

Development and Flight Testing of an Automatic Flight Control System with Dynamic Control Objective Prioritization

Erik Anders Karlsson

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: apl. Prof. Dr.-Ing. Christian W. M. Breitsamter

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Florian Holzapfel
2. Prof. Dr.-Ing. Stephan Myschik

Die Dissertation wurde am 13.05.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 22.10.2024 angenommen.

Abstract

The thesis presents the development of a modular automatic flight control system for control of the kinematic flight path and the aerodynamic speed, implemented and integrated as part of a modular flight guidance and control system, along with flight test results from initial controller deployment and subsequent test campaigns on the DA42 OE-FSD flying testbed.

A system development process tailored to the model-based flight control law development approach is defined, considering applicable guidance material for the development of civil aircraft and systems in general, and automatic flight control systems in particular. System functions and requirements are developed based on a principal automatic flight control system architecture with generic architecture elements. A modular functional architecture supports the definition of generic as well as application-specific functions and corresponding system requirements. The automatic flight control system may thus be tailored for different application scenarios for manned and unmanned aircraft of different sizes, envelopes, and regulatory environments.

The controller architecture is subsequently developed. The flight path controller is based on a reference model-based dynamic inversion of the translational equations of motions, directly commanding the required specific forces to the inner loop and autothrust controllers in order to achieve the desired vertical and lateral flight path curvatures, as well as acceleration along the flight path. The architecture allows for the full utilization of the available inner loop performance for path control. Arising control objective conflicts during high-bandwidth maneuvering are managed by an integrated energy rate and force prioritization architecture, with reference model cross-feeds that enable active prioritization between vertical flight path or acceleration as well as vertical or lateral flight path curvature. The airspeed envelope and the energy integrity of the aircraft are protected as the energy flow rate is automatically distributed in favor of the acceleration at the envelope edges. The controller is implemented as a modular architecture with distributed mode logic that allows configuration and adaptation of functionalities for new application platforms without extensive retesting. A Functional Hazard Assessment is performed based on the functional design, and different system architecture strategies and their implication on the development assurance levels of the control system elements are discussed.

Further, a configurable mode control and monitoring interface, consisting of a mode control panel and a compact mode control and monitoring display developed in conjunction with the controller, is presented. The mode control and monitoring design matured through iterations of executable desktop applications interfacing with the model-in-the-loop simulation environment, greatly supporting control law prototyping and early validation of system operation. Display software and mode control logic were validated in flight tests and remote-control operation.

The implementation of the design model for the flight control algorithms is based on a model-based software development framework, with which the tailored system development process is aligned. The automatic flight control system design model is integrated with the interfacing flight control algorithm modules and extensively tested in a set of model-in-the-loop environments before automatic source code generation, hardware-software integration, and hardware-in-the-loop testing. Controller functional and performance requirements are verified in linear as well as nonlinear assessment with automated requirements evaluation over the envelope for applicable aircraft configurations.

The thesis concludes with results from flight test activities performed in 2016. The controller deployment on the DA42 OE-FSD testbed is described, with initial flight tests showing good overall controller performance without the need for any major modifications. Additional flight test campaigns on the DA42 OE-FSD throughout 2016 are summarized.

Zusammenfassung

In dieser Arbeit wird die Entwicklung eines modularen, automatischen Flugbahnregelungssystems, implementiert und integriert als Teil eines modularen Flugregelungs- und führungssystems, vorgestellt, sowie Flugtestergebnisse aus dem ersten Einsatz des Systems und nachfolgenden Testkampagnen auf dem fliegenden Erprobungsträger DA42 OE-FSD.

Es wird ein der modellbasierten Entwicklungsansatz der Flugregelungsalgorithmen angepasstes Systementwicklungsprozess definiert, der zutreffenden Standards und Richtlinien für die Entwicklung von sicherheitskritischen Systemen in der Luftfahrt im Allgemeinen und automatischen Flugsteuerungssystemen im Besonderen berücksichtigt. Systemfunktionen und -anforderungen werden auf der Grundlage einer prinzipiellen Systemarchitektur eines automatischen Flugsteuerungssystems mit generischen Architekturelementen entwickelt. Eine modulare Funktionsarchitektur unterstützt die Definition generischer sowie anwendungsspezifischer Funktionen und entsprechender Systemanforderungen. Das automatische Flugbahnregelungssystem lässt sich somit auf eine Reihe von Anwendungsszenarien für bemannte und unbemannte Flugzeuge unterschiedlicher Größe, Betriebsbereichen und regulatorischer Umgebung angepasst werden.

Anschließend wird die Struktur der Flugbahnregelung entwickelt. Die Regelung der Flugbahn und der aerodynamischen Geschwindigkeit entlang der Flugbahn basiert auf einer dynamischen Inversion der translatorischen Bewegungsgleichungen mit gekoppelten, linearen Referenzmodellen, die direkt die erforderlichen spezifischen Kräfte, die zum Erzielen der gewünschten vertikalen und lateralen Flugbahnkrümmungen sowie die Beschleunigung entlang der Flugbahn notwendig sind, an die Innere Schleifen und Schubregler kommandieren. Der Ansatz ermöglicht die vollständige Nutzung der verfügbaren Leistung der inneren Schleifen für die Flugbahnregelung. Entstehende Zielkonflikte bei Manövern mit hoher Bandbreite werden durch eine integrierte Energieraten- und Kraftpriorisierungsarchitektur mit Referenzmodell-Querzuführungen aufgelöst, wo eine aktive Priorisierung zwischen vertikaler Flugbahn oder Beschleunigung sowie vertikaler oder lateraler Flugbahnkrümmung realisiert werden. Die Fluggeschwindigkeit und die Energieintegrität des Flugzeugs werden geschützt, da die Energieflussrate automatisch zugunsten der Beschleunigung am Ende des Betriebsbereiches priorisiert wird. Der Flugbahnregler ist als modulare Architektur mit verteilter Betriebsartenlogik implementiert, die eine Konfiguration und Anpassung von Funktionalitäten für neue Anwendungsplattformen ohne umfangreiche erneute Tests ermöglicht. Basierend auf

dem funktionalen Design wird eine Functional Hazard Assessment ausarbeitet und verschiedene Systemarchitekturstrategien und ihre Auswirkungen auf die Entwicklungssicherheitsniveaus der Steuerungssystemelemente diskutiert.

