be modelled by an appropriate scaling of the covariance matrix $\hat{\boldsymbol{W}}_{n+1} = \frac{1}{\mu} \boldsymbol{W}_{n+1}$.

Numerical issues

In the last sections the inverse of the Gramian matrix $\boldsymbol{K}$ was used multiple times. This is admissible as long as the Fourier transform of the kernel function is strictly positive. However, from a numerical point of view the inverse of the Gramian matrix $\boldsymbol{K}$ can be difficult to obtain because most of the eigenvalues are close to zero. This can be resolved by applying regularization.

Instead of treating the velocity vector $\boldsymbol{v}_n$ as an exact quantity $\boldsymbol{v}_n$ can also be interpreted as a measurement of the true velocity vector $\hat{\boldsymbol{v}}_n$ that is subject to very small additive error

$$\boldsymbol{v}_n = \hat{\boldsymbol{v}}_n + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \in \mathbb{R}^n$$

with $\mathrm{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$ and a covariance matrix $\mathrm{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}\right] = \gamma \boldsymbol{I}$. Consequently, the expectation becomes

$$\mathrm{E}[v_{n+1} | \boldsymbol{v}^n] = \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n)(\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n) + \gamma \boldsymbol{I})^{-1} \boldsymbol{v}_n, \tag{5.11}$$

and the variance

$$\Sigma_{v_{n+1}} = \boldsymbol{K}(t_{n+1}, t_{n+1}) - \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n)(\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n) + \gamma \boldsymbol{I})^{-1} \boldsymbol{K}(\boldsymbol{t}_n, t_{n+1}). \quad (5.12)$$

This leads to a better conditioning for inverting $\boldsymbol{K}$ [77, p. 16]. The error implemented by this reformulation can be set by the additional parameter γ that balances the accepted grade of deviation from the original velocities $\hat{\boldsymbol{v}}^n$.

The Ornstein-Uhlenbeck process

As noted before, the Ornstein-Uhlenbeck process is a special case of the constructed velocity process $v^p(t, \varsigma)$. Because of its relevance the state equation and the covariance matrix for this process is introduced in short. The basic assumption for this model is a Gauss Markov process representing the velocity $v(t, \varsigma)$ with the autocorrelation function [37, p. 95,201-203]

$$\boldsymbol{r}_v(t_i, t_j) \triangleq \sigma^2 e^{-\zeta |t_i - t_j|}. \quad (5.13)$$

With $\zeta \triangleq \frac{\beta}{m}$ and $\sigma^2 \triangleq \frac{1}{2\beta m}$ Eq. (5.13) becomes the autocorrelation function from Eq. (4.31). In analogy to Sec. 5.2.2 the process is integrated and the resulting dynamic model is described by

$$\begin{bmatrix} x_{n+1}^- \\ v_{n+1}^- \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \frac{1}{\zeta}(1 - e^{-\zeta \Delta t}) \\ 0 & e^{-\zeta \Delta t} \end{bmatrix}}_{\boldsymbol{A}_{n+1}} \begin{bmatrix} x_n \\ v_n \end{bmatrix}, \quad (5.14)$$

with $\Delta t \triangleq t_{n+1} - t_n$. The covariance matrix for the estimate is a 2×2 matrix with the elements

$$\begin{aligned}
[\boldsymbol{W}_{n+1}]_{2,2} &= \mathrm{E}\left[(v_{n+1}^- - v_{n+1})^2\right] = \\
&= \sigma^2 (1 - e^{-2\zeta \Delta t}), \\
[\boldsymbol{W}_{n+1}]_{1,2} &= [\boldsymbol{W}_{n+1}]_{2,1} = \mathrm{E}\left[(x_{n+1}^- - x_{n+1})(v_{n+1}^- - v_{n+1})\right] = \\
&= 2\sigma^2 \left[\frac{1}{\zeta}(1 - e^{-\zeta \Delta t}) - \frac{1}{2\zeta}(1 - e^{-2\zeta \Delta t})\right], \\
[\boldsymbol{W}_{n+1}]_{1,1} &= \mathrm{E}\left[(x_{n+1}^- - x_{n+1})^2\right] = \\
&= \frac{2\sigma^2}{\zeta} \left[\Delta t - \frac{2}{\zeta}(1 - e^{-\zeta \Delta t}) + \frac{1}{2\zeta}(1 - e^{-2\zeta \Delta t})\right].
\end{aligned}$$

Note, this well known model represents a special case of the proposed model where the limitation in power and bandwith of the force is not implemented. With regard to the eigenvalue decay from Fig. 4.16, minimization of the regularized risk functional using the kernel from Eq. (5.13) proves to be poorly conditioned. A kernel function representing the Ornstein-Uhlenbeck process is also applied for comparison in Chapter 6 of this thesis.

5.2.4. Evaluating the Markov property

For evaluating the state equation Eq. (5.10) the entire velocity vector $\boldsymbol{v}_n \triangleq [v_1 \cdots v_n]^{\mathrm{T}}$ needs to be properly known. Recursive Bayesian estimation as introduced in Sec. 2.4.2 uses a finite dimensional state space description. For appliance, the *Markov property* of the respective process is therefore essential. For the problem at hand this means that there is a number $l \in \mathbb{N}$ such that $\mathrm{p}(v_{n+1}|v_n, ..., v_1) = \mathrm{p}(v_{n+1}|v_n, ..., v_{n-l+1})$. Unfortunately, this property can not be shown for the constructed processes.

$v^p(t,\varsigma)$ does not fulfill the Markov property

By construction of $v^p(t,\varsigma)$, the probability density $\mathrm{p}(v_{n+1}|\boldsymbol{v}_n)$ for the estimate is Gaussian and can be fully described by Eq. (5.11) and Eq. (5.12). Let

$$\boldsymbol{w}_n^{\mathrm{T}} = [w_1 \cdots w_n] \triangleq \boldsymbol{K}(t_{n+1}, \boldsymbol{t}_n)\boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n)^{-1}$$

denote the update vector for estimating v_{n+1} such that $\mathrm{E}[v_{n+1}|\boldsymbol{v}_n] = \boldsymbol{w}_n^{\mathrm{T}}\boldsymbol{v}_n$. By definition, $\boldsymbol{w}_n$ only depends on the sampling times $t_1, ..., t_{n+1}$ and the kernel function. As $\boldsymbol{w}_n$ weights the influence of the velocity vector $\boldsymbol{v}_n$ on $\mathrm{p}(v_{n+1}|\boldsymbol{v}_n)$ it also indicates the Markov property of the underlying process.

Assumed, there is a number l such that $w_i = 0$ for $1 < i \leq n - l$ for all possible samplings $t_1, ..., t_n$, then the stochastic process $v^p(t,\varsigma)$ fulfills the Markov condition of order l. By the definition of $\boldsymbol{w}_n$, this coincides with the existence of coefficients $w_i \in \mathbb{R}$ such that

$$k(t_{n+1}, t) = \sum_{i=n-l+1}^{n} w_i k(t_i, t), \quad \forall t \in \{t_1, ..., t_n\}.$$

This can be shown for a special choice of parameters for the kernel function. Assumed the kernel function becomes the autocorrelation of the Ornstein Uhlenbeck process from Eq. (4.31) with

$$k(t', t) \triangleq \frac{1}{2\beta m} e^{-\left|\frac{\beta(t'-t)}{m}\right|} = \frac{1}{2\beta m} e^{-\frac{\beta(t'-t)}{m}}, \quad t' \in \{t_n, t_{n+1}\},$$

then

$$k(t_{n+1}, t) = e^{\frac{-\beta(t_{n+1}-t_n)}{m}} k(t_n, t) = w_n k(t_n, t).$$

For the constructed kernel this property can not be shown in general. Fig. 5.2 shows the vector $\boldsymbol{w}_n$ for a sampling $t_1 = 0, t_2 = 0.35, ..., t_{286} = 99.75$ and the kernel function with the parameters $a = 10$, $b = 2$, $m = 10$ and $\beta = 6$. In addition, the influence of the additional parameter γ introduced in Eq. (5.11).

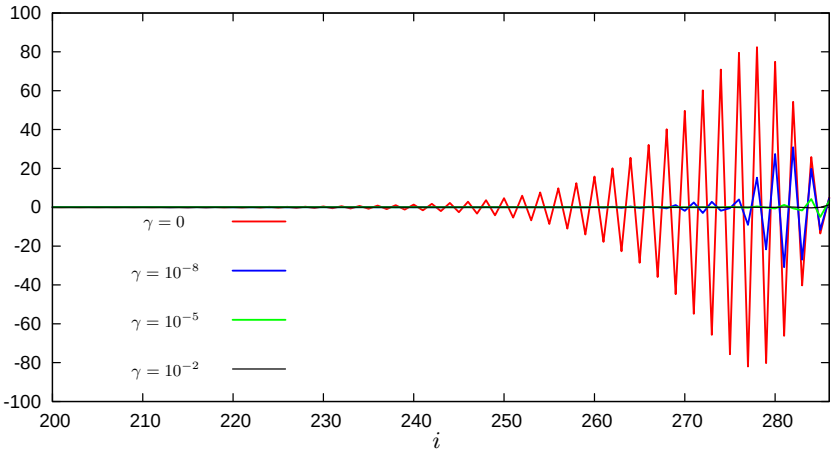

Figure 5.2.: Cutted update vector $\boldsymbol{w}_n$ for different γ

The plot indicates that the influence of past velocities vanishes over time. This is comprehensible as $k(t_i, t_j) \approx 0$ for great time differences $|t_i - t_j|$. Almost no correlation is therefore expected between $v(t_i)$ and $v(t_j)$. Furthermore, the parameter γ is greatly increasing this effect. This can be explained by the smoothing effect of the parameter on the high frequent part of the update vector $\boldsymbol{w}_n$. A cutted version of the update vector $\check{\boldsymbol{w}}_n^{\mathrm{T}} \triangleq [w_{n-l+1} \cdots w_n]^{\mathrm{T}}$ applied on a cutted velocity vector $\check{\boldsymbol{v}}_n^{\mathrm{T}} \triangleq [v_{n-l+1} \cdots v_n]^{\mathrm{T}}$ is expected to yield almost the same result as $\boldsymbol{w}_n$ applied on $\boldsymbol{v}_n$.

$$\check{\boldsymbol{w}}_n^{\mathrm{T}} \check{\boldsymbol{v}}_n \approx \boldsymbol{w}_n^{\mathrm{T}} \boldsymbol{v}_n = \mathrm{E}[v_{n+1} | \boldsymbol{v}_n] \,.$$

In particular this is true for large γ.

A toy example

Consider a setting with $\boldsymbol{v}_n = [v_1 \cdots v_{286}]^{\mathrm{T}}$ sampled from $v^p(t, \varsigma)$ at times $t_1 = 0, t_2 = 0.35, ..., t_{286} = 99.75$. It is further assumed that $v_1, ..., v_{272}$ are available quantities and $v_{273}, ..., v_{286}$ need to be estimated. The estimation is performed using Eq. (5.11) with a kernel function defined by the parameters $a = 10$, $b = 2$, $m = 10$ and $\beta = 6$. Fig. 5.3 shows the original sampling $\boldsymbol{v}_n$ and the estimates for different values of the additional parameter γ as indicated in the plot.

Although the update vectors $\boldsymbol{w}_n$ for $\gamma = 0$ and for $\gamma = 10^{-8}$ greatly differ in value as illustrated in Fig. 5.2 the estimates are almost identical. For an increasing parameter γ the estimation error increases as expected. The update vector with $\gamma = 10^{-8}$ delivers accurate results for this setup with a greatly

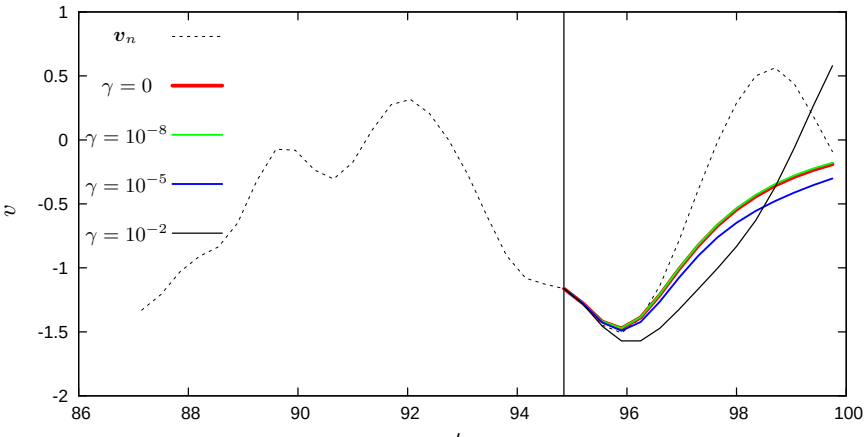

Figure 5.3.: Estimation for different update vectors

simplified update vector. $\gamma = 10^{-8}$ was therefore used for the next plot to visualize the influence of the parameter l.

To this end, the stochastic process was treated as a Markov process of different orders defined by l. The update vector was cutted to a fixed length l and then used for extrapolation. Fig. 5.4 shows the results for different lengths as indicated within the plot.

Although disregarding older velocities, the plot shows that for the investigated case a constant vector length of $l = 50$ almost delivers the same result as the uncutted update vector.

Two conclusions can be drawn from these investigations. By the nature of the process itself it is possible to find a parameter l such that the process can be treated as a Markov process of order l. Furthermore, the additional parameter γ leads to a better conditioning of the estimation problem and allows to decreases the parameter l. With this approximation, the constructed dynamic model complies with the preconditions stated for Bayesian estimation using the update equation from Eq. (2.27).

5.2.5. EKF Filtering

The best known approach for resolving the Bayesian update equation is the EKF that was already mentioned in Sec. 2.4.3. This approach is applied for online estimation of the ROLAM problem. Note, as the observation model for ROLAM is not linear EKF filtering can not be considered to be optimal with respect to the problem. However, for simplicity, more sophisticated methods like UKF's or PF's are not considered within this thesis.

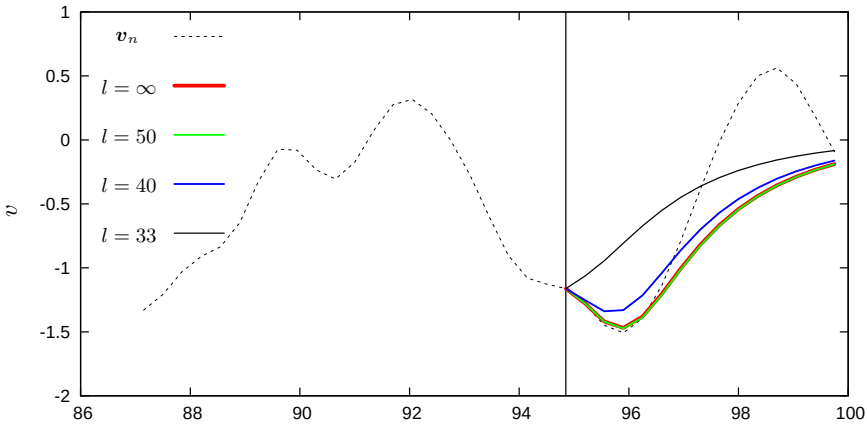

Figure 5.4.: Estimation using cutted update vector

State vector

The model from Eq. (5.10) only considers one dimension for the target parameters x_n and $\boldsymbol{v}_n$. Furthermore, ROLAM also requires to incorporate the unknown beacon positions $\boldsymbol{p}^{\langle k \rangle}$ into the state vector. This way, the initial estimate for the beacon positions can be corrected in real time by measurement updates.

The state vector at timestep n is therefore expanded to three dimensions

$$\boldsymbol{s}_n \triangleq \begin{bmatrix} \boldsymbol{s}_{1,n}^{\mathrm{T}} & \boldsymbol{s}_{2,n}^{\mathrm{T}} & \boldsymbol{s}_{3,n}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{5.15}$$

For each dimension j the state vector is further expanded by the beacon positions at timestep n

$$\boldsymbol{s}_{j,n} \triangleq \begin{bmatrix} x_{j,n} & \boldsymbol{v}_{j,n}^{\mathrm{T}} & \boldsymbol{p}_{j,n}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad j \in \{1,2,3\}.$$

Within this context, $\boldsymbol{v}_{j,n}$ is denoting the cutted velocity vector with a fixed length as introduced in Sec. 5.2.4. Moreover, to obtain a unique solution a proper definition of a coordinate system is required. To this end, the coordinate system is defined according to Sec. 3.1.2. This implies a predefinition of some of the beacons coordinates of the form

$$\boldsymbol{p}_{j,n} \triangleq [p_{j,1+j} \cdots p_{j,k}]_n^{\mathrm{T}}.$$

Note, by definition of the coordinate system the state vector is of different dimension for every axis.

Dynamic model for the beacons

Because the beacons are assumed static a dynamic model of the form

$$\boldsymbol{p}_i^- = \boldsymbol{p}_i \quad \forall 1 \leq i \leq k$$

is sufficient for the application. Subsequently, the estimation error from the model becomes zero for each dimension

$$\mathrm{E}\left[(p_i^- - p_i)^2\right] = \epsilon_{p_i}, \quad \epsilon_{p_i} \triangleq 0. \tag{5.16}$$

Note, slight movements of the beacons with low dynamics can be incorporated by using an estimation error $\epsilon_{p_i} > 0$. Nevertheless, the potential of such an approach is not be examined here and may be of interest in future investigations.

State equation

Consequently, the transition matrix for each dimension is added by an appropriate identity matrix such that

$$\boldsymbol{s}_{j,n+1}^- = \begin{bmatrix} \boldsymbol{A}_{n+1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{k-j} \end{bmatrix} \boldsymbol{s}_{j,n}, \quad \boldsymbol{I}_{k-j} \in \mathbb{R}^{(k-j)\times(k-j)},$$

with $\boldsymbol{I}_{k-j}$ denoting the identity matrix. Expanded to three dimensions the state equation becomes

$$\boldsymbol{s}_{n+1}^- = \underbrace{\begin{bmatrix} \boldsymbol{A}_{n+1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{k-1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{n+1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{k-2} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{n+1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{k-3} \end{bmatrix}}_{\boldsymbol{F}_{n+1}} \boldsymbol{s}_n + \boldsymbol{\zeta}_n,$$

with $\boldsymbol{\zeta}_n$ as the noise vector with covariance

$$\boldsymbol{Q}_{n+1} = \begin{bmatrix} \frac{1}{\mu_1}\boldsymbol{W}_{n+1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \frac{1}{\mu_2}\boldsymbol{W}_{n+1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{1}{\mu_3}\boldsymbol{W}_{n+1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Note, the transition matrix $\boldsymbol{F}_{n+1}$ and covariance matrix $\boldsymbol{Q}_{n+1}$ can become bulky for evaluation because of its sheer size. This is especially true when the number

of beacons of unknown position is increasing during estimation as the number of elements in both matrices increase quadratically. However, by construction of the dynamic model, the three dimensions for target estimation can also be evaluated separately. Although further decoupling of the variables might be possible, this is not the focus here and therefore not pursued further in this thesis.