Darüber hinaus wird eine konfigurierbare Schnittstelle zur Bedienung und Überwachung des Flugbahnregelungssystems vorgestellt, die aus einem Bedienfeld und einem kompakten Bedienungs- und Überwachungsdisplay besteht, das in Verbindung mit dem Flugbahnregler entwickelt wurde. Das Bedienungs- und Überwachungsdesign reifte durch Iterationen ausführbarer Desktop-Anwendungen, die mit der Model-in-the-Loop-Simulationsumgebung verbunden sind, und unterstützte so die Prototypenerstellung der Regelungsalgorithmen und die frühe Validierung des Systembetriebs erheblich. Anzeigesoftware und Bedienungslogik wurden in Flugtests und im Fernsteuerungsbetrieb validiert.

Die Implementierung des Design-Modells für die Flugregelungsalgorithmen basiert auf einem modellbasierten Softwareentwicklungsprozess, an dem der angepassten Systementwicklungsprozess ausgerichtet ist. Das Design-Modell des Flugbahnreglers wird mit den weiteren funktionalen Modulen des Flugsteuerungsalgorithmus integriert und umfassend in mehreren Model-in-the-Loop-Umgebungen getestet, bevor der automatischen Quellcodegenerierung, Hardware-Software-Integration und Hardware-in-the-Loop-Tests. Die Funktions- und Leistungsanforderungen des Flugbahnreglers werden in der linearen und nichtlinearen Simulation, mit automatisierter Anforderungsbewertung über den gesamten Betriebsbereich der zutreffenden Flugzeugkonfigurationen, evaluiert.

Die Arbeit schließt mit den Ergebnissen der im Jahr 2016 durchgeführten Flugtestaktivitäten ab. Der erste Einsatz auf dem fliegenden Erprobungsträger DA42 OE-FSD wird beschrieben, wobei erste Flugtests eine gute Gesamtleistung ohne die Notwendigkeit größerer Anpassungen der Reglerparameter zeigten. Weitere Flugtestkampagnen mit der DA42 OE-FSD im Jahr 2016 werden zusammengefasst.

Acknowledgment

The main part of the work presented in this thesis was conducted during my time as research associate at the Institute of Flight System Dynamics from 2010 to 2016.

I would like to thank Prof. Dr.-Ing. Florian Holzapfel, head of the Institute of Flight System Dynamics, for the opportunity to engage in all the numerous and exciting research and industrial collaboration projects, the opportunity to challenge myself and contribute to the curriculum of the Institute, as well as his continued inspiration and support to write this thesis.

I would like to thank Prof. Dr.-Ing. Christian Breitsamter for taking over the chair of the examination board and Prof. Dr.-Ing. Stephan Myschik for the commitment as the second examiner.

I thank my former colleagues at the Institute of Flight System Dynamics, with whom I got to learn, experience, and achieve so much. Especially, I would like to thank Christoph Dörhöfer, Lars Peter, Simon Schatz, Patrick Lauffs, Volker Schneider, Agnes Steinert, Kajetan Nürnberger, and Lukas Steinert.

I thank my parents for their continuous support of my education and studies. Finally, I want to thank my wife Elin, who supported, motivated, and put up with me through this long journey.

Stockholm, November 2024

Table of Contents

List of Figures	XIII
List of Tables	XXI
Acronyms	XXV
Nomenclature	XXXI
1 Introduction	1
1.1 Background and Motivation.....	1
1.2 Literature Study / State-of-the-Art	4
1.2.1 Terminology.....	4
1.2.2 Automatic Flight Control – Manned and Unmanned Applications	5
1.2.3 Regulatory Framework and Guidance Material	8
1.2.4 Safety Assessment and Challenges in Automation Safety.....	11
1.2.5 Systems Engineering Standards and Handbooks	13
1.3 Scope, Research Objectives and Delimitations	13
1.3.1 Scope	13
1.3.2 Research Objectives	14
1.3.3 Delimitations and Related Works	17
1.4 Demonstration Platforms.....	19
1.4.1 Diamond DA42 OE-FSD.....	19
1.4.2 Dornier Do-228 D-CODE.....	21
1.4.3 SAGITTA Research Demonstrator	22
1.5 Contributions	24
1.5.1 Contribution 1: Tailored System Development Process	24
1.5.2 Contribution 2: NDI-Based Flight Path Control Application	24
1.5.3 Contribution 3: Active Control Objective Prioritization Concept.....	25
1.5.4 Contribution 4: Mode Control and Monitoring Interface.....	26
1.6 Thesis Structure	27
2 System Development Process	31
2.1 Development Process Context.....	32

2.1.1	Modular Flight Guidance and Control System	32
2.1.2	AFCS as System-of-Interest	34
2.1.3	AFCS and Application Life Cycles	35
2.1.4	Assumptions for the AFCS Development Process	37
2.2	System and Software Life Cycle Processes	40
2.2.1	ARP4754B Development Process Model	40
2.2.2	DO-178C/DO-331 Model-Based Software Development Process Model ..	45
2.2.3	ISO/IEC/IEEE 15288 Life Cycle Process Model	49
2.3	AFCS Development Process	50
2.3.1	Overview	50
2.3.2	System Definition	52
2.3.3	Safety Assessment	57
2.3.4	System Realization	62
2.3.5	Verification and Validation	63
3	Flight Dynamics Preliminaries	67
3.1	Aircraft Equations of Motion	67
3.1.1	Translational Equations of Motion	69
3.1.2	Rotational Equations of Motion	72
3.1.3	Attitude Propagation	72
3.1.4	Position Propagation	73
3.2	External Forces and Moments	74
3.2.1	Gravitational Force	75
3.2.2	Aerodynamic Forces and Moments	75
3.2.3	Propulsion Forces and Moments	76
3.2.4	Path Dynamics Couplings	77
3.3	Aircraft Environment	80
3.3.1	Wind	80
3.3.2	Continuous Turbulence	83
3.3.3	Discrete Gusts	85
3.4	Subsystems	86
3.4.1	Closed Inner Loop and Actuation Systems	86
3.4.2	Closed Thrust Loop and Propulsion Systems	87
3.4.3	Sensor Systems	88
3.5	Energy Constraints and Climb and Acceleration Tradeoffs	89
3.6	Force Constraints and Path Curvature Tradeoffs	91
4	Control Theory Preliminaries	95
4.1	Nonlinear Dynamic Inversion	95
4.2	Reference Models for Desired Closed Loop Dynamics	100
4.3	Error Dynamics and Stabilizing Controller	102
4.4	Consideration of Actuation Dynamics through Pseudo Control Hedging	106

5	System Definition	111
5.1	System Functions & Architecture	111
5.1.1	Example Use Case: Optionally Piloted Vehicle AFCS	112
5.1.2	AFCS System Functions	113
5.1.3	System Requirements	123
5.2	Controller Design	136
5.2.1	Functional Architecture Preliminary Considerations	136
5.2.2	Path Control Architecture	140
5.2.3	Control of the Vertical Flight Path	146
5.2.4	Control of the Lateral Flight Path	148
5.2.5	Control of the Aerodynamic Speed	151
5.2.6	Trajectory Control Modes	154
5.2.7	Radio Navigation & Approach Modes	157
5.2.8	Energy-Based Prioritizations	158
5.2.9	Force-Based Prioritizations	167
5.2.10	Inner Loop – Normal and Lateral Specific Force Control	171
5.2.11	Autothrust – Longitudinal Acceleration Control	175
5.2.12	Direct Path Curvature Control	177
5.3	Gain and Parameter Design	179
5.3.1	Gain and Parameter Design Preliminary Considerations	179
5.3.2	Controller Design and Assessment Framework	180
5.3.3	Path Control Parameters	187
5.3.4	Speed Control Parameters	190
5.3.5	Energy Protection Parameters	190
5.4	Mode Control and Monitoring Interface Design	192
5.4.1	Preliminary Design Considerations	192
5.4.2	Architecture	192
5.4.3	Mode Control Panel Design	195
5.4.4	Mode Control and Monitoring Display Design	197
6	Safety Assessment	201
6.1	FHA Preliminary Considerations	202
6.1.1	System Functions under Analysis	202
6.1.2	Flight Phases	203
6.1.3	Guidance Material	203
6.2	Summary of FHA Results	204
6.2.1	Significant Failure Conditions	204
6.2.2	Quantitative Safety Objectives	206
6.2.3	Functional Development Assurance Levels	207
6.3	System Architecture Considerations	207
6.3.1	Failure Condition Mitigation Strategies	208
6.3.2	Architecture and Item Development Assurance Approaches	209
6.3.3	Primary and Secondary AFCS	219

7	System Realization	221
7.1	Design Model Implementation.....	221
7.1.1	Implementation and Integration Environment.....	223
7.1.2	AFCS Design Model Implementation	224
7.1.3	Library Block Unit Testing.....	230
7.1.4	FCC Design Model-in-the-Loop.....	234
7.1.5	AFCS Design Model-in-the-Loop.....	235
7.2	Source Code Generation and Software Integration	237
7.3	Hardware-Software Integration	238
7.4	Aircraft Integration.....	241
8	Verification and Validation	243
8.1	Reference Configuration	244
8.1.1	Test Condition and Configuration Index	244
8.1.2	Closed Inner Loop and Autothrust Dynamics	245
8.1.3	Controller Gains and Parameters	247
8.1.4	Internal Command Processing and Sensor Delays.....	250
8.2	Linear Assessment.....	251
8.2.1	Path Tracking Performance	252
8.2.2	Disturbance Rejection	255
8.2.3	Stability and Robustness	257
8.3	Nonlinear Assessment	260
8.3.1	Path Tracking in Smooth Air.....	261
8.3.2	Path Tracking in Turbulent Air	263
8.3.3	Control Objective Conflict Resolution	266
8.3.4	Energy Protections	271
8.3.5	Mode Transition Logic	273
8.4	Flight Testing.....	275
8.4.1	Modular FGCS Deployment	276
8.4.2	Nominal Flight Path Tracking	283
8.4.3	Energy Protections	285
9	Conclusions and Outlook.....	287
9.1	Contribution 1: Tailored System Development Process.....	288
9.2	Contribution 2: NDI-Based Flight Path Control Application	288
9.3	Contribution 3: Active Control Objective Prioritization Concept	288
9.4	Contribution 4: Mode Control and Monitoring Interface	289
9.5	Outlook.....	289
	Bibliography	291
A	Scientific Publications.....	301
B	Coordinate Systems and Transformations.....	303
B.1	Coordinate Systems.....	304