Observation Model

To obtain reasonable results, external information has to be to be incorporated. Compliant to the formulation as state observer from Eq. (2.2) this is performed in form of the observation model

$$\boldsymbol{r}_i = h(\boldsymbol{s}_i) = ||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| + \eta_i.$$

η_i denotes the noise of the measurements with the probability density function $\mathrm{p}(\eta_i) = \mathcal{N}(\eta_i; 0, \sigma_i)$. For the covariance update in the EKF a linearization of the observation model is required. As the observation is of one-dimensional kind, the gradient

$$\boldsymbol{h}_{n+1} \triangleq [\nabla_{\boldsymbol{s}} h(\boldsymbol{s})] \,|_{\boldsymbol{s}=\boldsymbol{s}^-_{n+1}}$$

defines the required quantity. Except for the elements

$$\frac{\partial h(\boldsymbol{s})}{\partial x_{j,n+1}} = \frac{x_{j,n+1} - p_{j,a_{n+1}}}{||\boldsymbol{x}_{n+1} - \boldsymbol{p}_{a_{n+1}}||}, \quad \frac{\partial h(\boldsymbol{s})}{\partial p_{j,a_{n+1}}} = -\frac{x_{j,n+1} - p_{j,a_{n+1}}}{||\boldsymbol{x}_{n+1} - \boldsymbol{p}_{a_{n+1}}||}$$

the gradient is zero valued for all $j \in \{1, 2, 3\}$.

Filter equations

With these quantities, the EKF filter equations can be directly applied to the ROLAM problem [39, p. 20].

$$\begin{aligned}
\boldsymbol{s}^-_{n+1} &= \boldsymbol{F}_{n+1}\boldsymbol{s}_n \\
\boldsymbol{C}^-_{n+1} &= \boldsymbol{Q}_{n+1} + \boldsymbol{F}_{n+1}\boldsymbol{C}_n\boldsymbol{F}^{\mathrm{T}}_{n+1} \\
\boldsymbol{s}_{n+1} &= \boldsymbol{s}^-_{n+1} + \boldsymbol{G}_{n+1}(r_{n+1} - h(\boldsymbol{s}^-_{n+1})) \\
\boldsymbol{C}_{n+1} &= \boldsymbol{C}^-_{n+1} - \boldsymbol{G}_{n+1}\boldsymbol{S}_{n+1}\boldsymbol{G}^{\mathrm{T}}_{n+1},
\end{aligned}$$

where

$$\boldsymbol{S}_{n+1} = \boldsymbol{h}^{\mathrm{T}}_{n+1}\boldsymbol{C}^-_{n+1}\boldsymbol{h}_{n+1} + \sigma_{n+1}$$

and

$$\boldsymbol{G}_{n+1} = \boldsymbol{C}^-_{n+1}\boldsymbol{h}_{n+1}\boldsymbol{S}^{-1}_{n+1}.$$

Note, as the state consists of multiple velocities of the past, the past is improved over time as well.

Initialization

The EKF requires normally distributed initial values for proper results. The initialization is performed by an offline algorithm that uses all available data followed by an EKF that is iteratively incorporating new range measurements. This two step optimization process using a *Levenberg-Marquardt algorithm* (LM) and an *Extended Kalman Filter* (EKF) is depicted in Fig. 5.5.

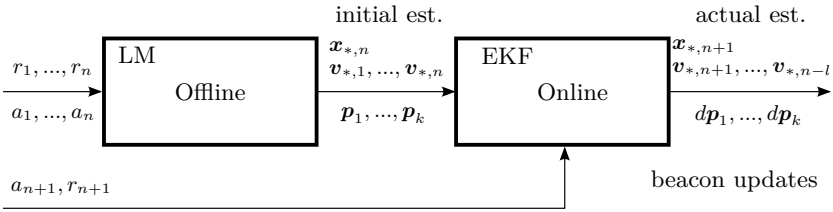

Figure 5.5.: Optimization scheme in two steps

Unfortunately, the required number of observations for proper initialization can not be quoted a priori. This is dependent on the concrete scenario. For example, a straight movement along a line never leads to a satisfactory initial solution as it leaves a rotational degree of freedom for the beacon positions around that line. Consequently, the resulting probability density for the beacon positions can not be approximated by a normal distribution. However, when using the EKF this is required. In the worst case, the offline solution has to be recalculated multiple times until all probability density functions for the estimated variables can be approximated by a Gaussian density function. The covariance matrix of the initial estimate for the state vector obtained by Sec. 5.1.2 may give a good hint of when this assumption is fulfilled. The benefit of a proper initialization is observed in Chapter 6 of this thesis.

5.2.6. Coincidence of solutions

In the former section the EKF for online estimation is designed to approximate the minimum of the regularized risk. Nevertheless, this section is dedicated to clarify the relation between the results obtained by the offline and online approach. Indeed, both solutions coincide approximately.

Immediate and Final Solution

Consider the following simplified scenario. Let $x(t) \in \mathbb{R}$ denote a trajectory that has to be estimated in one dimension using n measurements $\boldsymbol{z}_n \triangleq [z_1 \cdots z_n]^{\mathrm{T}}$ at

times $t_1, ..., t_n$. The simplified observation model is $z_i = h(x(t_i)) \triangleq x(t_i) + \eta_i$. Within this scenario, the measurements are mutually independent and η_i is normally distributed such that $\eta_i \sim \mathcal{N}(0, \sigma^2)$ and $\mathrm{E}[\eta_i \eta_j] = 0, \forall i \neq j$. According to Eq. (4.37) the trajectory can be represented by

$$x(t) = x_0 + \sum_{i=1}^{n} \alpha_i (\kappa(t_i) + \kappa(t - t_i)),$$

with $\kappa(t)$ defined as in Eq. (4.38). Accordingly, the one-dimensional velocity function becomes

$$v(t) = \sum_{i=1}^{n} \alpha_i k(t, t_i).$$

The initial value x_0 and the coefficient vector $\boldsymbol{\alpha} \triangleq [\alpha_1 \cdots \alpha_n]^{\mathrm{T}}$ are obtained by minimizing the simplified regularized risk functional

$$R_{\mathrm{reg}}[\boldsymbol{x}_n] \triangleq \sum_{i=1}^{n} (z_i - x_i)^2 + \mu \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}, \tag{5.17}$$

with $\boldsymbol{x}_n \triangleq [x_1 \cdots x_n]^{\mathrm{T}}$ and $x_i \triangleq x(t_i)$. The covariance matrix of the solution is estimated by the inverse Hessian matrix of $S \triangleq \frac{1}{2} R_{\mathrm{reg}}[\boldsymbol{x}]$ as described in Sec. 5.1.2.

Let $\check{\boldsymbol{x}}_n, \check{\boldsymbol{v}}_n$ denote the minimum of Eq. (5.17) using all measurements $\boldsymbol{z}_n$ and $\check{\boldsymbol{x}}_k, \check{\boldsymbol{v}}_k$ the minimum using the subset $\boldsymbol{z}_k \triangleq [z_1 \cdots z_m]^{\mathrm{T}}$ with $m \triangleq \frac{1}{2} n$ only. Fig. 5.6 shows a model calculation using a kernel function with parameters $a = 10$, $b = 1$, $m = 10$ and $\beta = 6$.

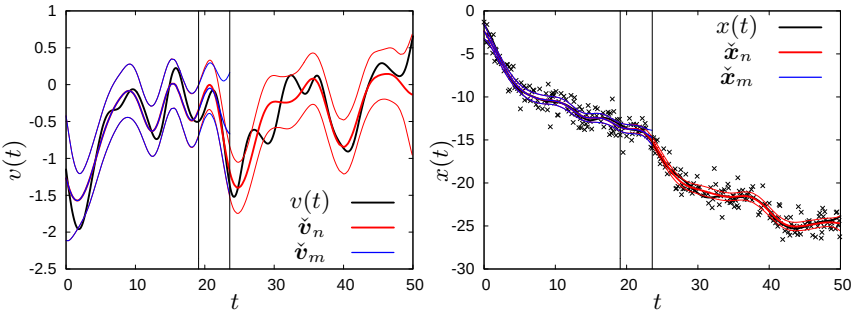

Figure 5.6.: Estimates for $v(t)$ and $x(t)$ using n and m measurements

For clarity, an area of 2σ was plotted by a thinner line for each case. Basically, both solution do not coincide as they both make use of different data sets.

However, until a certain timestep t_{m-l} both solutions seem to be almost identical. t_{m-l} is indicated by a vertical line within the plot. From that point the solutions drift apart such that at time t_m, which is also indicated by a vertical line, the solutions are not identical any more. This is obvious as for $\check{\boldsymbol{v}}_n$ and $\check{\boldsymbol{x}}_n$ future data can be used to further improve the estimate. For a better view, Fig. 5.7 depicts the same plot within the area between t_{m-l} and t_m.

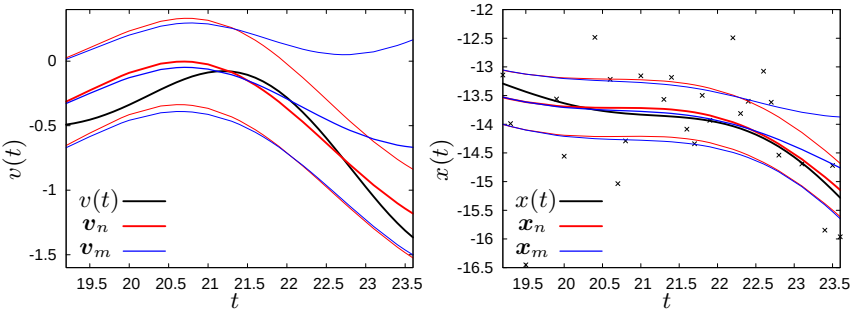

Figure 5.7.: Zoomed area of Fig. 5.6

At timestep t_m and without knowing the future, the best estimate for the trajectory that can be obtained by kernel regression is indicated by $[\check{\boldsymbol{x}}_m]_m, [\check{\boldsymbol{v}}_m]_m$. This estimate is called *immediate solution*. After further measurements the estimate can be improved when using kernel regression. As the influence of future data on the past vanishes over time, see also Sec. 5.2.4 on this topic, the solution converges to a certain point at time t_m which is indicated by $[\check{\boldsymbol{x}}_n]_m, [\check{\boldsymbol{v}}_n]_m$. The final estimate for the trajectory at t_m using all future data is called *final solution*.

The solutions coincide

By design, the EKF introduced in Sec. 5.2.5 iteratively approximates the minimum of the regularized risk functional. In analogy to the simplified problem definition from Sec. 5.2.6 a version of the EKF for solving the one-dimensional problem can be derived. According to this, the state vector becomes $\boldsymbol{s}_n \triangleq [x_n \quad \boldsymbol{v}_n]^{\mathrm{T}}$ and dynamic model is identical to Eq. (5.10). The observation model can be described by the linear equation $\boldsymbol{z}_n = \boldsymbol{H}\boldsymbol{s}_n$ with $\boldsymbol{H} \triangleq \begin{bmatrix} 1 & \mathbf{0}_n^{\mathrm{T}} \end{bmatrix}$ with $\mathbf{0}_n \triangleq [0 \cdots 0]^{\mathrm{T}}$.

This simplified EKF was applied to the one-dimensional problem from the former section. By definition of the state vector, the immediate estimate for the position, the immediate estimate for the velocity and the final solution for the

velocity are entries of the state vector. The final solutions for the position can be retrieved by backward integration starting at the actual position and using the final solutions for the velocity. Both, the immediate solution as well as the final solution obtained online by the EKF are plotted in Fig. 5.8. The initial estimate for the state vector was obtained by offline minimization using the observations $z_1, ..., z_m$ as proposed in Sec. 5.2.5. For comparison, the immediate and final solution obtained by Levenberg Marquardt (LM) minimization was plotted too.

As expected, the immediate and the final solutions for the EKF coincide with the solutions that obtained by the LM method. Slight deviations originate from the approximations mentioned in the former sections and numerical insufficiency. Unsurprisingly, the final solution is of better quality than the immediate solution and of lower variance. As a logical consequence, all estimations finally end at the same estimate for the position at $t = t_n$.

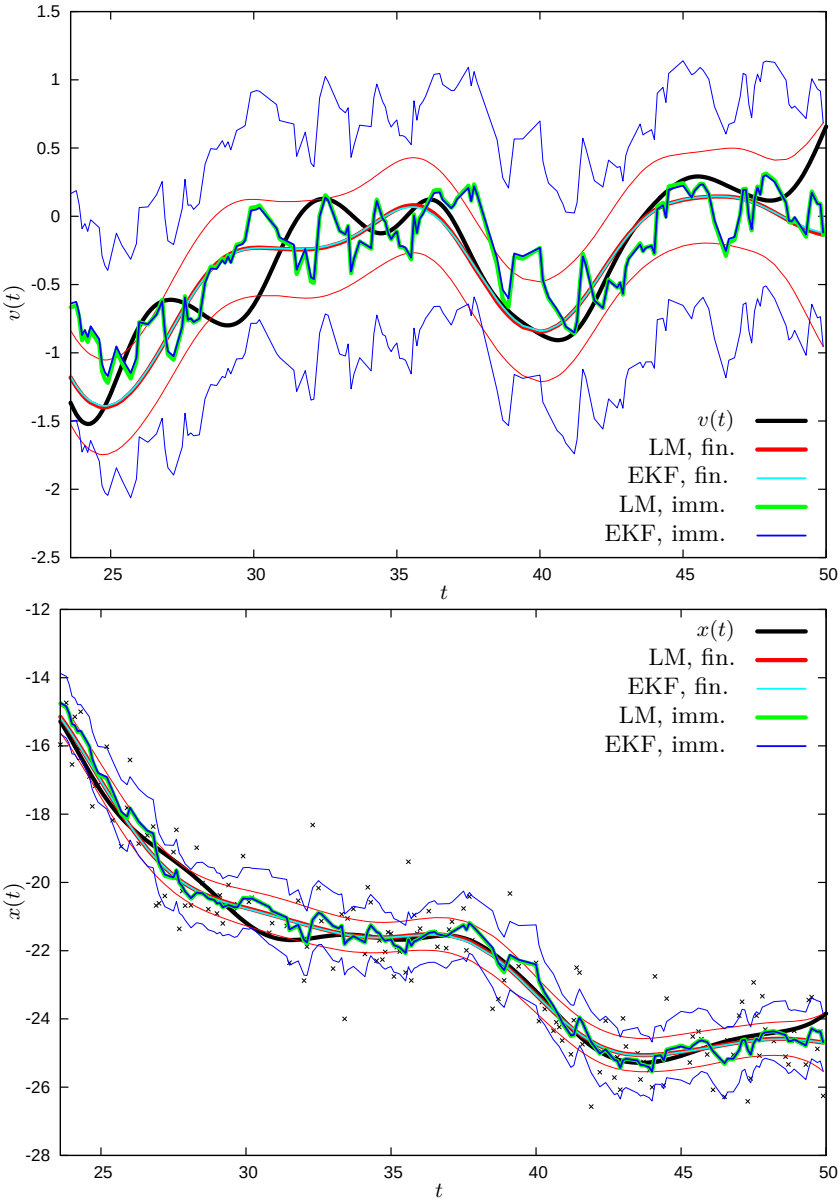

Figure 5.8.: Immediate and final solutions of the EKF and LM method

6. Test results

This chapter summarizes the tests performed within this thesis. For comparison, two alternative approaches were implemented. The ML estimator in conjunction with data alignment in time domain as proprosed in Sec. 3.2.3 and the kernel based approach featured by an alternative kernel function representing the Ornstein Uhlenbeck process.

Furthermore, the impact of the initialization from Fig. 5.5 on the final result is investigated. To this end, the proposed EKF from Sec. 5.2 was implemented for both kernel functions.

While simulations are examined in Sec. 6.1, a real life scenario is investigated in Sec. 6.2. A scenario with UWB beacons located at unknown positions and a rover capable of measuring distances to these beacons shows the applicability of the proposed approach.

6.1. Simulations

This section presents the results for solving the ROLAM problem based on simulated scenarios.

6.1.1. Test bed

The test bed for the simulations assumes 8 distributed beacon nodes at the positions indicated by Tab. 6.1.

Node	1	2	3	4	5	6	7	8
p_1	0	20	20	0	0	20	20	0
p_2	0	0	20	0	20	0	20	20
p_3	0	0	0	20	0	20	20	20

Table 6.1.: Beacon positions for simulation

This symmetric geometrical layout allows a good localization accuracy within the vicinity of the beacon positions. The positional dilution of precision (PDOP) for this area, as introduced in Sec. 2.2.3, ranges between $0.6 \leq \text{PDOP} \leq 2.0$ when assuming 8 range measurements for each position and well known beacon positions.

Trajectory generation

The examined trajectories were generated by integrating velocity functions sampled from $v^p(t, \varsigma)$. The kernel parameters were chosen to be $a \triangleq 10$, $b \triangleq 1$, $\beta \triangleq 6$ and $m \triangleq 10$. The sampling times for generating reference trajectories were chosen equidistantly to be $\tau_1 \triangleq 0, \tau_2 \triangleq 0.1, ..., \tau_m \triangleq 80$.

To assure a three dimensional trajectory in the center of the scenario the components of the initial value $\boldsymbol{x}_0 \triangleq [x_{1,0} \quad x_{2,0} \quad x_{3,0}]^{\mathrm{T}} \in \mathbb{R}^3$ were sampled from the probability densities $x_{1,0} \sim x_{2,0} \sim x_{3,0} \sim \mathcal{N}(10, 5^2)$. For numerical reasons the coefficients for the trajectory were sampled using the eigenvector decomposition for the Gramian matrix $\boldsymbol{K}$. Altogether, 100 trajectories were generated for the tests. Fig. 6.1 illustrates the first sample whose complex geometry can not be represented by a simple regression model.

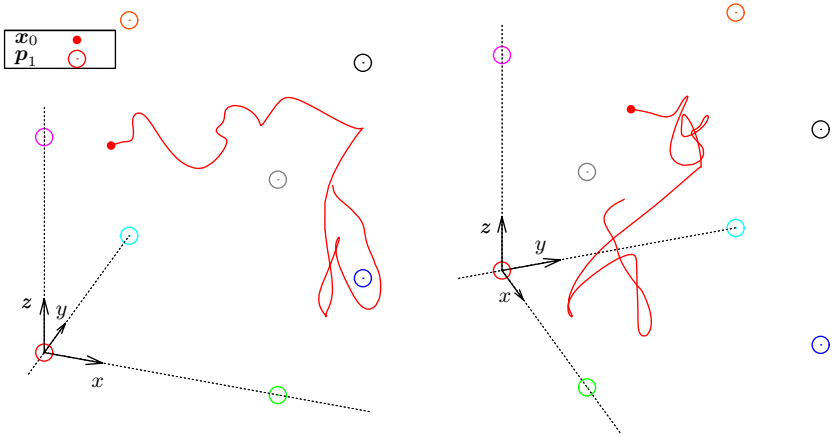

Figure 6.1.: Sampled test trajectory taken from 2 different viewpoints

Observations

To complete the simulated scenario, unsynchronized range measurements from the beacon positions including the data to measurement associations are required. It is assumed that the scenario comprises of n range measurements from $k \triangleq 8$ beacons.

Each entry of the data to measurement association vector $\boldsymbol{a} = [a_1 \cdots a_n]^{\mathrm{T}}$ is sampled from the probability distribution $P(a_i = j) = k^{-1}$ for $j \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. This way every beacon has the same probability to be used for measurement.

The times of measurements, combined in the vector $\boldsymbol{t} \triangleq [t_1 \cdots t_n]^{\mathrm{T}}$, were sampled from an equally distributed random variable within the interval $[0, 80]$.

The simulated range measurements were sampled following the postulated noise model $r_i \triangleq ||\boldsymbol{x}_i - \boldsymbol{p}_{a_i}|| + \eta_i$ introduced in Eq. (3.1). The measurement noise η_i is sampled from a random variable described by the probability density function $\mathcal{N}(0, \sigma^2)$.

Scenarios

Six different scenarios $S_n(\sigma)$ are investigated that differ in measurement noise and number of samples as described in Tab. 6.2.

$\sigma \backslash n$	400	200	100
0.25	$S_{400}(0.25)$	$S_{200}(0.25)$	$S_{100}(0.25)$
0.5	$S_{400}(0.5)$	$S_{200}(0.5)$	$S_{100}(0.5)$

Table 6.2.: Investigated scenarios for simulation

For $n = 100$, $n = 200$ and $n = 400$ the times for measurement and measurement to data associations were sampled independently.

6.1.2. Offline Estimation

The solutions obtained by kernel regression from Eq. (4.48) are compared to the ML estimator that uses alignment in time domain. Both methods were applied on all generated scenarios. The quality of the fit is measured by the positional *Root Mean Square* (RMS) error

$$\mathrm{RMS} = \sqrt{\sum_{i=1}^{n} \frac{||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||^2}{n}} \tag{6.1}$$

with $\boldsymbol{x}_i$ denoting the true position and $\hat{\boldsymbol{x}}_i$ the estimated position. The sample times for evaluation within Eq. (6.1) were chosen in accordance to the time of measurements $t_1, ..., t_n$. Because of the alignment in time, the obtained solution of the ML estimator is not defined at these times instances. To cope with this, linear interpolation in time was applied to obtain intermediate solutions.

Parameterization

For all simulations, the regularization parameters from Eq. (4.47) were chosen to be $\mu_1 \triangleq \mu_2 \triangleq \mu_3 \triangleq 1$. Two different kernel functions were investigated. The first one, denoted with $k(t_i, t_j)$ was chosen in accordance to the trajectory

model, the second $k_{ou}(t_i, t_j)$ in such a way that it almost equals the Ornstein Uhlenbeck process. The parameters are listed in Tab. 6.3.

param.	a	b	β	m
k	10	1	6	10
k_{ou}	10	10^4	6	10

Table 6.3.: Kernel parameters for the investigated kernels

Results for the aligned approach

To apply the ML-estimator on the ROLAM problem using data alignment in time, the parameter ΔT from Eq. (3.9) needs to be defined. As the optimal parameter for this quantity is dependent on the number of measurements n, noise σ and the complexity of the trajectory the minimization problem from Eq. (3.10) was solved for all scenarios with $\Delta T \in \{2, ..., 10\}$.

Fig. 6.2 shows the mean of the positional RMS error for the trajectory over all scenarios. For $n = 100$, no usable results could be achieved. For this reason these scenarios are left out of the plot. The best alignment parameter found for $n = 100$ was $\Delta T = 10$ which caused an RMS error of above 13 for both error models.

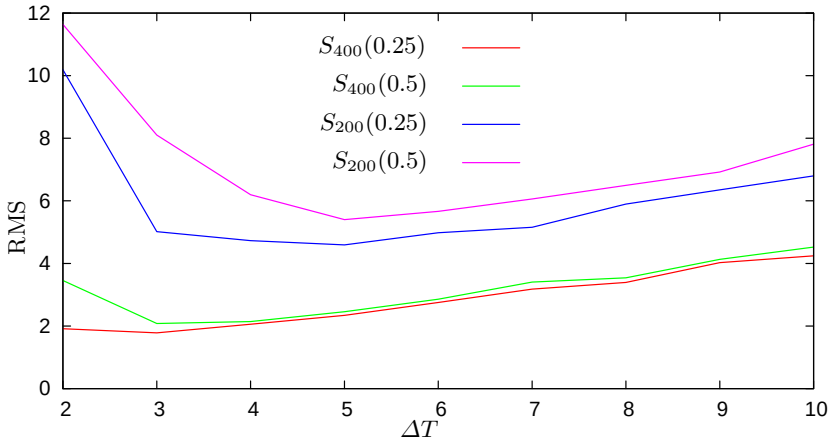

Figure 6.2.: RMS errors for different alignment parameters ΔT

The standard deviations for the results are presented in Tab. 6.4.

$\sigma \backslash \Delta T$	2	3	4	5	6	7	8	9	10
$S_{400}(0.25)$	1.57	1.02	1.76	1.52	1.68	2.62	2.60	2.69	3.73
$S_{400}(0.5)$	4.63	1.34	1.60	1.62	1.76	2.68	2.51	2.46	3.80
$S_{200}(0.25)$	8.07	3.55	3.19	4.06	4.72	7.18	5.15	5.44	5.72
$S_{200}(0.5)$	8.32	7.27	5.79	5.09	5.53	5.67	5.78	6.37	7.15

Table 6.4.: standard deviations of the RMS estimate

As expected, the RMS error for $n = 200$ is significantly higher than for $n = 400$ measurements. Further inspection of Fig. 6.2 delivers $\Delta T = 5$ as the best parameter for $n = 200$ for both noise levels. For smaller ΔT the problem becomes ill conditioned and for an alignment parameter greater than 5 the discretization error becomes dominant. For $n = 400$ the alignment parameter $\Delta T = 3$ was found to be optimal for both noise levels.

Aligned vs kernel based approach

For the kernel based approach, all scenarios were solved by minimizing the regularized risk from Eq. (4.47) applying the LM method. This was performed for both kernel functions $k(t_i, t_j)$ and $k_{ou}(t_i.t_j)$. The single RMS errors for both kernel functions are plotted in full detail in Fig. A.1 and Fig. A.2. The mean of the RMS errors for the kernel based approach and the aligned approach from Fig. 6.2 for $\Delta T = 2$ and $\Delta T = 5$ is summarized in Tab. 6.5.

scenario	k	k_{ou}	align.
$S_{400}(0.25)$	0.576	0.600	1.7811
$S_{400}(0.5)$	1.096	1.161	2.0806
$S_{200}(0.25)$	1.069	1.126	4.5938
$S_{200}(0.5)$	1.886	1.910	5.4005
$S_{100}(0.25)$	2.865	3.616	13.281
$S_{100}(0.5)$	4.451	5.023	13.646

Table 6.5.: Resulting RMS errors using k, k_{ou} and ML estimation

While both of the kernel based approaches significantly perform better, the difference of the results for k and k_{ou} is marginal. Fig. A.3 depicts the solution using $k(t_i, t_j)$ on sample 97 within scenario $S_{200}(0.25)$. Fig. A.4 shows the solution with respect to the ranges. The range measurements, the estimated ranges and the true ranges are plotted for all 8 beacons. For comparison the solution using the ML estimator after alignment was plotted in Fig. A.5 and Fig. A.6.

The RMS errors for all methods increase with the noise of the measurements.

However, Fig. A.1 and Fig. A.2 also reveal other influencing factors on the quality of the solution.

First of all, the influence of the sampling times becomes visible when comparing the similarity of the resulting RMS errors with the solutions for $n = 100$, $n = 200$ and $n = 400$. Similarities of the RMS errors that occur over all samplings indicate a geometrical feature with the trajectory. For example, Fig. A.7 shows the solution using $k(t_i, t_j)$ on sample 97 within the scenario $S_{100}(0.5)$. Again Fig. A.8 shows the solution with respect to the ranges. In consideration of the amount of data, the result shows a very good performance for this setup.

However, trajectories that do not sufficiently stimulate all 3 degrees of freedom and trajectories that suffer from a high DOP are poorly conditioned. As an example for both cases Fig. A.9 and Fig. A.11 show the results for sample 44 and 83 within $S_{100}(0.5)$. Fig. A.10 and Fig. A.12 show the results with respect to the ranges.

As noted before, there is almost no visual difference between k and k_{ou} within the scenarios with $n = 200$ and $n = 400$. This is due to the close relationship between both methods. However, with regard to the eigenvalue decay from Sec. 4.4.2 this also indicates further potential in reducing the number of measurements when using $k(t_i, t_j)$. As expected, the difference increases for $n = 100$.

6.1.3. Initialization impact on the EKF

A two step approach was suggested in Sec. 5.2.5 for solving ROLAM. For simplification, one might consider to apply the online approach only with random initialization values. However note, the observation model is non linear. Therefore the linearization performed within the EKF may introduce significant errors such that an arbitrary initialization may lead to slow convergence or even to divergence of the EKF (see [87, 88]).

EKF with random initialization

Divergence of the EKF with random initialization was also observed for the ROLAM problem. The proposed EKF from Sec. 5.2.5 was applied to all samples of scenario $S_{400}(0.25)$. The parameter l that determines the length of the state vector was chosen to be $l = 50$. Each element of the initial state vector $\boldsymbol{s}_0$ for the EKF filter was randomly initialized by a sample according to $[\boldsymbol{s}_0]_i = \mathcal{N}(0, 10^2), \quad \forall i \in \{1, ..., 3l + 3k - 6\}$. The initial covariance matrix $\boldsymbol{C}_0$ was chosen to be a diagonal matrix with $[\boldsymbol{C}_0]_{ii} = 100$ and $[\boldsymbol{C}_0]_{ij} = 0, \forall i \neq j$. Fig. 6.3 shows the absolute errors of the final position estimate versus the samples.

For none of the samples a satisfactory result could be achieved. While the data sets for $S_{400}(0.25)$ were sufficient for calculating usable results using an offline approach the proposed EKF with random initialization did not converge using

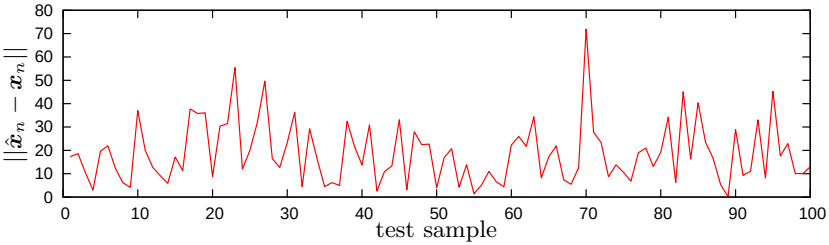

Figure 6.3.: Absolute errors of the final position estimate

the same data. The reason can be found in the non-linearity of the observation model.

A toy example

Consider a simplified two dimensional scenario as illustrated in Fig. 6.4.

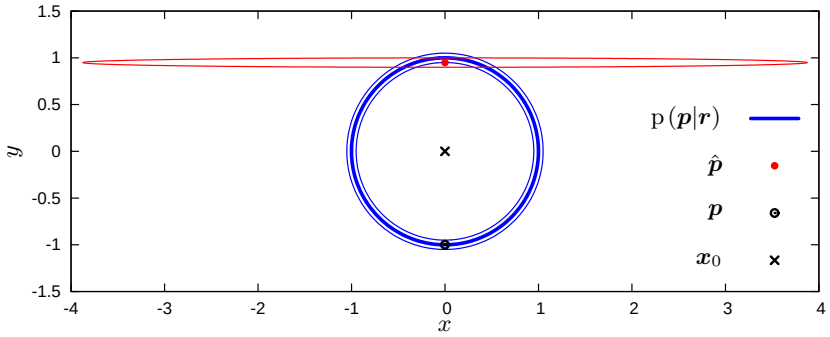

Figure 6.4.: True probability density and probability density within the EKF

A target with known position at $\boldsymbol{x}_0 = [0 \quad 0]^{\mathrm{T}}$ is standing still and measuring ranges $r_i = ||\boldsymbol{x}_0 - \boldsymbol{p}|| + \eta_i$ with $\eta_i = \mathcal{N}(0, 0.5^2)$ to a beacon at the position $\boldsymbol{p} = [p_1 \quad p_2]^{\mathrm{T}} = [0 \quad -1]^{\mathrm{T}}$. It is assumed, $\boldsymbol{p}$ is a priori unknown and to be estimated by an EKF. $n = 100$ range measurements were taken for this test.

The true probability density function for the position of the beacon $\mathrm{p}(\boldsymbol{p}|\boldsymbol{r})$ using the measurements is rotational invariant around the position of the target. This is indicated within the plot with mean and standard deviation.

After $n = 100$ steps of the EKF the estimate is represented by a normal distribution as indicated in Fig. 6.4 by the estimate $\hat{\boldsymbol{p}}$ and its covariance ellipse.

The initial estimate for the beacon position was chosen to be $\boldsymbol{p}_0 = [0 \quad 1.5]^{\mathrm{T}}$ and the initial covariance $\mathrm{E}\left[p_1^2\right] = \mathrm{E}\left[p_2^2\right] = \sigma^2 \triangleq 8^2$ and $\mathrm{E}[p_1 p_2] = 0$. The EKF misleadingly confines the beacons position at $y = 1$.

A particle filter which is known to be able to deal with non-linearities may be implemented as a conceptual alternative. However, this was not further investigated within this thesis.

EKF with initialization using an initial offline solution

The EKF was also applied on scenario $S_{400}(0.25)$ with an initialization as proposed by the two step approach from Sec. 5.2.5. The first half of the data $r_1, a_1, ..., r_{200}, a_{200}$ was used for offline estimation using the LM method. The second half $r_{201}, a_{201}, ..., r_{400}, a_{400}$ for the EKF. For comparison, all scenarios were also solved using the offline approach for all available data.

As mentioned before, the estimates for the beacon positions are the critical quantity. If the initial estimates can not be treated as a normally distributed random variables, the EKF can not produce accurate results. For 80 out of 100 samples the initialization resulted in an estimate that was almost identical to the estimate obtained by offline optimization. For these samples, Tab. 6.6 shows the mean of the RMS errors for the estimate of the beacon positions.

RMS_{init}	RMS_{final}	RMS_{off}	RMS_{dist}
1.3717	0.7410	0.6733	0.3318

Table 6.6.: RMS errors of beacon estimates

In the tabular, RMS_{init} denotes the RMS error for the initial estimate and RMS_{final} for the final estimate using the EKF. RMS_{off} denotes the RMS error for the estimate obtained by the LM minimization and RMS_{dist} the RMS distance between offline estimate and final estimate from the EKF. These values show how the initial estimates for the beacons were improved by the EKF.

As an example, Fig. A.13 depicts the result for sample 59. The plot shows the initial and the immediate solution for the two step approach using the EKF and the true trajectory. Note, while the initial solution for beacons $\tilde{\boldsymbol{p}}_j$ with $1 \leq j \leq k$ and trajectory $\tilde{x}$ is shifted with respect to true trajectory the final estimates for the beacons $\hat{\boldsymbol{p}}_j$ and the immediate solution for the trajectory $\hat{\boldsymbol{x}}$ converge to the true beacon positions and true trajectory respectively.

Although the observation model is non-linear, the two step approach introduced in Sec. 5.2.5 proves to perform well for the tested scenarios.

6.2. Real world applications

A real world application based on UWB signals is investigated in this section. The test environment was part of an European Project funded by the *Information Society Technologies* (IST) programme of the *6th Frame Programme* (FP6) called PULSERS (*Pervasive Ultra-wideband Low Spectral Energy Radio Systems*) PHASE 2.

6.2.1. Test bed

The investigated test scenario was part of work package 3b within the PULSERS project. It comprises of the mobile rover P3AT from the company *MobileRobots* and is depicted in Fig. 6.5.

Figure 6.5.: Moving platform used for the test

The robot was additionally equipped with a reference navigation system to quantize the localization results. Wheel encoders within every wheel combined with a 2D laser scanner allow absolute positioning of the rover in 2 dimensions with an error of less than $50mm$. To this end, the reference navigation system requires a 2D map. This was provided for the room pictured in Fig. 6.6.

During the tests, the rover was steered within an area of about 4×4 meters. 8 fixed beacons were installed on tripods for UWB positioning at the positions written in Tab. 6.7.

Figure 6.6.: Test area

Error model of the observations

For all tests *Line Of Sight* (LOS) conditions were ensured. However, the hitherto existing assumption of normally distributed noise for the range measurements could not be maintained for the real life scenario. Instead, the measurement error was modelled by an exponential distribution in accordance to measurements taken from a prototype of an UWB based positioning system.

For all $i \in \{1, ..., n\}$ the error model is described by

$$\mathrm{p}\left(\eta_i | \boldsymbol{x}_i, \boldsymbol{p}_{a_i}, a_i\right) = \begin{cases} \frac{1}{c_0 + c_1 r_i} e^{\frac{-\eta_i}{c_0 + c_1 r_i}} & \eta_i \geq 0 \\ 0 & \eta_i < 0, \end{cases} \tag{6.2}$$

with r_i as the true range, $\eta_i = \hat{r}_i - r_i$ as the error of the true and measured range and $c_0, c_1 \in \mathbb{R}$.

To counter multipath effects within the channel, the UWB beacons feature a parameter to tune the *delay spread*. This quantity describes the mean time delay between reflection and direct signal. Because different settings for the

Node (j)	1	2	3	4	5	6	7	8
$p_{j,1}$	-3.415	-7.018	-4.229	-0.704	-0.607	-6.272	-7.090	-0.684
$p_{j,2}$	3.680	0.530	-2.027	3.472	-0.003	2.769	-1.472	-3.430
$p_{j,3}$	0.730	0.736	0.736	4.071	0.732	4.056	4.065	4.051

Table 6.7.: UWB beacon positions within the real life scenario

delay spread induce different probability density functions, c_0 and c_1 are chosen accordingly. A delay spread of $d_s = 30ns$ is modelled by the parameters $c_0 = 0.119$ and $c_1 = 0.0138$. This parameterization marks the worst case scenario. The best case scenario with a delay spread of $d_s = 5ns$ is modelled by $c_0 = 0.064$ and $c_1 = 0.0026$. The resulting probability densities for both scenarios are depicted in Fig. 6.7.