B.2	Transformation Matrices and Angular Rates.....	307
C	Mode Control and Monitoring Interface Specification.....	313
C.1	MCP/MCMD Command Logic.....	313
C.2	MCMD Elements	329
D	Additional Flight Test Data.....	335
D.1	AFCS Deployment Flight Test Cards	335
D.2	Flight Test Logs.....	344
E	Functional Hazard Assessment Worksheets	353
E.1	FHA Worksheet Explanation	353
E.2	Depth of Analysis	355
E.3	Automatic Flight Control Functions	357
E.4	HMI Functions	362
E.5	Autopilot Functions.....	364
E.6	Autothrust Functions	365
E.7	Safety Functions	366

List of Figures

Figure 1.1: Modular FGCS overview with the main contributions of the author and the scope of this thesis (the development of the AFCS module with the automatic flight path controller, and the HMI) highlighted in red. 3

Figure 1.2: Regulatory structure in the United States and European Union for the certification of manned civil aircraft and systems. 9

Figure 1.3: Interrelations of principal guidance material for civil aircraft and systems, their safety assessment, and hardware and software development. 10

Figure 1.4: DA42 OE-FSD at Wiener Neustadt airport in January 2016 (photo by the author). 20

Figure 1.5: The DLR Dornier Do-228 D-CODE prior to first flight tests in Oberpfaffenhofen, August 2016 (photo by the author). 21

Figure 1.6: SAGITTA Research Demonstrator mockup on display at TUM (photo by the author). 23

Figure 1.7: Thesis structure and chapters. 29

Figure 2.1: A hierarchical view of the modular FGCS, with principal system levels and the narrower and wider SOI indicated. 32

Figure 2.2: Principal Modular FGCS life cycle model with example AFGCS application life cycle models. 36

Figure 2.3: ARP4754B general life cycle model, adopted from [37]. 40

Figure 2.4: ARP4754B aircraft or system development process model, adopted from [37]. ... 41

Figure 2.5: ARP4754B safety assessment process and interactions with the development process, adopted from [37]. 43

Figure 2.6: ARP4754B Function/Item Development Assurance Level (FDAL/IDAL) assignment process, adopted from [37]. 44

Figure 2.7: DO-178C software development process model. 46

Figure 2.8: DO-178C development and verification traceability requirements depending on development assurance level. 48

Figure 2.9: ISO/IEC/IEEE 15288:2023 system life cycle process framework. 49

Figure 2.10: AFCS development process with related thesis chapters and sections. 51

Figure 2.11: Principal AFCS architecture with generic architecture elements, adopted from [35]. 53

Figure 3.1: Specific forces for path control. 80

Figure 3.2: Wind velocity component and relation between kinematic and aerodynamic flight path angles.....	81
Figure 3.3: Turbulence levels and exceedance probabilities according to MIL-HDBK-1797 [108] and AS94900A [103], adopted from [108].	84
Figure 3.4: "1-cosine" discrete gust.....	86
Figure 4.1: Reference model based dynamic inversion.	100
Figure 4.2: Linear reference model of order n for pseudo-controls up to order r	101
Figure 4.3: Reference model based dynamic inversion with error dynamics stabilization. ..	104
Figure 4.4: Linear reference model of order n with Pseudo-Control Hedging.	108
Figure 4.5: Reference model based dynamic inversion with error dynamics stabilization and Pseudo-Control Hedging.	109
Figure 5.1: Example mission profile for an OPV, with typical vertical/speed control and lateral control modes for conventional and UAV operation.	113
Figure 5.2: Principal coupling between automatic flight control modes and autopilot/autorust functions.	116
Figure 5.3: Principal development of system functions to sets of general and function-specific functional requirements.	124
Figure 5.4: Relation between general and function-specific requirements, design model, and verification artifacts.....	125
Figure 5.5: Vertical/speed control modes transitions, selected and managed.	129
Figure 5.6: Lateral control modes transitions, selected and managed.....	130
Figure 5.7: Desired and adequate step response performance example.	131
Figure 5.8: Derivation of inner loop performance envelopes for desired and adequate path loop control performance.....	134
Figure 5.9: Linear MATLAB model structure for derivation of inner loop performance requirements for vertical/speed control.	135
Figure 5.10: Principal AFCS functional architecture elements.	136
Figure 5.11: Principal overview of the AFCS distributed mode control logic.....	138
Figure 5.12: Basic flight path control structure with reference models and error controllers.	140
Figure 5.13: First- and second-order reference model comparison, with command to output state and pseudo-control step responses and closed loop Bode diagrams, respectively. ...	144
Figure 5.14: Flight path angle limit filtering at open climb/descent activation.	153
Figure 5.15: Trajectory frame as utilized by the trajectory controller with the trajectory deviations and their corresponding time derivatives, adopted from [111].	155
Figure 5.16: Altitude capture and track geometry.	156
Figure 5.17: Basic flight path control structure with control plane coupling for energy rate and force prioritizations.	159
Figure 5.18: Phase plane regions for energy protection.	163
Figure 5.19: Protection limits scaling principle.	164
Figure 5.20: Acceleration, flight path angle, and throttle limits for energy integrity protection.	166