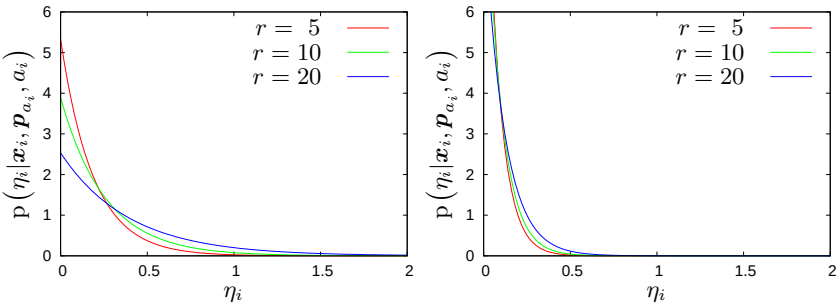

Figure 6.7.: Range dependend error for $d_s = 30ns$ (left) and $d_s = 5ns$ (right)

Note, at the time of formation of this thesis no real UWB beacons where available. The reference navigation system was therefore used to feed a simulator in real time that created range measurements following the error model from Eq. (6.2) and provided the measurements via the original interface.

Range bias and variance compensation

In order to deal with the assumption normally distributed and bias free range errors from Eq. (3.3) the range measurements were preprocessed. As Eq. (6.2) describes an exponential distributed random variable η_i the mean and covariance for the error can directly be quoted with

$$\mathrm{E}[\eta_i] = c_0 + c_1 r_i \quad \text{and} \quad \mathrm{Var}[\eta_i] = (c_0 + c_1 r_i)^2. \tag{6.3}$$

Unfortunately, these quantities as well as the whole probability density function for the range errors are dependent on the unknown range. To this end, the measurements $\hat{r}_i = r_i + \eta_i$ are used to build an unbiased estimator for the ranges such that $\mathrm{E}[\bar{r}] = r$ and an unbiased estimator for the variance such that $\mathrm{E}\left[\bar{\sigma}^2\right] = \mathrm{E}\left[(\bar{r} - r)^2\right]$.

The ansatz

$$\mathrm{E}[\bar{r}] = \mathrm{E}[d_1\hat{r} + d_0] = r \tag{6.4}$$

is used for the range estimator $\bar{r}$ and

$$\mathrm{E}\left[\bar{\sigma}^2\right] = \mathrm{E}\left[g_2\hat{r}^2 + g_1\hat{r} + g_0\right] = \mathrm{E}\left[(\bar{r} - r)^2\right] \tag{6.5}$$

is used for the variance $\bar{\sigma}^2$. The unknown parameters $d_1, d_0, g_2, g_1, g_0 \in \mathbb{R}$ are calculated by expanding Eq. (6.4), Eq. (6.5) and substituting Eq. (6.3). Equating the coefficients finally delivers the parameters.

The final estimators become

$$\bar{r} = \frac{1}{1 + c_1}\hat{r} - \frac{c_0}{1 + c_1} \tag{6.6}$$

and

$$\bar{\sigma}^2 = \frac{c_1^2}{(1 + 2c_1 + 2c_1^2)(1 + c_1)^2}\hat{r}^2 + \frac{c_0 c_1}{(1 + c_1)^3(1 + 2c_1 + 2c_1^2)}\hat{r} + \frac{c_0^2}{(1 + c_1)^3}. \tag{6.7}$$

For the following tests, the estimated ranges and variances were obtained by preprocessing the range measurements using Eq. (6.6) and Eq. (6.7).

6.2.2. UWB indoor localization

The rover was manually steered multiple times within the plane test area while gathering range measurements. Predefined by the test setup, all generated trajectories were planar. As a consequence the beacon positions could not be uniquely determined by the range measurements.

Exemplary, Fig. A.15 shows an obtained result when defining the coordinate system as in Sec. 3.1.2. Although the trajectory can be identified to be almost planar, some of the beacon positions are estimated to be above and some to be below the plane of movement. This is due to that any mirroring of the beacon positions on that plane would lead to the same range measurements. Because the coordinate system is defined by the beacon positions the whole scenario even seems to be rotated. Fig. A.16 still shows a good fit of the estimated on the measured ranges. However, as the exact beacon positions are unknown the plane of movement is not properly defined.

To obtain a unique result for the real life tests, further information is incorporated. To this end, the freely selectable coordinate system is defined in a different

manner than before. It is assumed that the trajectory is known to be planar and that the normal vector to plane of movement is defining the z-axis. Second, all beacon positions are assumed to have positive z-coordinates $p_{3,j} \geq 0$ for all $j \in \{1, ..., k\}$. Fig. 6.8 illustrates the new definition of the coordinate system.

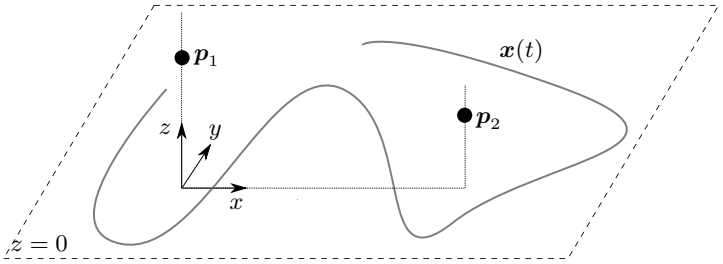

Figure 6.8.: Definition of the coordinate system for planar trajectories

By definition, the z-axis is orthogonal to the plane defined by the test area and the test area itself has the coordinate $z = 0$. As in Sec. 3.1.2 the x and y axis is defined through the positions of the beacons 1 and 2 with $\boldsymbol{p}_1 \triangleq [0 \quad 0 \quad p_{3,1}]^{\mathrm{T}}$ and $\boldsymbol{p}_2 \triangleq [0 \quad p_{2,2} \quad p_{3,2}]^{\mathrm{T}}$.

The two realized trajectories for $d_s = 30ns$ and $d_s = 5ns$ are plotted in Fig. 6.9 and Fig. 6.10 respectively.

The data rate for the range measures for both cases was $100ms$. The robustness of the method was investigated by reducing the number of measurements. To this end, a fixed number of range measurements is randomly extracted from the set of all measurements.

Worst case ($d_s = 30ns$)

For the case $d_s = 30ns$ the plot in Fig. 6.11 shows the RMS error of the estimated trajectory when using the kernel based approach with respect to the number of samples used. The plot also depicts the results obtained by the ML approach with data alignment for $\Delta T \in \{3, 5, 10\}$. The parameters of the kernel function were intuitively chosen to be $a \triangleq 45$, $b \triangleq 0.3$, $\beta \triangleq 10$ and $m \triangleq 25$.

The obtained results show the superiority in quality of the kernel based approach. For any choice of ΔT and any number of samples the RMS error is significantly smaller when using the kernel based approach. The ML approach with alignment shows a bias-variance tradeoff induced by the alignment parameter. For a sparse amount of range data the estimator using a great ΔT produces a better estimate than the estimator using a small ΔT. However, when increasing the number of samples a decreased ΔT produces better results

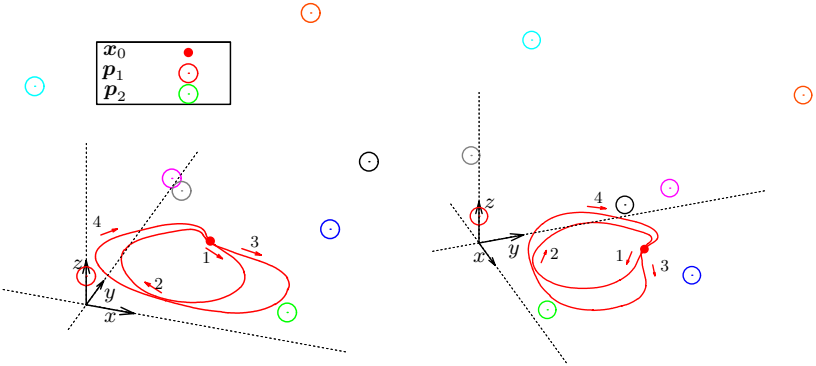

Figure 6.9.: Sampled test trajectory taken from 2 different viewpoints

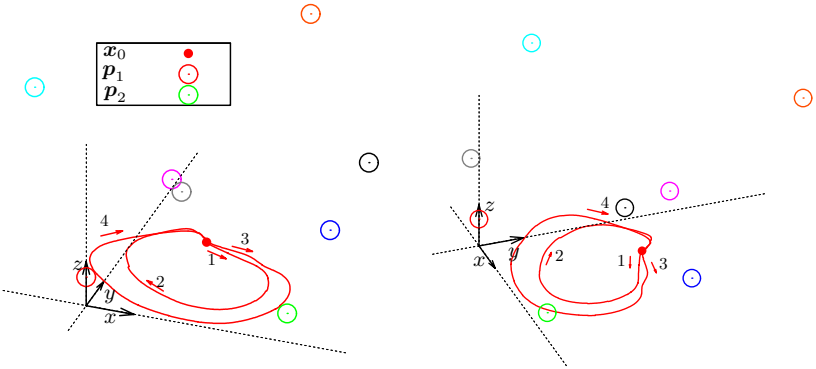

Figure 6.10.: Sampled test trajectory taken from 2 different viewpoints

as the estimator becomes more and more unbiased. For about $n > 800$ the RMS error obtained by $\Delta T = 3$ almost coincides with the RMS error of the kernel based approach. Note however, for $\Delta T = 3$ the search space becomes very big and the optimization process very slow. In this example, the optimization time for the kernel based approach was about 0.0001 of the time required for the ML approach with $\Delta T = 3$.

Best case ($d_s = 5ns$)

Fig. 6.12 shows the same investigation for the second scenario with a delay spread of $d_s = 5ns$. In general, the same effects can be observed. Note however

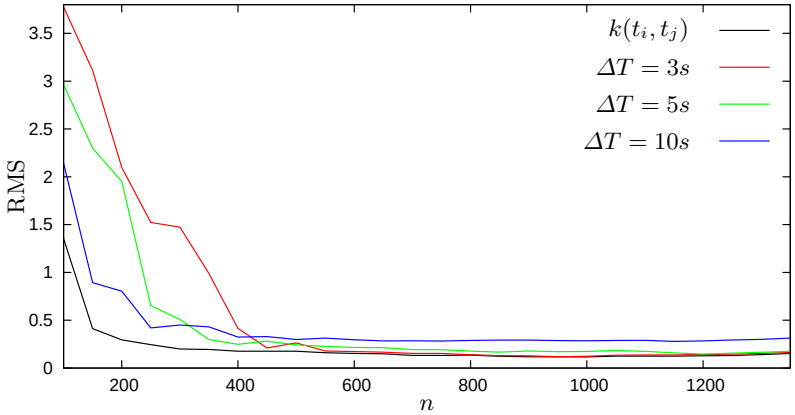

Figure 6.11.: RMS error versus the number of measurements for $d_s = 30ns$

the significantly lower RMS error.

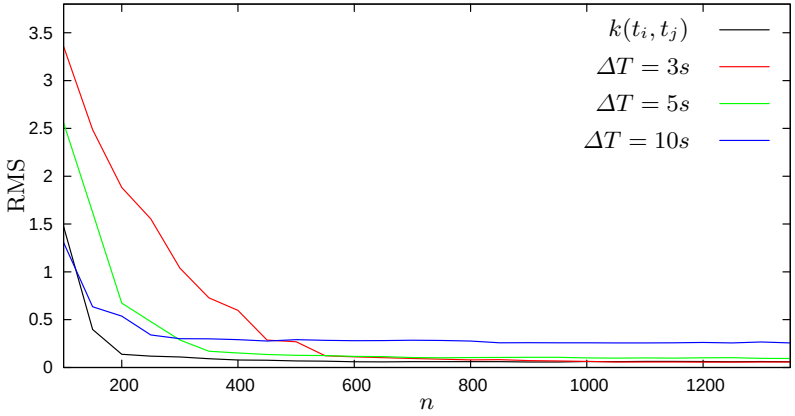

Figure 6.12.: RMS error versus the number of measurements for $d_s = 5ns$

The case $n = 400$ was plotted in three dimensions for two approaches. Fig. A.17 shows the result using the kernel featured approach and Fig. A.19 using the ML approach with the alignment parameter $\Delta T = 10$.

Real time estimation

To investigate the capability for online estimation the two step approach as depicted in Fig. 5.5 was applied for both cases. About 100 initial measurements

of the data proved to be enough to produce an accurate offline initialization. For $d_s = 5ns$, Fig. A.21 shows the result in three dimensions. The plot shows the true trajectory, the true beacon positions, the initial estimate for trajectory $\tilde{\boldsymbol{x}}$ and beacons $\tilde{\boldsymbol{p}}_j$ with $1 \leq j \leq k$ as well as the immediate solutions $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{p}}_j$ obtained by the EKF. Note the improvement over time for the trajectory and the beacon estimates.

7. Summary and Outlook

A novel approach for resolving ill-posed localization and mapping problems was proposed in this thesis.The developed method allows to localize an object with only unsynchronized range measurements to beacons of unknown positions for disposal. Having a generic design, the method can be applied to a wide range of related problems.

Generally, one has to bear in mind that an ill-posed problem can never be properly solved. This thesis therefore followed a typical principle that can be quoted as follows:

> "If there is a problem you can't solve, then there is an easier problem you can solve: find it." - *George Polya*

Of course, such a procedure can only be crowned with success if the new problem formulation is well-posed and the solution expresses the desired result.

Summary

For localization problems this was achieved by the extension of the available information by two additional assumptions.

1. Every movement is originated from a driving force.
2. The driving force is limited in power.

These two constraints are modelled as linear and time invariant dynamic systems driven by a band limited random force process. The solutions to the resulting stochastic differential equations were shown to result in a stationary stochastic process that represents the velocities and a non-stationary stochastic process that represents the realizable trajectories. This interpretation of the realizable trajectories and velocities serves as prior information.

To obtain a manageable problem formulation, the solution space for the velocities is embedded within a function space and the trajectories are obtained by integration. To this end, all relevant realizations of the stochastic process representing the velocities were shown to be contained in a reproducing kernel Hilbert space when the associated kernel function equals the autocorrelation function of that stochastic process. Furthermore, the norm within that Hilbert space was shown to be a measure for the posterior probability of trajectory occurrence.

Exploiting the former stochastic interpretation, an a posteriori estimator for the localization problem was formulated in terms of kernel regression. Moreover, it was shown that consistency of the problem formulation can be assured when introducing an upper bound on the norm of the reproducing kernel Hilbert space. For an increasing number of measurements, the minimum of the regularized risk functional therefore converges to a unique solution.

Usually, kernel regression is performed for a fixed amount of training data and minimization is performed within a batch process. For the localization problem at hand an adaptive implementation of kernel regression was developed. Initialized by a solution obtained by a Levenberg-Marquardt minimization, an extended Kalman filter was derived that minimizes the regularized risk functional continuously in time. The proposed method therefore allows simultaneous self localization and mapping of the static beacons in realtime after a short initialization period.

Especially for a sparse amount of data the final tests show the benefits of the derived model and the proposed kernel design. While an alignment of observation data in time domain was also shown to be a valid method for resolving ill-posed localization problems, synthetic and real life tests showed the superiority of the proposed method with respect to accuracy and efficiency.

Outlook

The main contribution of this thesis is to open up kernel regression techniques for real time localization problems and to embed them into the classical localization framework. For a lot of ill-posed localization problems unavailable control information can be compensated by the proposed dynamic model. For example, simultaneous localization and mapping based on a monocular camera may greatly gain from a kernel based approach. Also, well-posed problems could benefit by the additional amount of information and even available control information might be featured by the model. In case of available measurements from an inertial measuring unit, the data can be included as additional observations.

The three dimensional acceleration model presented in this thesis is rather simple as it implicitly assumes a holomorphic platform that can be accelerated any time in any direction. However, more sophisticated models for steering may also be applicable and improve the performance of the regularization approach. Imagine a stochastic process applied on a steering wheel of a car for instance

Although the proposed optimization methods proved to be sufficient for the investigated case of range based localization and mapping, more elaborate methods might be required for other applications. Furthermore, the initialization step might be superfluous when using alternative approaches for online optimization like a particle filter.

A. Appendix

A.1. Calculations

The following calulations make use of the *error function* which is defined as

$$\operatorname{erf}(t) \triangleq \frac{2}{\sqrt{\pi}} \int_0^t e^{-\tau^2} \mathrm{d}\tau.$$

Consequently, the derivative of the error function is

$$\frac{d}{dt}\operatorname{erf}(t) = \frac{\sqrt{\pi}}{2} e^{-t^2},$$

and the antiderivative of the error function

$$\frac{d}{dt}\left(t \cdot \operatorname{erf}(t) + \frac{1}{\sqrt{\pi}} e^{-t^2}\right) = \operatorname{erf}(t).$$

A.1.1. Transfer function $w(t)$

By definition

$$w(t) \triangleq g_{\mathcal{L}} * h = \int_{-\infty}^{\infty} \frac{ab}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2 b^2}{2}} e^{-\frac{\beta\tau}{m}} \frac{H(\tau)}{m} \mathrm{d}\tau =$$

$$= \frac{ab}{m\sqrt{2\pi}} \int_0^{\infty} e^{-\frac{(t-\tau)^2 b^2 m + 2\beta\tau}{2m}} \mathrm{d}\tau.$$

Since

$$(t-\tau)^2 b^2 m + 2\beta\tau = mb^2\tau^2 - 2(mb^2 t - \beta)\tau + mb^2 t^2 =$$

$$= \left(\tau b\sqrt{m} - \frac{mb^2 t - \beta}{b\sqrt{m}}\right)^2 - \frac{(mb^2 t - \beta)^2}{b^2 m} + mb^2 t^2 =$$

$$= \left(\tau b\sqrt{m} - \frac{mb^2 t - \beta}{b\sqrt{m}}\right)^2 + \frac{\beta(2mb^2 t - \beta)}{b^2 m},$$

the integral simplifies to

$$\frac{ab}{m\sqrt{2\pi}}C_1 \int_0^\infty e^{-\frac{(\tau b\sqrt{m}-\frac{mb^2t-\beta}{b\sqrt{m}})^2}{2m}} \mathrm{d}\tau =$$
$$= \frac{ab}{m\sqrt{2\pi}}C_1 \int_0^\infty e^{-\left(\tau\frac{b}{\sqrt{2}}-\frac{mb^2t-\beta}{\sqrt{2}bm}\right)^2} \mathrm{d}\tau.$$

With the definitions

$$C_1 \triangleq e^{-\frac{\beta(2b^2tm-\beta)}{2b^2m^2}} \quad \text{and} \quad C_2 \triangleq -\frac{mb^2t-\beta}{\sqrt{2}bm},$$

a substitution of the integrant

$$\zeta \triangleq \tau\frac{b}{\sqrt{2}} + C_2, \quad \frac{d\zeta}{d\tau} = \frac{b}{\sqrt{2}},$$

and a split up of the integral finally yields

$$w(t) = \frac{a}{m\sqrt{\pi}}C_1 \int_{C_2}^\infty e^{-\zeta^2}\mathrm{d}\zeta = \frac{a}{m\sqrt{\pi}}C_1 \left(\int_0^\infty e^{-\zeta^2}\mathrm{d}\zeta - \int_0^{C_2} e^{-\zeta^2}\mathrm{d}\zeta\right) =$$
$$= \frac{a}{2m}C_1(1-\operatorname{erf}(C_2)) = \frac{a}{2m}C_1(1+\operatorname{erf}(-C_2)) =$$
$$= \frac{a}{2m}e^{-\frac{\beta(2b^2tm-\beta)}{2m^2b^2}} \left(1+\operatorname{erf}\left(\frac{mb^2t-\beta}{\sqrt{2}mb}\right)\right).$$

A.1.2. Covariance function of $x^p(t,\varsigma)$

$$\begin{aligned}
\boldsymbol{r}_{x^p}(t_1,t_2) &\triangleq \mathrm{E}[x(t_1,\varsigma)x(t_2,\varsigma)] = \\
&= \mathrm{E}\left[\int_0^{t_1} v^p(\tau_1,\varsigma)\mathrm{d}\tau_1 \int_0^{t_2} v^p(\tau_2,\varsigma)\mathrm{d}\tau_2\right] = \\
&= \mathrm{E}\left[\int_0^{t_1}\int_{-\infty}^{\infty} n(\tau_1-s,\varsigma)w(s)\mathrm{d}s\mathrm{d}\tau_1 \int_0^{t_2}\int_{-\infty}^{\infty} n(\tau_2-\hat{s},\varsigma)w(\hat{s})\mathrm{d}\hat{s}\mathrm{d}\tau_2\right] = \\
&= \mathrm{E}\left[\int_0^{t_1}\int_0^{t_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(s)w(\hat{s})n(\tau_1-s,\varsigma)n(\tau_2-\hat{s},\varsigma)\mathrm{d}\hat{s}\mathrm{d}s\mathrm{d}\tau_2\mathrm{d}\tau_1\right] = \\
&= \int_0^{t_1}\int_0^{t_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(s)w(\hat{s})\mathrm{E}[n(\tau_1-s,\varsigma)n(\tau_2-\hat{s},\varsigma)]\,\mathrm{d}\hat{s}\mathrm{d}s\mathrm{d}\tau_2\mathrm{d}\tau_1 = \\
&= \int_0^{t_1}\int_0^{t_2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(s)w(\hat{s})\delta(\tau_1-s-\tau_2+\hat{s})\mathrm{d}\hat{s}\mathrm{d}s\mathrm{d}\tau_2\mathrm{d}\tau_1 = \\
&= \int_0^{t_1}\int_0^{t_2}\int_{-\infty}^{\infty} w(s)w(\tau_2-\tau_1+s)\mathrm{d}s\mathrm{d}\tau_1\mathrm{d}\tau_2 = \int_0^{t_1}\int_0^{t_2} \boldsymbol{r}_{v^p}(\tau_2,\tau_1)\mathrm{d}\tau_2\mathrm{d}\tau_1.
\end{aligned}$$

The change of the order of integration here is grounded on *Fubini's theorem* [71, p. 422]. As $\boldsymbol{r}_{v^p}(\tau_2,\tau_1) = \boldsymbol{r}_{v^p}(\tau_2-\tau_1) = \boldsymbol{r}_{v^p}(\tau_1-\tau_2)$ is invariant to a shift in time

$$\int_0^{t_2} \boldsymbol{r}_{v^p}(\tau_2,\tau_1)\mathrm{d}\tau_2 = \int_{-\tau_1}^{t_2-\tau_1} \boldsymbol{r}_{v^p}(\tau)\mathrm{d}\tau = \int_0^{\tau_1} \boldsymbol{r}_{v^p}(\tau)\mathrm{d}\tau + \int_0^{t_2-\tau_1} \boldsymbol{r}_{v^p}(\tau)\mathrm{d}\tau.$$

With the additional definition

$$F(t) \triangleq \int_0^t \boldsymbol{r}_{v^p}(\tau)\mathrm{d}\tau,$$

the integral reduces to

$$\int_0^{t_2} \boldsymbol{r}_{v^p}(\tau_2,\tau_1)\mathrm{d}\tau_2 = F(\tau_1) + F(t_2-\tau_1).$$

Substitution of the second integral and a change in variables delivers

$$\begin{aligned}\boldsymbol{r}_{x^p}(t_1,t_2) &= \int_0^{t_1} F(\tau_1)\mathrm{d}\tau_1 + \int_0^{t_1} F(t_2-\tau_1)\mathrm{d}\tau_1 = \\ &= \int_0^{t_1} F(\tau_1)\mathrm{d}\tau_1 - \int_{t_2}^{t_2-t_1} F(\tau)\mathrm{d}\tau = \\ &= \int_0^{t_1} F(\tau_1)\mathrm{d}\tau_1 + \int_0^{t_2} F(\tau_1)\mathrm{d}\tau_1 - \int_0^{t_2-t_1} F(\tau)\mathrm{d}\tau = \\ &= R(t_1) + R(t_2) - R(t_2-t_1)\end{aligned}$$

with

$$R(t) \triangleq \int_0^t F(\tau_1)\mathrm{d}\tau_1 = \int_0^t \int_0^{\tau_1} \boldsymbol{r}_{v^p}(\tau_2)\mathrm{d}\tau_2\mathrm{d}\tau_1 = R(-t).$$

A.1.3. Proof of $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{\Sigma}_v^{-1}\boldsymbol{v} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_x^{-1}\boldsymbol{x}$

Define

$$\boldsymbol{\beta} \triangleq \boldsymbol{\Sigma}_x^{-1}\boldsymbol{x}$$

and let

$$\alpha(t) \triangleq \sum_{i=1}^{n} \alpha_i \delta(t-t_i), \quad \beta(t) \triangleq \sum_{i=1}^{n} \beta_i \delta(t-t_i)$$

using the dirac delta distribution. Then the negative exponent of the probability density becomes

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_x^{-1}\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta} = \int_{-\infty}^{\infty} x(\tau_1)\beta(\tau_1)\mathrm{d}\tau_1. \tag{A.1}$$

Since

$$v(t_i) = \int_{-\infty}^{\infty} k_{v^p}(t_i,\tau_2)\alpha(\tau_2)\mathrm{d}\tau_2$$

and

$$x(\tau_1) = \int_0^{\tau_1} v(\gamma_1)\mathrm{d}\gamma_1 = \int_{-\infty}^{\infty}\int_0^{\tau_1} k_{v^p}(\gamma_1,\tau_2)\alpha(\tau_2)\mathrm{d}\gamma_1\mathrm{d}\tau_2$$

equation (A.1) can be rewritten

$$\int_{-\infty}^{\infty} x(\tau_1)\beta(\tau_1)\mathrm{d}\tau_1 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\tau_1} k_{v^p}(\gamma_1,\tau_2)\alpha(\tau_2)\beta(\tau_1)\mathrm{d}\gamma_1\mathrm{d}\tau_2\mathrm{d}\tau_1. \tag{A.2}$$

As already stated, the covariance function within the process $x^p(t,\varsigma)$ is known to be

$$k_{x^p}(t_1,t_2) = \int_{0}^{t_1}\int_{0}^{t_2} k_{v^p}(\gamma,\xi)\mathrm{d}\gamma\mathrm{d}\xi. \tag{A.3}$$

Differentiating equation (A.3) delivers

$$\frac{d}{dt_2}k_{x^p}(t_1,t_2)\Big|_{t_2=\tau_2} = \int_{0}^{t_1} k_{v^p}(\gamma,\tau_2)\mathrm{d}\gamma.$$

Since

$$v(\tau_1) = \frac{d}{dt}x(t)\Big|_{t=\tau_1} = \frac{d}{dt}\int_{-\infty}^{\infty} k_{x^p}(t,\gamma_1)\beta(\tau_1)\mathrm{d}\gamma_1 =$$

$$= \int_{-\infty}^{\infty}\int_{0}^{\tau_1} k_{v^p}(\gamma_1,\tau_2)\beta(\tau_1)\mathrm{d}\gamma_1\mathrm{d}\tau_2,$$

the insertion into (A.2) finally yields

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_x^{-1}\boldsymbol{x} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{\Sigma}_v^{-1}\boldsymbol{v} = \alpha^{\mathrm{T}}\boldsymbol{K}\alpha.$$

A.1.4. Covariance matrix of the state vector

In this section the single elements of the covariance matrix

$$\boldsymbol{W}_{n+1} \triangleq \mathrm{E}\left[\begin{bmatrix} x_{n+1}^- - x_{n+1} \\ \boldsymbol{v}_{n+1}^- - \boldsymbol{v}_{n+1} \end{bmatrix}\begin{bmatrix} x_{n+1}^- - x_{n+1} \\ \boldsymbol{v}_{n+1}^- - \boldsymbol{v}_{n+1} \end{bmatrix}^{\mathrm{T}}\right]$$

for the state vector $\begin{bmatrix} x_{n+1}^- & \boldsymbol{v}_{n+1}^- \end{bmatrix}^{\mathrm{T}}$ are calculated.

Prerequisites

For simplicity let

$$\boldsymbol{K} \triangleq \mathrm{E}\left[\boldsymbol{v}_n^2\right] = \boldsymbol{K}(\boldsymbol{t}_n, \boldsymbol{t}_n) \quad k_0 \triangleq \mathrm{E}\left[v_{n+1}^2\right] = k(0,0)$$

and

$$\boldsymbol{k}_{n+1} \triangleq \mathrm{E}[v_{n+1}\boldsymbol{v}_n] = \boldsymbol{K}(\boldsymbol{t}_n, t_{n+1}).$$

The covariance between velocity process and the trajectory process is obtained by

$$\begin{aligned}\mathrm{E}[(x_{n+1} - x_n)v_i] &= \mathrm{E}\left[\int_{t_n}^{t_{n+1}} v(\tau)\mathrm{d}\tau v_i\right] = \int_{t_n}^{t_{n+1}} k(\tau, t_i)\mathrm{d}\tau \overset{\xi \triangleq \tau - t_i}{=} \\ &= \int_{t_n - t_i}^{t_{n+1} - t_i} k(\xi + t_i, t_i)\mathrm{d}\xi = \int_{t_n - t_i}^{t_{n+1} - t_i} k(\xi, 0)\mathrm{d}\xi = \\ &= \kappa(t_{n+1} - t_i) - \kappa(t_n - t_i)\end{aligned}$$

with κ from Eq. (4.38) and therefore

$$\mathrm{E}[(x_{n+1} - x_n)\boldsymbol{v}_n] = \boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n.$$

with Eq. (5.8). Consequently

$$\mathrm{E}[(x_{n+1} - x_n)v_{n+1}] = \kappa(0) - \kappa(t_n - t_{n+1}) = \kappa(t_{n+1} - t_n).$$

The covariance for the position process is given by Eq. (4.32)

$$\mathrm{E}[x_i x_j] = R(t_i) + R(t_j) - R(t_i - t_j),$$

with $R(t)$ from Eq. (4.33).

The expectation of the squared difference of the actual and the new position becomes

$$\begin{aligned}\mathrm{E}\left[(x_{n+1} - x_n)^2\right] =& \mathrm{E}\left[x_{n+1}^2\right] + \mathrm{E}\left[x_n^2\right] - 2\mathrm{E}[x_{n+1}x_n] = \\ =& 2R(t_{n+1} - t_n).\end{aligned}$$

Covariances

With these quantities, the elements of the covariance matrix can be quoted. The variance of the estimation error for the actual velocity becomes

$$\begin{aligned}[\boldsymbol{W}_{n+1}]_{n+2,n+2} &= \mathrm{E}\left[(v_{n+1}^- - v_{n+1})^2\right] = \mathrm{E}\left[(v_{n+1} - \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{v}_n)^2\right] = \\ &= k_0 - \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{k}_{n+1},\end{aligned} \tag{A.4}$$

which is identical to Eq. (5.5). By construction of the dynamic model, $v_i^- - v_i = 0$ for all $1 \leq i \leq n$ and therefore

$$[\boldsymbol{W}_{n+1}]_{i+1,j+1} = \mathrm{E}\left[(v_i^- - v_i)(v_j^- - v_j)\right] = 0, \quad \forall 1 \leq j \leq n+1.$$

For the same reason this also holds for the covariance of the estimation error for the past velocities and actual position,

$$[\boldsymbol{W}_{n+1}]_{1,j+1} = \mathrm{E}\left[(x_{n+1}^- - x_{n+1})(v_j^- - v_j)\right] = 0, \quad \forall 1 \leq j \leq n.$$

However, the covariance of the estimation error for the actual velocity and actual position yields

$$\begin{aligned}[\boldsymbol{W}_{n+1}]_{1,n+2} = \mathrm{E}\left[(x_{n+1}^- - x_{n+1})(v_{n+1}^- - v_{n+1})\right] = \mathrm{E}\left[x_{n+1}^- v_{n+1}^-\right] - \\ - \mathrm{E}\left[x_{n+1}^- v_{n+1}\right] - \mathrm{E}\left[x_{n+1} v_{n+1}^-\right] + \mathrm{E}[x_{n+1} v_{n+1}]\,.\end{aligned} \tag{A.5}$$

The single terms result in

$$\begin{aligned}\mathrm{E}\left[x_{n+1}^- v_{n+1}^-\right] &= \mathrm{E}\left[\left(x_n + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\boldsymbol{v}_n\right)\boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{v}_n\right] = \\ &= \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\mathrm{E}[\boldsymbol{v}_n x_n] + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\mathrm{E}\left[\boldsymbol{v}_n \boldsymbol{v}_n^{\mathrm{T}}\right]\boldsymbol{K}^{-1}\boldsymbol{k}_{n+1} = \\ &= \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\mathrm{E}[\boldsymbol{v}_n x_n] + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\boldsymbol{k}_{n+1}\end{aligned}$$

$$\begin{aligned}\mathrm{E}\left[x_{n+1}^- v_{n+1}\right] &= \mathrm{E}\left[\left(x_n + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\boldsymbol{v}_n\right) v_{n+1}\right] = \\ &= \mathrm{E}[x_n v_{n+1}] + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\mathrm{E}[\boldsymbol{v}_n v_n + 1] = \\ &= \mathrm{E}[x_n v_{n+1}] + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\boldsymbol{k}_{n+1}\end{aligned}$$

$$\mathrm{E}\left[x_{n+1} v_{n+1}^-\right] = \mathrm{E}\left[x_{n+1}\boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{v}_n\right] = \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\mathrm{E}[x_{n+1}\boldsymbol{v}_n]\,.$$

The covariance is given by

$$\begin{aligned}[\boldsymbol{W}_{n+1}]_{1,n+2} &= \mathrm{E}[(x_{n+1} - x_n)v_{n+1}] - \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}\mathrm{E}[\boldsymbol{v}_n(x_{n+1} - x_n)] = \\ &= \kappa(t_{n+1} - t_n) - \boldsymbol{k}_{n+1}^{\mathrm{T}}\boldsymbol{K}^{-1}(\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n).\end{aligned}$$

The variance of the estimation error for the actual position

$$\begin{aligned}\mathrm{E}\left[(x_{n+1}^- - x_{n+1})^2\right] &= \mathrm{E}\left[(x_{n+1} - x_n - (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}_n^{-1}\boldsymbol{v}_n)^2\right] = \\ &= \mathrm{E}\left[(x_{n+1} - x_n)^2\right] - 2(\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\mathrm{E}[(x_{n+1} - x_n)\boldsymbol{v}_n)] + \\ &\quad + (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}\mathrm{E}\left[\boldsymbol{v}_n\boldsymbol{v}_n^{\mathrm{T}}\right]\boldsymbol{K}^{-1}(\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n) = \\ &= 2R(t_{n+1} - t_n) - (\boldsymbol{\kappa}_{n+1}^{\mathrm{T}} - \boldsymbol{\kappa}_n^{\mathrm{T}})\boldsymbol{K}^{-1}(\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n).\end{aligned} \tag{A.6}$$

The matrix $\boldsymbol{W}_{n+1}$ has only four non zero entries specified by Eq. (A.4), Eq. (A.5) and Eq. (A.6).