Figure 5.21: Force limitations for lateral (upper) and vertical (lower) path curvature priority.	170
Figure 5.22: Body axes specific force and bank angle command generation.	173
Figure 5.23: Autothrust controller structure.	176
Figure 5.24: Direct path curvature stick mode with limitations for energy protection.	178
Figure 5.25: Linear MATLAB model structure for vertical plane design and assessment, including closed plant and inner loop dynamics.	181
Figure 5.26: Linear MATLAB model structure for lateral plane design and assessment, including closed plant and inner loop dynamics.	182
Figure 5.27: Main gain and parameter design routine for longitudinal/lateral AFCS.	186
Figure 5.28: Effect of reference model time constant on path step response.	188
Figure 5.29: Simplified plant model with disturbances.	189
Figure 5.30: Effect of different values of proportional gain k_P on flight path state and pseudo control disturbance responses, with integral gain $k_I = 0$	189
Figure 5.31: Effect of different values of integral gain k_I on flight path state and pseudo control disturbance responses, with proportional gain $k_P = 1$	190
Figure 5.32: Overview of HMI architecture and MCP-MCMD-FCC-ACE-DCU interfaces.	193
Figure 5.33: Mode Control Panel (MCP) layout.	195
Figure 5.34: Mode Control and Monitoring Display (MCMD) layout with sections.	198
Figure 6.1: Principal simplex, duplex and dual duplex AFCS architectures.	211
Figure 6.2: Functional fault tree analysis for the simplex architecture.	213
Figure 6.3: Functional fault tree analysis for the duplex fail-passive architecture.	214
Figure 6.4: Functional fault tree analysis for the dual duplex fail-operational architecture.	217
Figure 7.1: AFCS top-level model.	224
Figure 7.2: AFCS main algorithm block.	227
Figure 7.3: AFCS algorithms implemented as individual library blocks linked in the main model.	228
Figure 7.4: AFCS algorithm implementation example: calculation of maximum acceleration for speed or vertical path prioritization, according to Eqs. (5.93)-(5.94).	228
Figure 7.5: AFCS algorithm implementation example: library block containing the altitude capture/hold and flight level change mode control logic, as part of the distributed mode control logic of the AFCS.	229
Figure 7.6: AFCS algorithm implementation example: local flight level change and altitude capture/hold logic, as part of the distributed mode control logic of the AFCS.	229
Figure 7.7: AFCS algorithm implementation example: transition criteria and transition action functions as part of the flight level change and altitude capture/hold logic.	230
Figure 7.8: Generic MATLAB Unit Test Framework test flow, used to automate library block testing as well as model-in-the-loop and nonlinear assessment tests.	231
Figure 7.9: Simulink test harness model structure for library block unit testing, with main test function and local test functions executing the test harness model.	231
Figure 7.10: Simulink test harness model structure example for library block unit testing, here the altitude hold/flight level change logic block.	232

Figure 7.11: Simulink test harness model structure for AFCS model-in-the-loop verification.237

Figure 7.12: System architecture overview of the hardware-in-the-loop (HIL) laboratory setup for the Do-228 D-CODE experimental AFCS, with AFCS subsystems and HIL subsystems identified.240

Figure 7.13: The hardware-in-the-loop (HIL) laboratory setup for the Do-228 D-CODE experimental AFCS at TUM-FSD (photo by the author).240

Figure 8.1: Test condition and configuration index for the assessment of the AFCS for the reference configuration DA42 OE-FSD. Test conditions are defined over indicated airspeed and altitude envelope, with trimmed flight conditions scheduled over static and dynamic pressure. Double x- and y-scales illustrates the corresponding values. Test configurations include flaps up and down, as well as gear up and down configurations.245

Figure 8.2: Inner loop and autothrust performance and over the envelope: step response curves, overshoots, and settling times for linear specific force (left column), bank angle (middle column), and normal specific force (right column). Gear up, flaps up configuration.246

Figure 8.3: AFCS reference model time constants and error controller gains over the envelope for the flight path angle loop. Gear up, flaps up configuration.248

Figure 8.4: AFCS reference model time constants and error controller gains over the envelope for the speed by thrust loop. Gear up, flaps up configuration.248

Figure 8.5: AFCS reference model time constants and error controller gains over the envelope for the speed by pitch loop. Gear up, flaps up configuration.249

Figure 8.6: AFCS reference model time constants and error controller gains over the envelope for the track angle loop. Gear up, flaps up configuration.250

Figure 8.7: Linear step responses over the envelope for speed by thrust (left column), track angle (middle column), and flight path angle (right column). Gear up, flaps up configuration.253

Figure 8.8: Step response performance requirements for the speed by thrust, track angle, and flight path angle loops. Overshoot and settling time analysis and performance requirement evaluation over the envelope. Gear up, flaps up configuration.254

Figure 8.9: Linear step responses over the envelope for a speed/path disturbance input. Speed by thrust (left column), track angle (middle column) and flight path angle (right column) responses. Gear up, flaps up configuration.256

Figure 8.10: Flight path angle and speed by thrust loop stability and robustness performance requirements assessment over the envelope. Gain and phase margins for the flight path angle loop cut at the elevator actuator command $\phi\eta C$ (left) and for the speed by thrust loop cut at throttle command $\delta T, C$. Gear up, flaps up configuration.258

Figure 8.11: Nichols diagrams for the flight path angle loop cut at the elevator actuator command $\phi\eta C$ (left) and for the speed by thrust loop cut at throttle command $\delta T, C$. Gear up, flaps up configuration.258