A.2. Plots

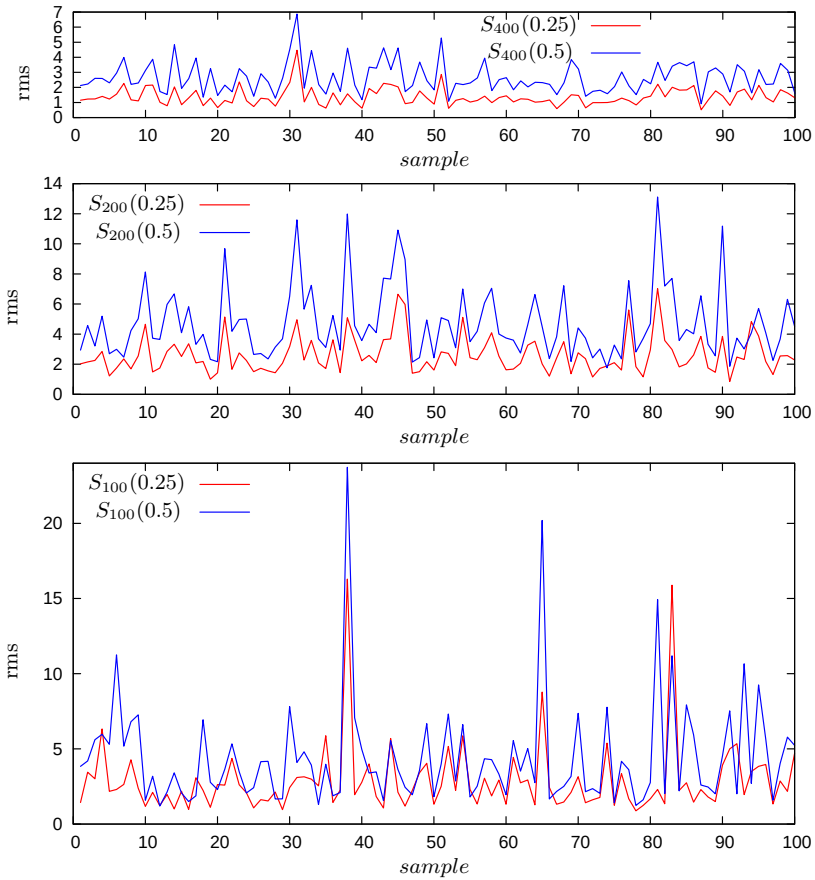

Figure A.1.: rms errors using k

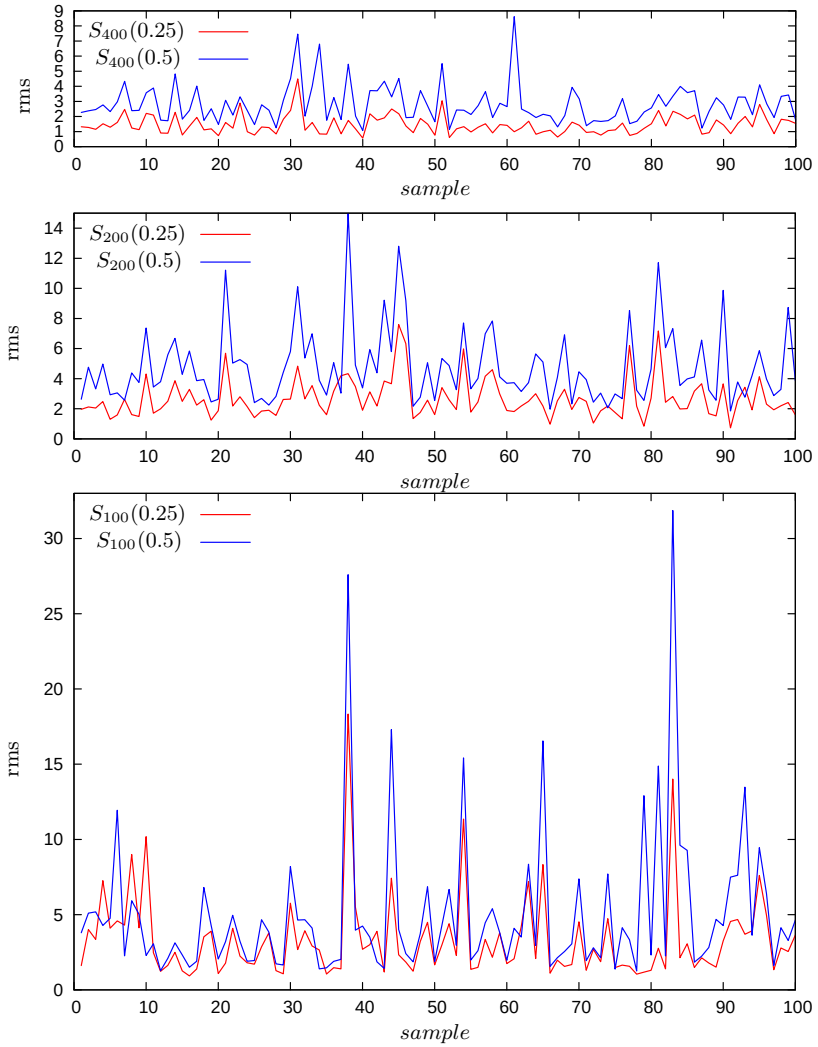

Figure A.2.: rms errors using k_{ou}

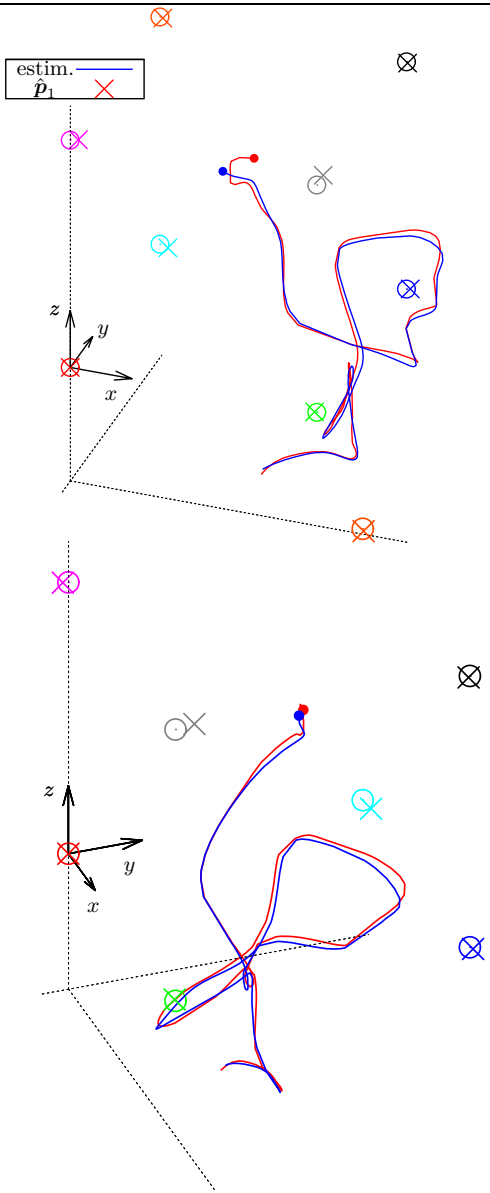

Figure A.3.: Trajectory 97 and estimation using $k(t_i, t_j)$ within $S_{200}(0.25)$

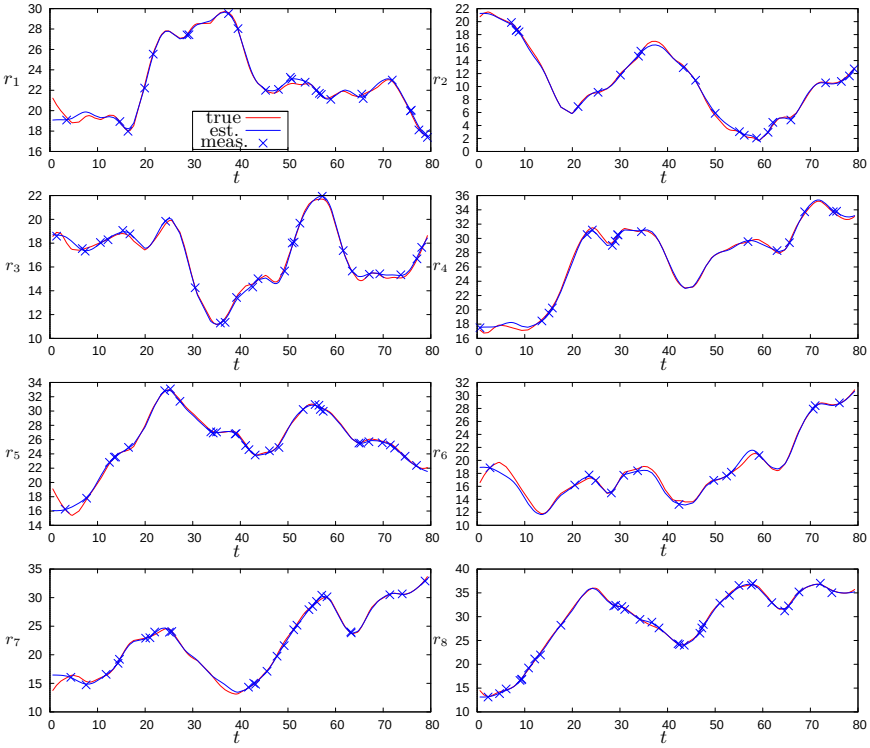

Figure A.4.: True, estim. and measured ranges for the scenario from Fig. A.3

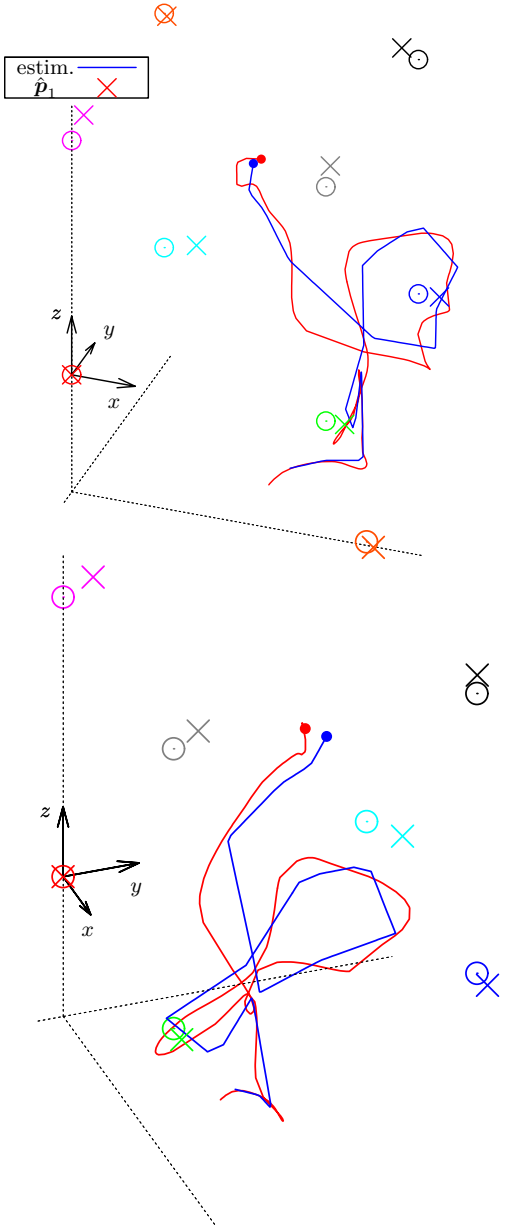

Figure A.5.: Trajectory 97 and estimation using an alignment $\Delta T \triangleq 5$ within $S_{200}(0.25)$

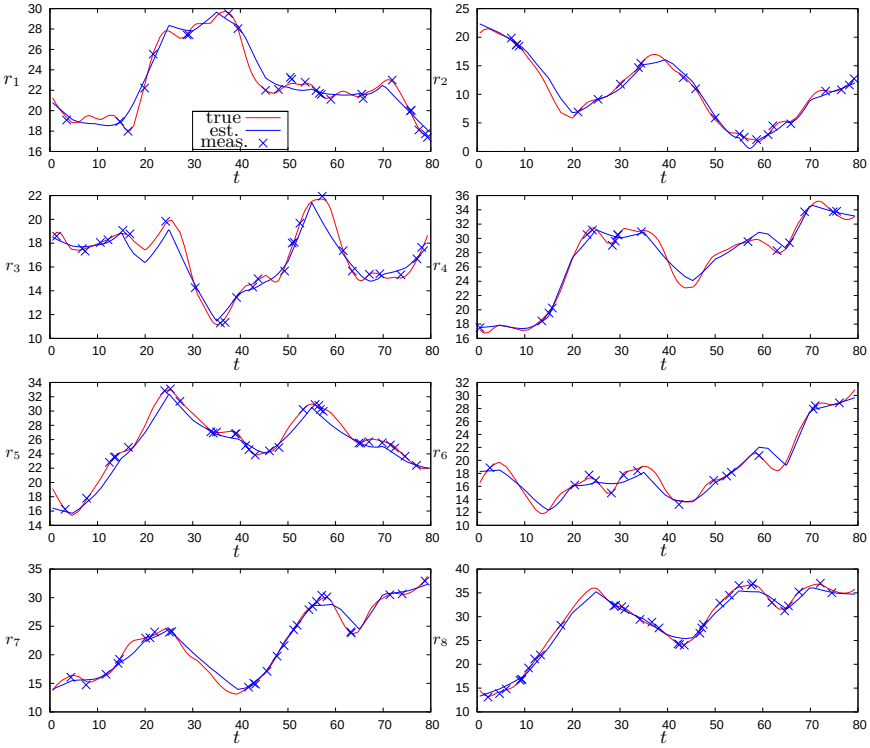

Figure A.6.: True, estim. and measured ranges for the scenario from Fig. A.5

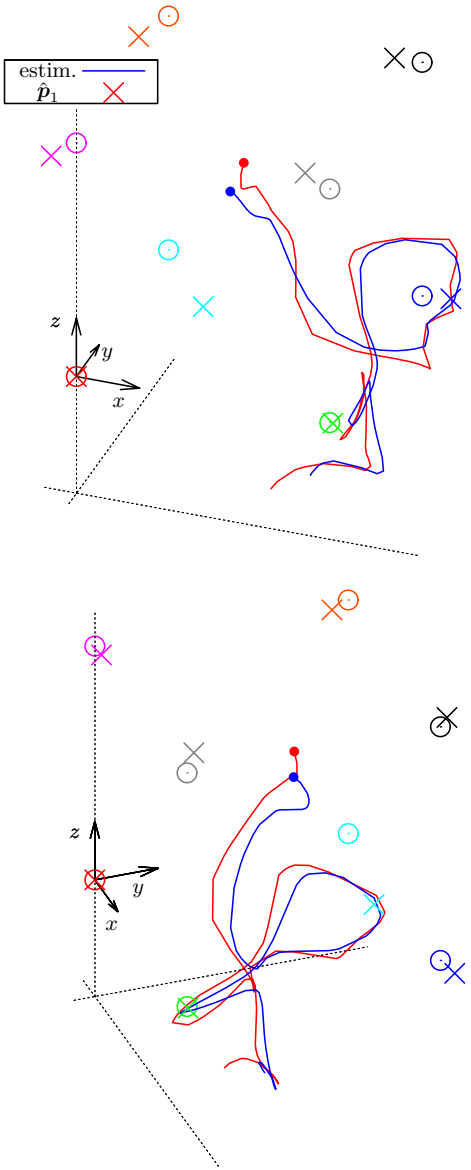

Figure A.7.: Trajectory 97 and estimation using $k(t_i, t_j)$ within $S_{100}(0.5)$

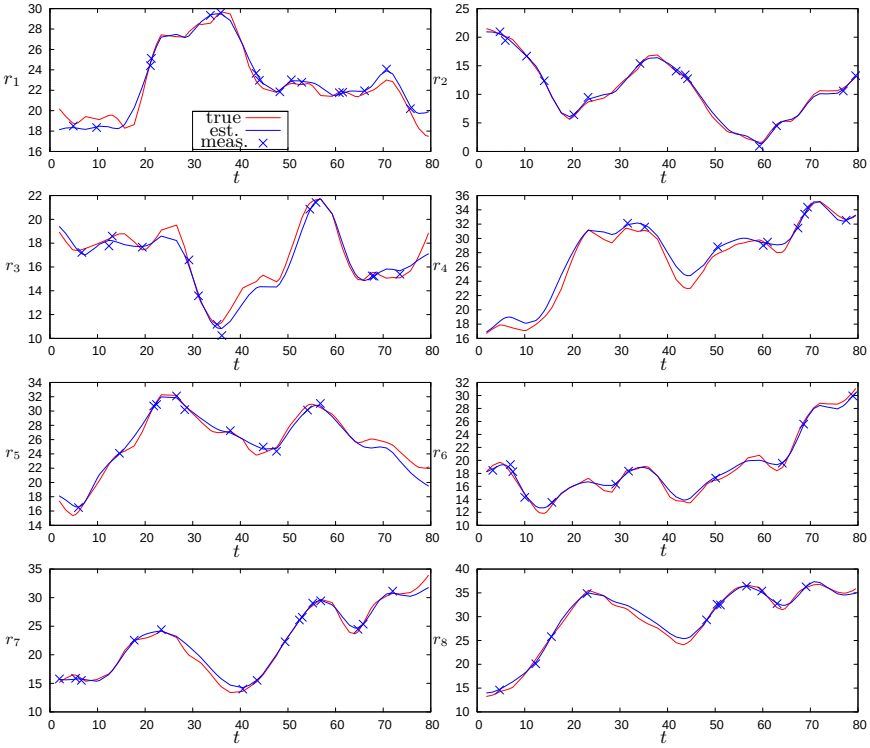

Figure A.8.: True, estim. and measured ranges for the scenario from Fig. A.7

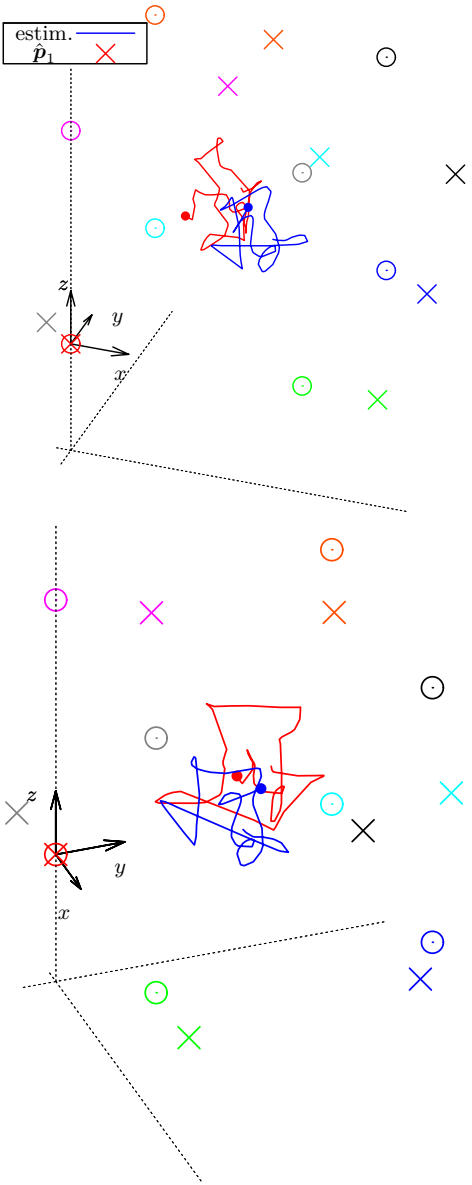

Figure A.9.: Trajectory 44 and estimation using $k(t_i, t_j)$ within $S_{100}(0.5)$