Figure 8.12: Track angle loop stability and robustness performance requirements assessment over the envelope. Gain and phase margins for the for the track angle loop cut

at the aileron actuator command $\phi\xi C$ (left) and rudder actuator command $\phi\zeta C$ (right). Gear up, flaps up configuration.	259
Figure 8.13: Nichols diagrams for the track angle loop cut at the aileron actuator command $\phi\xi C$ (left) and rudder actuator command $\phi\zeta C$ (right). Gear up, flaps up configuration.....	259
Figure 8.14: Command histories for maneuvers with simultaneous speed, vertical path, and lateral path tracking for nonlinear simulation testing. Maneuver 1 consists of nominal commands to test tracking performance in smooth and turbulent air. Maneuver 2 combines greater command inputs to trigger control objective conflicts and test resolution functionalities.....	260
Figure 8.15: Speed and flight path command tracking during combined nominal vertical and lateral flight path and speed commands, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.....	262
Figure 8.16: Speed and flight path command tracking during combined nominal vertical and lateral flight path and speed commands, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.....	264
Figure 8.17: Excerpt of speed and flight path command tracking in Figure 8.16, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.....	265
Figure 8.18: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Speed and lateral plane curvature priority. $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.....	267
Figure 8.19: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Vertical path and vertical plane curvature priority. $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.....	268
Figure 8.20: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Speed and lateral plane curvature priority. Evaluated over envelope for gear up/flaps up configuration.....	269
Figure 8.21: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Vertical path and vertical plane curvature priority. Evaluated over envelope for gear up/flaps up configuration.....	270
Figure 8.22: Demonstration of low-speed protection (left column) and high-speed protection (right column) responses. Gear up, flaps up configuration.....	272
Figure 8.23: Demonstration of an open climb/descent maneuver with altitude capture and hold transitions. Mode changes indicated by a solid, black vertical line.....	274
Figure 8.24: AFCS deployment flight test, flight test #1, 2016-01-15. Vertical plane/speed active control modes and command tracking.....	278
Figure 8.25: AFCS deployment flight test, flight test #1, 2016-01-15. Lateral plane active control modes and command tracking.....	279
Figure 8.26: First AFCS controller activation, flight test #1, 2016-01-15. Flight test card 1, activation of pitch and roll hold modes, the default activation modes of the DA42 specific implementation.....	280
Figure 8.27: First AFCS lateral path mode activation, flight test #1, 2016-01-15. Flight test card 6, activation of heading mode, acquiring and tracking heading commands from the MCP.....	281

Figure 8.28: First AFCS vertical path mode activation, flight test #1, 2016-01-15. Flight test card 10, activation of vertical speed mode, tracking vertical speed commands from the MCP and acquiring and capturing altitude commands.282

Figure 8.29: Section of flight test #2, 2016-08-17. Vertical and lateral path as well as speed command tracking in flight.284

Figure 8.30: Section of flight test #2, 2016-08-17. Vertical and lateral path as well as speed command tracking in flight.286

Figure B.1: Coordinate systems and rotation angles.303

Figure C.1: Speed command logic inputs and outputs.313

Figure C.2: Speed command logic structure.314

Figure C.3: Speed preselect logic.314

Figure C.4: Speed preselect set flag logic.315

Figure C.5: Speed desired target set flag logic.315

Figure C.6: Heading/Track command logic inputs and outputs.316

Figure C.7: Heading/Track command logic structure.317

Figure C.8: Heading/Track toggle logic.318

Figure C.9: Heading/Track preselect logic.318

Figure C.10: Heading/Track preselect set flag logic.319

Figure C.11: Heading/Track desired target set flag logic.319

Figure C.12: Heading/Track turn left/right flag logic.319

Figure C.13: Altitude command logic inputs and outputs.320

Figure C.14: Altitude command logic structure.321

Figure C.15: Altitude feet/meter toggle logic.321

Figure C.16: Altitude preselect logic.322

Figure C.17: Altitude preselect target set flag logic.322

Figure C.18: Altitude desired target set flag logic.322

Figure C.19: Vertical speed / flight path angle command logic inputs and outputs.323

Figure C.20: Vertical speed / flight path angle command logic structure.324

Figure C.21: Vertical speed / flight path angle toggle logic.325

Figure C.22: Vertical speed preselect logic.325

Figure C.23: Flight Path Angle preselect logic.326

Figure C.24: Vertical speed / flight path angle preselect set flag logic.326

Figure C.25: Vertical speed / flight path angle desired target set flag logic.326

Figure C.26: Course command logic inputs and outputs.327

Figure C.27: Course command logic structure.327

Figure C.28: Course command logic.328

Figure D.1: Flight Test Cards – General information sheet.335

Figure D.2: Flight Test Card 1 – Controller activation, pitch/roll hold test.336

Figure D.3: Flight Test Card 2 – Altitude/roll hold test.336

Figure D.4: Flight Test Card 3 – Altitude/heading hold test.337

Figure D.5: Flight Test Card 4 – Altitude/heading test, small right turn heading change.337

Figure D.6: Flight Test Card 5 – Altitude/heading test, small left turn heading change.338

Figure D.7: Flight Test Card 6 – Altitude/heading test, large right turn heading change.....	338
Figure D.8: Flight Test Card 7 – Altitude/heading test, large left turn heading change.....	339
Figure D.9: Flight Test Card 8 – Controller reset, pitch/roll hold test.	339
Figure D.10: Flight Test Card 9 – Vertical speed/roll hold test.....	340
Figure D.11: Flight Test Card 10 – Vertical speed/altitude capture/hold test.	340
Figure D.12: Flight Test Card 11 – Controller reset, pitch/roll hold test.	341
Figure D.13: Flight Test Card 12 – Open climb/heading test.	341
Figure D.14: Flight Test Card 13 – Open descent/heading test.	342
Figure D.15: Flight Test Card 14 – Controller reset, pitch/roll/speed hold test.....	342
Figure D.16: Flight Test Card 15 – Altitude/heading/speed test.....	343
Figure D.17: Flight Test Card 16 – Altitude/lateral trajectory/speed test.....	343
Figure D.18: 2016-08-17, flight test #1. Demonstration check flight, tracking of lateral trajectory. Vertical plane/speed active control modes and command tracking.....	345
Figure D.19: 2016-08-17, flight test #1. Demonstration check flight, tracking of lateral trajectory. Lateral plane active control modes and command tracking	346
Figure D.20: 2016-08-17, flight test #2. AFCS additional flight tests. Vertical plane/speed active control modes and command tracking.	347
Figure D.21: 2016-08-17, flight test #2. AFCS additional flight tests. Lateral plane active control modes and command tracking.	348
Figure D.22: 2016-08-18, flight test #1. Demonstration flight, tracking of lateral trajectory. Vertical plane/speed active control modes and command tracking.	349
Figure D.23: 2016-08-18, flight test #1. Demonstration flight, tracking of lateral trajectory. Lateral plane active control modes and command tracking.	350
Figure D.24: 2016-08-18, flight test #2. AFCS flight test, ATOL preparatory tests. Vertical plane/speed active control modes and command tracking.	351
Figure D.25: 2016-08-18, flight test #2. AFCS flight test, ATOL preparatory tests. Lateral plane active control modes and command tracking.	352

List of Tables

- Table 1.1 Overview of common automatic flight control system terms. 4
- Table 1.2: Control objective prioritization strategy for the modular FGCS. 16
- Table 1.3: Diamond DA42 OE-FSD technical data. 20
- Table 1.4: Dornier Do-228 D-CODE technical data (adopted from [97, 91]). 21
- Table 1.5: SAGITTA Research Demonstrator technical data. 23
- Table 2.1: FHA worksheet columns and explanation. 58
- Table 2.2: Pilot reaction times, adopted from [104]. 59
- Table 2.3: Classification of failure conditions and associated qualitative probability requirements, adopted from [104]. 60
- Table 2.4: Classification of AFCS hardover depending on authority [104]. 61
- Table 2.5: Relationship between airplane classes, probabilities, failure condition severity, and IDAL requirements for primary (P) and secondary (S) items (adopted from [104] Figure 2).. 62
- Table 2.6: Means of verification and validation. 64
- Table 3.1: Sensor sources and measurement inputs to the path control. 88
- Table 5.1: Application-generic and application-specific system functions overview. 114
- Table 5.2: Guidance modes and combinations for vertical/lateral plane and speed. 115
- Table 5.3: Coupled path and trajectory control modes for vertical/lateral plane and speed. 118
- Table 5.4: Coupled control objective prioritization modes for vertical/lateral plane and speed. 119
- Table 5.5: HMI functions and guidance material references. 120
- Table 5.6: Autopilot and autothrust functions and guidance material references. 121
- Table 5.7: Safety functions and guidance material references. 121
- Table 5.8: Grouping of general and function-specific functional requirements. 124
- Table 5.9: List of application-generic general functional requirements. 127
- Table 5.10: Example general performance requirement – general stability margins. 128
- Table 5.11: Example general performance requirement – mode transition transients. 128
- Table 5.12: Partial list of application-generic function-specific functional requirements. 132
- Table 6.1: Application-generic system function subject to FHA. 202
- Table 6.2: Summary of relationship between probabilities, failure condition severity and DAL requirements for primary (P) and secondary (S) items for Class II aircraft (extract from [104] Figure 2). 204
- Table 6.3: Summary of significant failure conditions. 205

Table 6.4: Hardover malfunction classification and required development assurance for different application platform classifications.	208
Table 6.5: Valid IDAL assignments for the simplex architecture for top-level FDAL A.	213
Table 6.6: Valid FDAL/IDAL assignments for the duplex architecture with INS/ADS and FCC HW independence for top-level FDAL A.	215
Table 6.7: Valid FDAL/IDAL assignments for the duplex architecture with INS/ADS independence but not FCC HW independence for top-level FDAL A.	215
Table 6.8: Valid FDAL/IDAL assignments for the duplex architecture with FCC HW independence but not INS/ADS independence for top-level FDAL A.	216
Table 6.9: Valid FDAL/IDAL assignments for the dual duplex architecture with full INS/ADS, FCC HW, actuation system and position sensor independence for top-level FDAL A.	218
Table 6.10: Valid FDAL/IDAL assignments for the dual duplex architecture with partial INS/ADS independence for top-level FDAL A.	219
Table 6.11: Valid FDAL/IDAL assignments for the dual duplex architecture with partial FCC HW independence for top-level FDAL A.	219
Table 7.1: Excerpt of library block unit test coverage report using <i>Simulink Coverage</i> , with summary of decision, condition, and modified condition/decision coverage for the tested block and its child elements.	233
Table 7.2: Test case execution summary table for the <i>af_nlaflmod</i> block, with test cases included in the test script <i>af_nlaflmod_testscript</i>	233
Table 8.1: Flight test history of the modular FGCS to which the author contributed with planning and execution, and from which results are included in this thesis.	275
Table C.1: Alternative indications for preselected airspeed (upper row).	329
Table C.2: Alternative indications for commanded airspeed (lower row).	329
Table C.3: Alternative indications for preselected heading / track (upper row).	330
Table C.4: Alternative indications for commanded heading / track (lower row).	330
Table C.5: Alternative indications for preselected altitude (upper row).	331
Table C.6: Alternative indications for commanded altitude (lower row).	331
Table C.7: Alternative indications for preselected vertical speed or flight path angle (upper row).	332
Table C.8: Alternative indications for commanded vertical speed or flight path angle (lower row).	332
Table D.1: List of flight tests included in this appendix with reference to figures.	344
Table E.1: FHA worksheet columns and explanation.	353
Table E.2: Depth of analysis requirements for failure condition classification and type of system combinations.	355
Table E.3: Guidance modes FHA work sheet.	357
Table E.4: Coupled path and trajectory control modes FHA work sheet.	358
Table E.5: Coupled control objective prioritization modes FHA work sheet.	360
Table E.6: HMI Functions FHA work sheet.	362
Table E.7: Autopilot Functions FHA work sheet.	364
Table E.8: Autothrust Functions FHA work sheet.	365