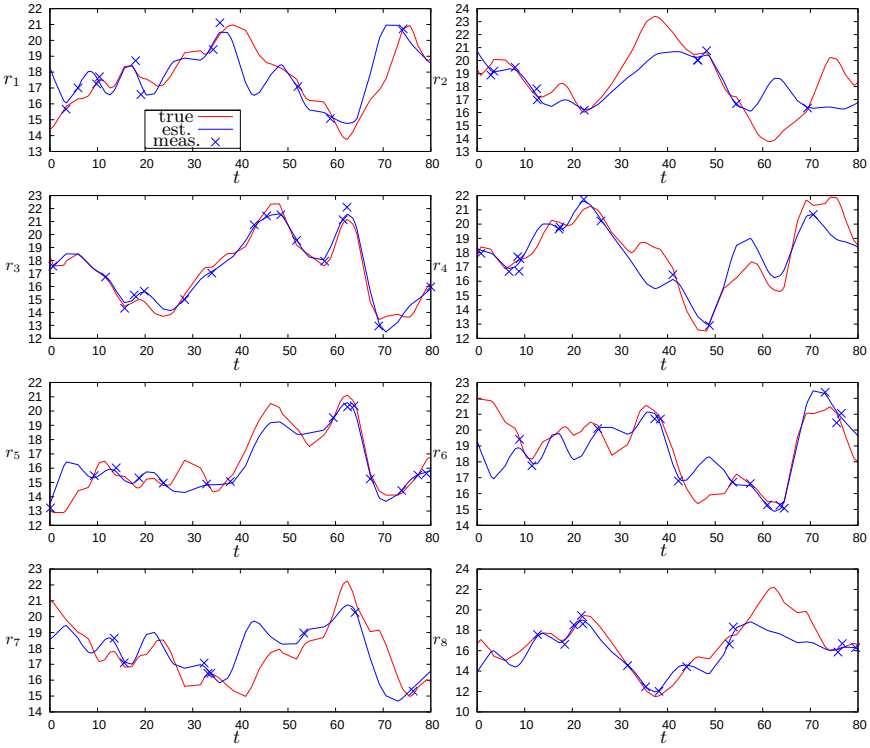

Figure A.10.: True, estim. and measured ranges for the scenario from Fig. A.9

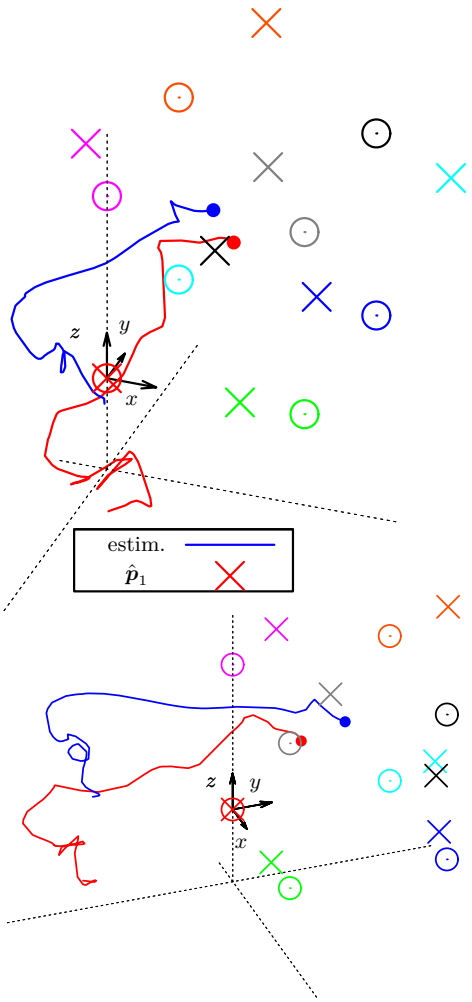

Figure A.11.: Trajectory 83 and estimation using $k(t_i, t_j)$ within $S_{100}(0.5)$

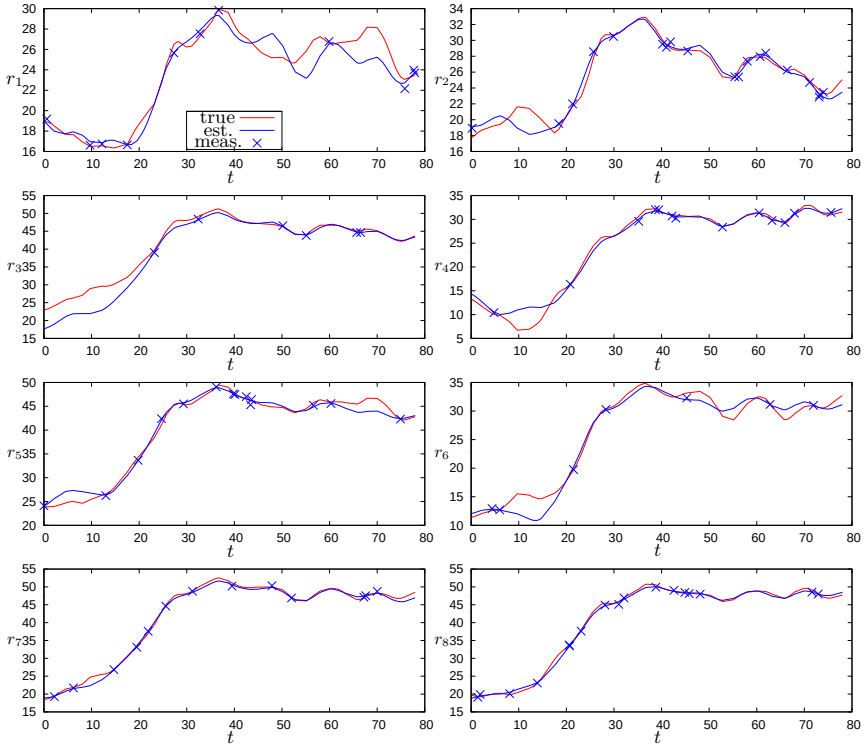

Figure A.12.: True, estim. and measured ranges for the scenario from Fig. A.11

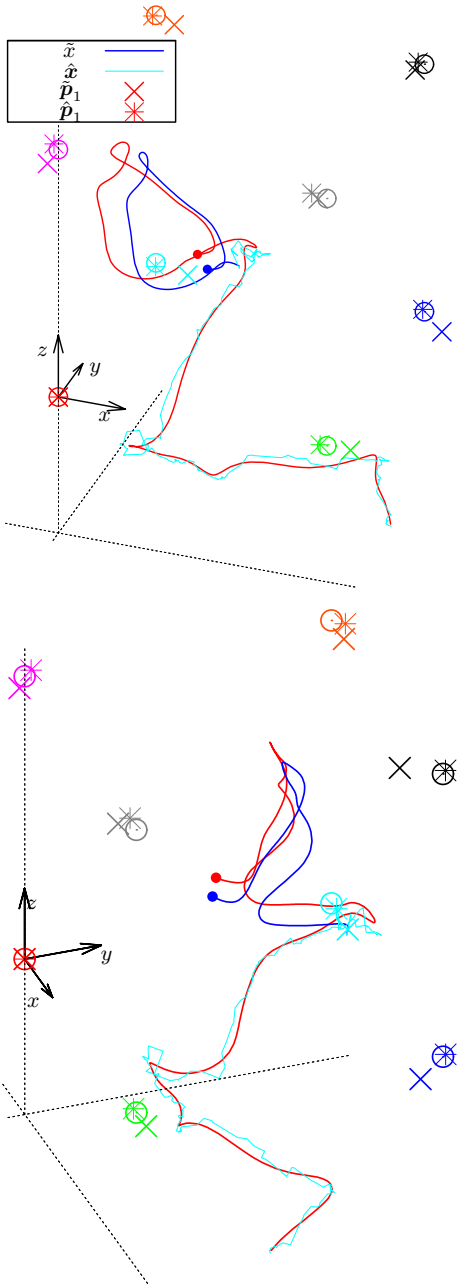

Figure A.13.: Immediate and initial estimate of the EKF for traj. 59 in $S_{400}(0.25)$

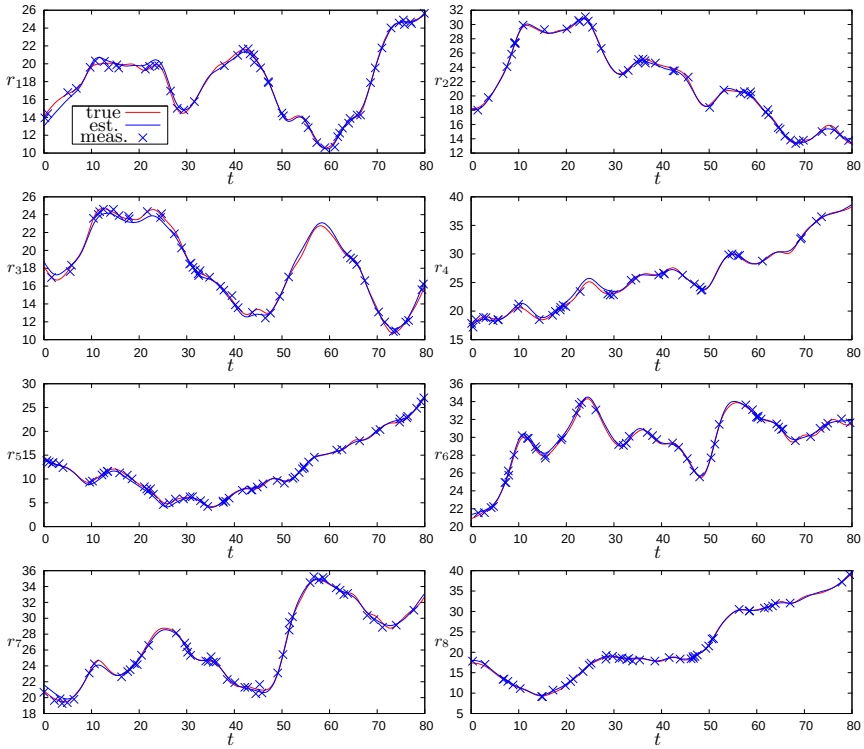

Figure A.14.: True, final estim. and measured ranges for the scenario from Fig. A.13

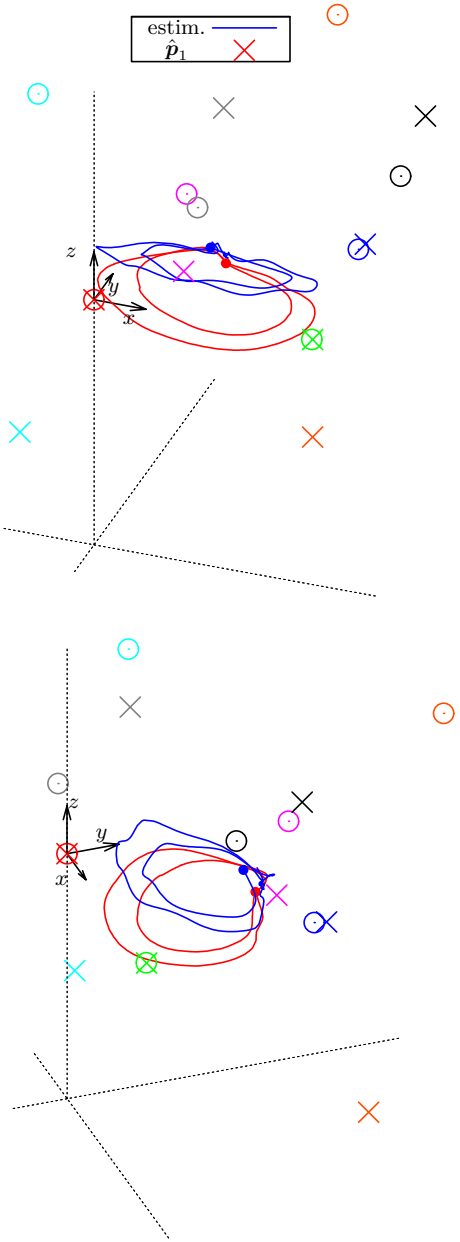

Figure A.15.: Real trajectory for $d_s = 5ns$ and estimation using $k(t_i, t_j)$

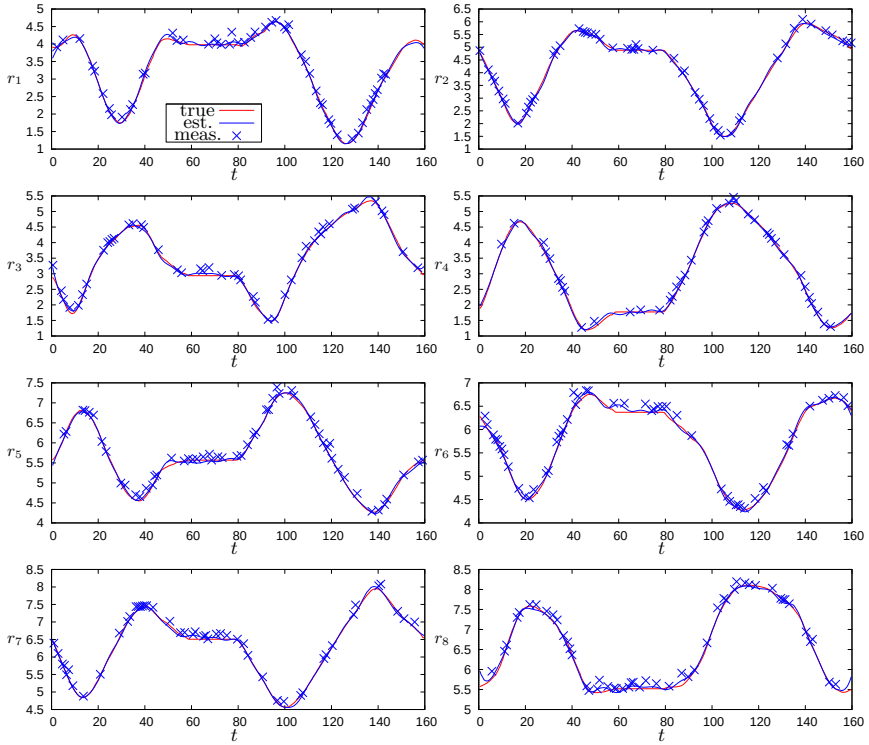

Figure A.16.: True, estim. and measured ranges for the scenario from Fig. A.15

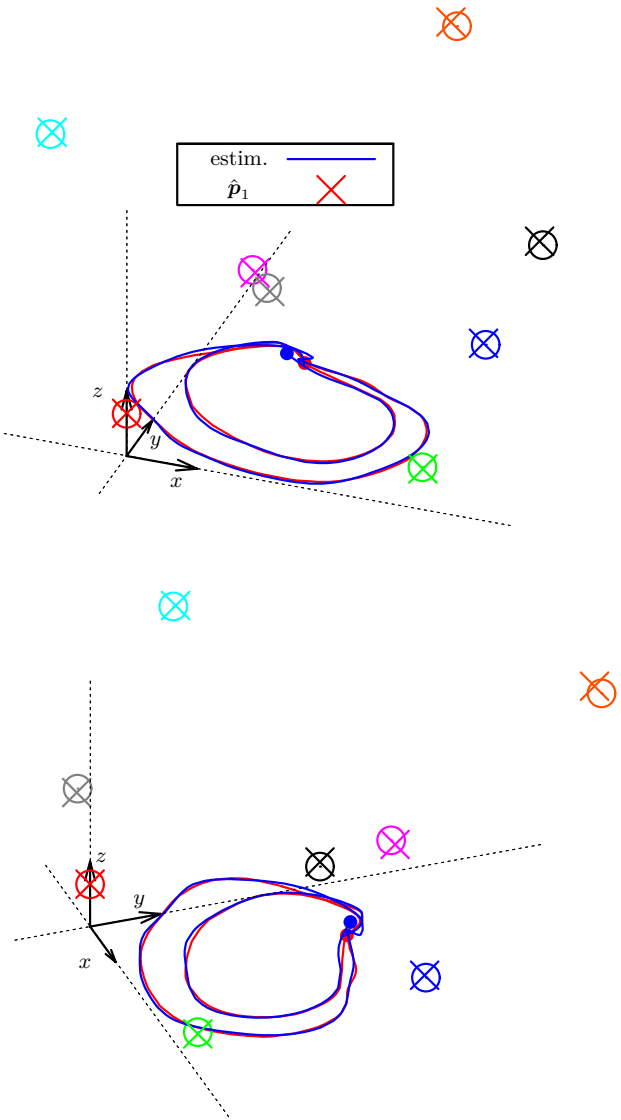

Figure A.17.: Real trajectory for $d_s = 5ns$ and estimation using $k(t_i, t_j)$

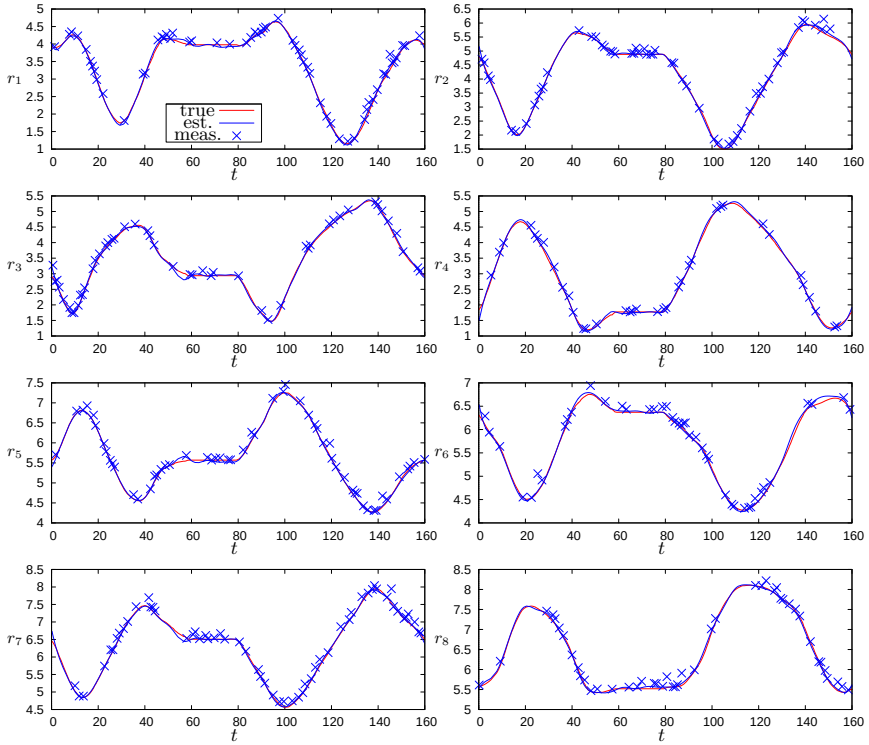

Figure A.18.: True, estim. and measured ranges for the scenario from Fig. A.17

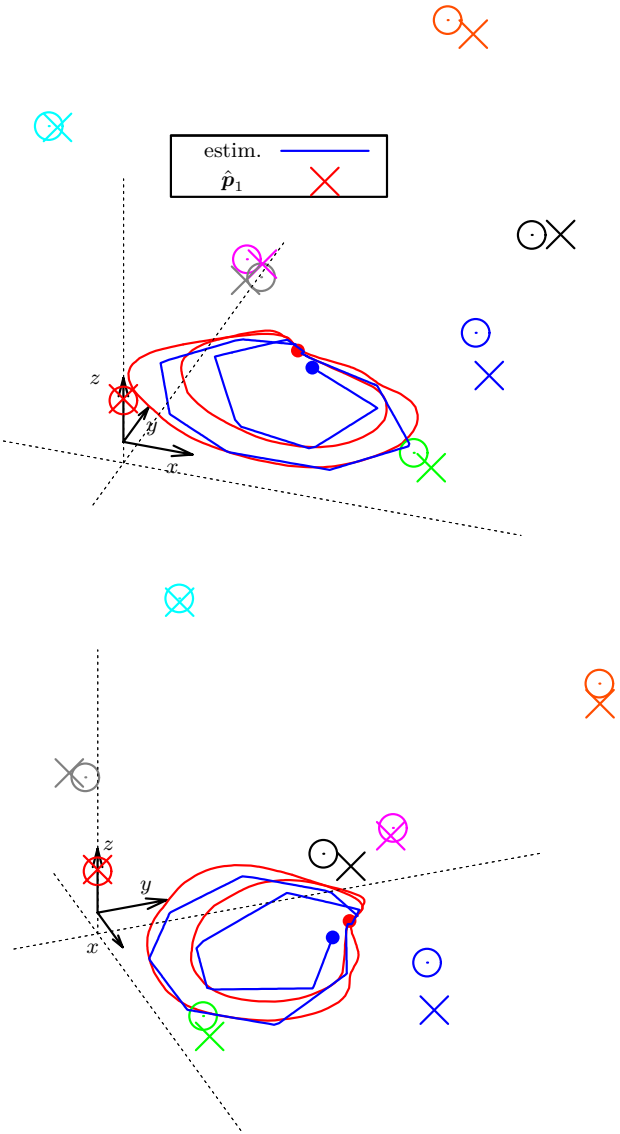

Figure A.19.: Real trajectory for $d_s = 5ns$ and estimation using $\Delta T = 10$

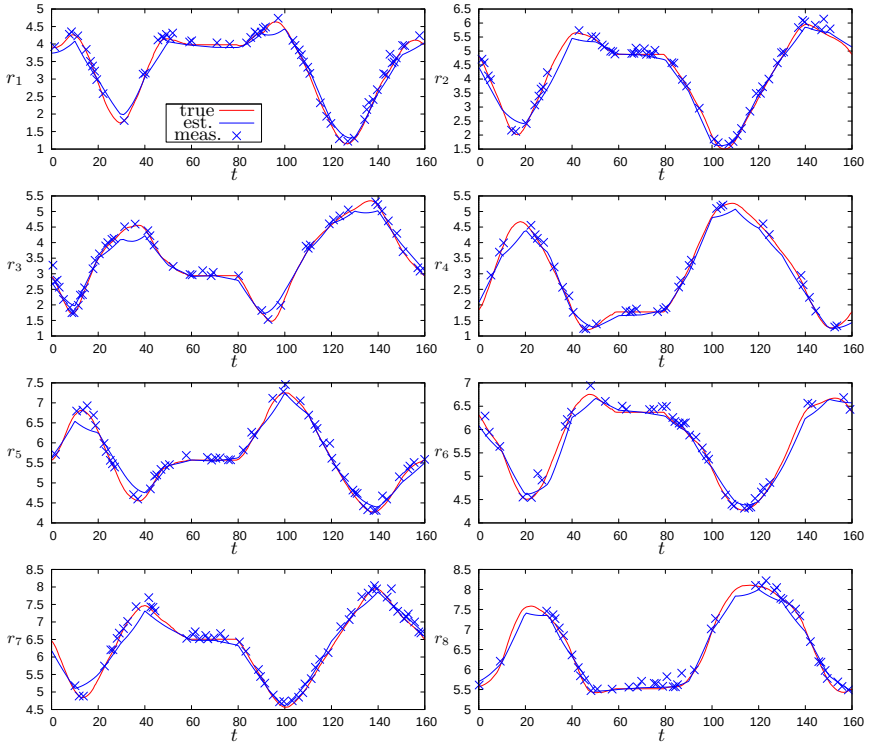

Figure A.20.: True, estim. and measured ranges for the scenario from Fig. A.19

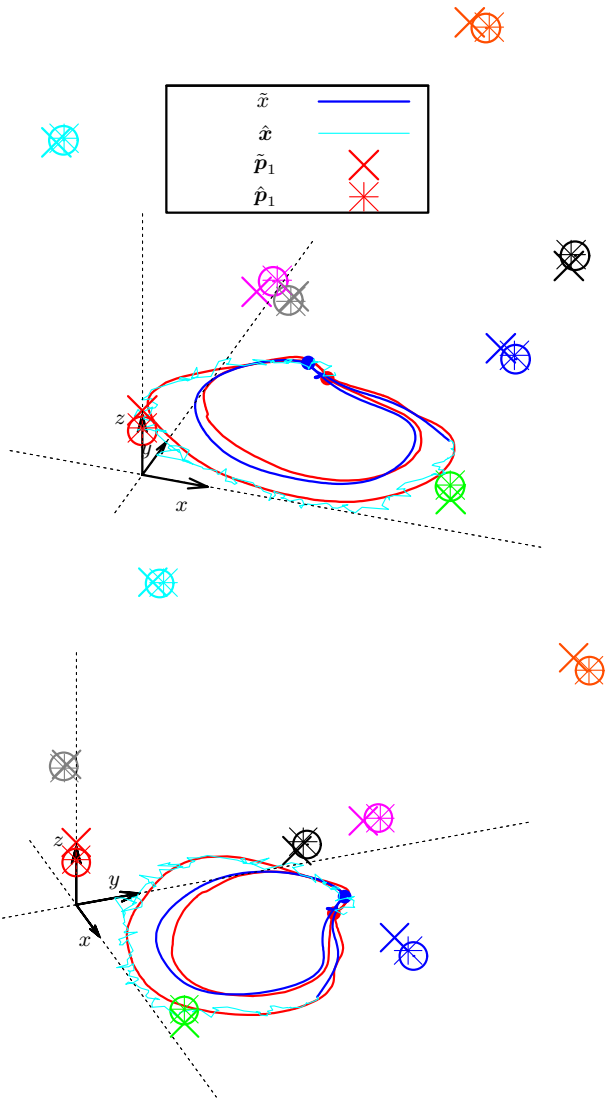

Figure A.21.: Immediate and initial estimate of the EKF for the test trajectory

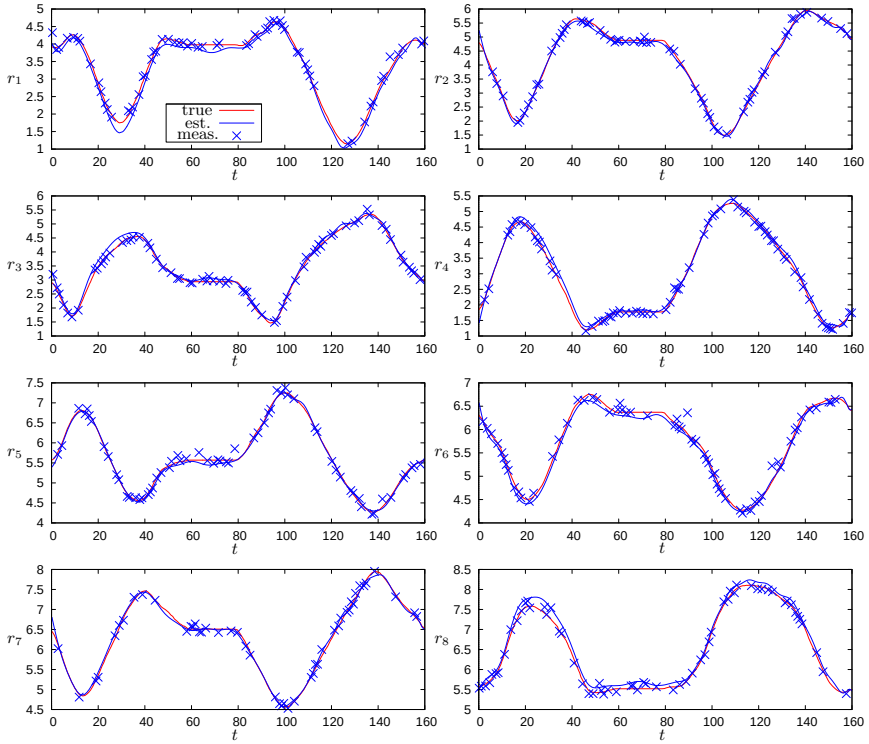

Figure A.22.: True, final estim. and measured ranges for Fig. A.21

Bibliography

[1] Bruce T. Clough. Metrics, schmetrics! how the heck do you determine a uav's autonomy anyway? *Proceedings of the Performance Metrics for Intelligent Systems Workshop*, 2002.

[2] Hui-Min Huang, Elena Messina, Robert Wade, Ralph English, and Brian Novak. Autonomy measures for robots. *Proceedings of IMECE: International Mechanical Engineering Congress.*, 2004.

[3] Hui-Min Huang, James Albus, and Elena Messina. Toward a generic model for autonomy levels for unmanned systems (alfus). *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2003.

[4] Charles Lesire and Catherine Tessier. A hybrid model for situation monitoring and conflict prediction in human supervised "autonomous" systems. Technical report, Association for the Advancement of Artificial Intelligence, 2006.

[5] Andreas Kugi. *Systemtheorie und Regelungstechnik*. Universität des Saarlandes, 2005.

[6] Milena Anguelova. Nonlinear observability and identifiability: General theory and a case study of a kinetic model. Master's thesis, Göteborg University, 2004.

[7] Krzysztof W. Kolodziej and Johan Hjelm. *Local Positioning Systems (LBS Applications and Services)*. CRC Press Taylor & Francis Group, 2006.

[8] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in building rf-based user location and tracking system. *Proceedings of IEEE Infocom*, 2000.

[9] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, and Dan S. Wallach. Robotics based location sensing using wireless ethernet. In *Proceedings of the eighth Annual International Conferecne on Mobile Computing and Networking (MOBICOM-02)*, pages 227–238. ACM Press, 2002.

[10] Moustafa A. Youssef, Ashok Agrawala, and A. Udaya Shankar. WLAN location determination via clustering and probability distributions. In *IEEE International Conferecne on Pervasive Computing and Communications (PerCom)*, 2003.

[11] *Fact sheet (Ubisense: Series 7000 Sensor).*

[12] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. *GPS (Theory and Practice).* Springer, 2001.

[13] Teemu Tonteri. A statistical modeling apporach to location estimation. Master's thesis, University of Helsinki, 2001.

[14] P. Krishnan, A. S. Krishnakumar, Wen-Hua Ju, Colin Mallows, and Sachin Ganu. A system for LEASE. *IEEE InfoCom*, 2004.

[15] Erwin Stenzel. Indoor localization. Technical report, EADS Innovation Works SC/IRT/LG-M, 2005.

[16] Paul M. Maxim, Suranga Hettiarachchi, William M. Spears, Diana F. Spears, Jerry Hamann, Thomas Kunkel, and Caleb Speiser. Trilateration localization for multi-robot teams. *Proceedings of the Sixth International Conference on Informatics in Control, Automation and Robotics, ICINCO*, 2008.

[17] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.

[18] Ilja N. Bronstein, Konstantin A. Semendjajew, Musiol Gerhard, and Mühlig Heiner, editors. *Taschenbuch der Mathematik.* Nauka Verlag, Moskau, 1979.

[19] Ake Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[20] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[21] John C. Nash. *Compact Numerical Methods for Computers.* Adam Hilger, 1979.

[22] Andreas Savvides, Heemin Park, and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. *First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA,*, pages 112–121, 2002.

[23] Harald Cramér. *Mathematical Methods of Statistics.* Princeton University Press, 1946.

[24] Calyampudi Rao. *Breakthroughs in Statistics Vol. 1*, volume 37, chapter Information and the accuracy attainable in the estimation of statistical parameters, pages 235–247. Springer, 1945.

[25] Athanasios Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill Companies, 1991.

[26] Alfred Leick. *GPS Satellite Surveying.* John Wiley & Sons, inc., 1994.

[27] http://www.rootsweb.ancestry.com/ mosmd/logln.htm.

[28] David H. Titterton and John L. Weston. *Strapdown Inertial Navigation Technology.* Peter Peregrinus Ltd., on behalf the Institution of Electrical Engineers, 1997.

[29] Din 9300-2: Luft- und raumfahrt; begriffe, größen und formelzeichen der flugmechanik; bewegungen des luftfahrzeugs und der atmosphäre gegenüber der erde; iso 1151-2:1985 (stand 1987) modifiziert.

[30] Oliver J. Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, 1997.

[31] Georges Sagnac. The demonstration of the luminiferous aether by an interferometer in uniform rotation. *Comptes Rendus de l'Academie des Sciences*, 157:708–710, 1913.

[32] Georges Sagnac. On the proof of the reality of the luminiferous aether by the experiment with a rotating interferometer. *Comptes Rendus de l'Academie des Sciences*, 157:1410–1413, 1913.

[33] Gaspard-Gustave de Coriolis. Mémoire sur le principe des forces vives dans les mouvements relatifs des machines. *Journal de lÉcole polytechnique*, v. XIII, cahier XXI (1832):142–154, 1835.

[34] Gaspard-Gustave de Coriolis. Mémoire sur les équations du mouvement relatif des systèmes de corps. *Journal de lÉcole polytechnique*, 24° cahier, XV, cahier XXIV:142–154, 1835.

[35] Neil Barbour and George Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*, 1:332–339, 2001.

[36] Giles C. Lee and Gori Marco, editors. *Adaptive Processing of Sequences and Data Structures.* Springer-Verlag, 1998.

[37] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.

[38] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[39] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter*. Artech House London, 2004.

[40] Gilbert Strang and Kai Borre. *Linear Algebra, Geodesy, and GPS*. Wellesly-Cambridge Press, 1997.

[41] Simon Julier, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45:477–482, 2000.

[42] Eric A. Wan and Rudolph Van Der Merwe. The Unscented Kalman filter for Nonlinear Estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000.*, pages 153–158, 2000.

[43] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[44] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F on Radar and Signal Processing*, 140:107–113, 1993.

[45] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. *IEEE International Conference on Robotics and Automation*, 4:850– 850, 1987.

[46] Michael Montemerlo and Sebastian Thrun. *FastSlam*. Springer, 2007.

[47] Randall Smith and Peter Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *International Journal of Robotics Research*, pages 56–68, 1986.

[48] Philippe Moutarlier and Raja Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. *Lecture Notes in Control and Information Sciences*, 139:327 – 346, 1989.

[49] John Leonhard and Hugh F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.

[50] Gamini M. W. M. Dissanayake, Paul M. Newman, Steve Clark, Hugh F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on robotics and automation*, 17:229 – 241, 2001.

[51] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[52] Holger Täubig and Christof Schröter. Simultaneuos Localization and Mapping (SLAM) mit hierarchischen Partikelfiltern. *3rd Workshop on Self-Organization of adaptiVE Behavior (SOAVE)*, pages 157–166, 2004.

[53] Andrew J. Davison, Ian Reid, Nicholas Molton, and Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.

[54] Yi Shang, Wheeler Ruml, and Ying Zhang. Localization from mere Connectivity. *Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking & Computing*, pages 201 – 212, 2003.

[55] Xiang Ji and Hongyuan Zha. Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling. In *IEEE Conference on Computer Communications*, 2004.

[56] David Moore, John Leonhard, Daniela Rus, and Seth Teller. Robust Distributed Network Localization with Noisy Range Measurements. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50 – 61, 2004.

[57] Alexander T. Ihler, John W. Fisher, Alan Willsky, and Randolph Moses. Nonparametric Belief Propagation for Self-Calibration in Sensor Networks. *IEEE Selected Areas in Communications*, pages 809– 819, 2005.

[58] Robert Grabowski and Pradeep Khosla. Localization techniques for a team of small robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1067 – 1072, 2001.

[59] Aram Galstyan, Bhaskar Krishnamachari, Kristina Lerman, and Sundeep Pattem. Distributed Online Localization in Sensor Networks Using a Moving Target. *Proceedings of the 3rd international symposium on Information Processing in Sensor Networks*, pages 61 – 70, 2004.

[60] Volkan Cevher and James H. McClellan. Sensor Array Calibration via Tracking with the Extended Kalman Filter. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5:2817–2820, 2001.

[61] Paul Newman and John J. Leonard. Pure Range-Only Sub-Sea SLAM. *Conference on Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International*, 2:1921– 1926, 2003.

[62] Edwin Olson, John Leonard, and Seth Teller. Robust Range-Only Beacon Localization. *IEEE Journal of Oceanic Engineering*, 31:949–958, 2006.

[63] Christopher J. Taylor, Ali Rahimi, Jonathan Bachrach, Howard Shrobe, and Anthony Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 27–33, 2006.

[64] Hari Balakrishnan, Roshan Baliga, Dorothy Curtis, Michel Goraczko, Allen Miu, Bodhi Priyantha, Adam Smith, Ken Steele, Seth Teller, and Kevin Wang. Lessons from developing and deploying the cricket indoor location system. Technical report, MIT, 2003.

[65] Robert A. Heinlein. *The Moon Is a Harsh Mistress*. G. P. Putnam's Sons, 1966.

[66] Andrei Nikolaevich Tikhonov and Vasilii Y. Arsenin. *Solutions of ill-posed problems*. Wiley, 1977.

[67] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[68] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[69] John Shawe-Taylor and Nello Cristianini. *Margin Distribution and Soft Margin*, pages 349–358. MIT Press, 2000.

[70] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

[71] Dirk Werner. *Funktionalanalysis*. Springer, 1997.

[72] Natesh S. Pillai, Qiang Wu, Feng Liang, Sayan Mukherjee, and Robert L Wolpert. Characterizing the Function Space for Bayesian Kernel Models. *Jounal of Machine Learning Research*, 8:1769–1797, 2007.

[73] George Kimeldorf and Grace Wahba. Some Results on Thebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[74] William Rowan Hamilton. On a general method in dynamics. *Philosophical Transactions of the Royal Society*, II:247308, 1834.

[75] Raj Kumar Pathria. *Statistical Mechanics*. Pergamon Press, 1977.

[76] Bernt Oksendal. *Stochastic Differential Equations*. Springer, 2007.

[77] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[78] Grace Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[79] Milan N. Lukić and Jay H. Beder. Stochastic Processes With Sample Paths In Reproducing Kernel Hilbert Spaces. *Transactions of the American Mathematical Society*, 353:3945–3969, 2001.

[80] Stephen J Gustafson and Israel Michael Sigal. *Mathematical Concepts of Quantum Mechanics*. Springer, 2003.

[81] Harry Nyquist. Certain topics in telegraph transmission theory. *Proceedings of the IEEE (Reprint as classic paper)*, 90:280–305, 2002.

[82] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37:10–21, 1949.

[83] Walter Rudin. *Functional Analysis*. McGraw-Hill, 1991.

[84] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (The Art of Scientific Computing)*. Cambridge University Press, 1995.

[85] C. Tim Kelley. *Iterative Methods for Optimization*. SIAM (Society for Industrial and Aplied Mathematics), 1999.

[86] Luke Tierney and Joseph B. Kadane. Accurate Approximations for Posterior Moments and Marginal Densities. *Journal of the American Statistical Association*, 81:82–86, 1986.

[87] Jamil Dakhlallah, Sébastien Glaser, Said Mammar, and Yazid Sebsadji. Tire-Road Forces Estimation Using Extended Kalman Filter and Sideslip Angle Evaluation. *American Control Conference*, 2008.

[88] Sonia Maria Martinho Marques. *Small Satellites Attitude Determination Methods*. PhD thesis, Universidade de Lisboa, 2000.