Table E.9: Safety Functions FHA work sheet..... 366

Acronyms

Acronym	Definition
A/C	Aircraft
AC	Advisory Circular
ADI	Attitude Direction Indicator
ADS	Air Data System
AFCF	Automatic Flight Control Functions
AFCS	Automatic Flight Control System
AFGCS	Automatic Flight Guidance and Control System
AFGCS	Automatic Flight Guidance and Control System
AFS	Auto Flight System
AHRS	Attitude and Heading Reference System
AIL	Aircraft-in-the-Loop
AILS	Aircraft-in-the-Loop Simulation
ALT, ALT*	Altitude Hold/Capture (AFCS Mode)
AMC	Acceptable Means of Compliance
AP	Autopilot or Automatic Pilot
APPR	Approach
ARP	Aerospace Recommended Practice
ATHR	Autothrust/Autothrottle
ATOL	Automatic Takeoff and Landing
CAS	Calibrated Airspeed
CAT	Catastrophic
CCA	Common Cause Analysis

Acronym	Definition
CFR	Code of Federal Regulations
CMD	Command
COTS	Commercial off-the-Shelf
CRI	Certification Review Items
CRS	Course
CS	Certification Specification
CWS	Control Wheel Steering
DAL	Development Assurance Level
DCU	Data Concentrator Unit
DD	Dependency Diagram
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DME	Distance Measuring Equipment
EASA	European Aviation Safety Agency
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EDMN	Mindelheim-Mattsies Airport (ICAO Airport Code)
EFMS	Experimental Flight Management System
ELOS	Equivalent Level of Safety
EO	External Pilot
ETSO	European Technical Standard Order
EUROCAE	European Organisation for Civil Aviation Equipment
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulations
FBW	Fly-by-Wire
FCAU	Flight Control Actuation Units
FCS	Flight Control System
FD	Flight Director
FDAL/IDAL	Functional/Item Development Assurance Level
FDM	Flight Dynamics Model
FFS	Functional Failure Sets

Acronym	Definition
FGCS	Flight Guidance and Control System
FH	Flight Hour
FHA	Functional Hazard Assessment
FLCH	Flight Level Change
FMEA	Failure Modes and Effects Analysis
FMS	Flight Management System
FO	Flight Operator
FPA	Flight Path Angle (AFCS Mode)
FPV	Flight Path Vector
FTA	Fault Tree Analysis
G/S, G/S*	Glideslope Track/Capture (AFCS Mode)
GA	General Aviation
GNSS/INS	Global Navigation Satellite System-aided Inertial Navigation System
HAZ	Hazardous
HDG	Heading (AFCS Mode)
HIL	Hardware-in-the-Loop
HILS	Hardware-in-the-Loop Simulation
HMI	Human-Machine-Interface
HSI	Horizontal Situation Indicator
HW	Hardware
I/O	Input-Output
IAS	Indicated Airspeed
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INCOSE	International Council on Systems Engineering
ISA	International Standard Atmosphere
ISO	International Organization for Standardization
LAT	Lateral Plane Priority (AFCS Mode)
LLR	Low-Level Requirements

Acronym	Definition
LNAV	Lateral Navigation (AFCS Mode)
LOAN	Wiener Neustadt East Airport (ICAO Airport Code)
LOC, LOC*	Localizer Track/Capture (AFCS Mode)
MA	Markov Analysis
MAJ	Major
MBSE	Model-Based Systems Engineering
MCDC	Modified Condition/Decision Coverage
MCMD	Mode Control and Monitoring Display
MCP	Mode Control Panel
MEMS	Micro-Electro-Mechanical System
MIL	Model-in-the-Loop
MILS	Model-in-the-Loop Simulation
MIMO	Multiple-Input Multiple-Output
MIN	Minor
MMS	Mission Management System
M-NG	Multi-Purpose Platform, Next Generation
MON	Monitor
MOPS	Minimum Operational Performance Standard
MRE	Multiple Reciprocating Engine
MTE	Multiple Turbine Engine
MTOW	Maximum Takeoff Weight
NDI	Nonlinear Dynamic Inversion
NED	North-East-Down
OP CLB/DES	Open Climb/Descent (AFCS Mode)
OPV	Optionally Piloted Vehicle
PB	Pushbutton
PCH	Pseudo-Control Hedging
PFD	Primary Flight Display
PIT	Pitch Hold (AFCS Mode)
PPRIO	Path Priority (AFCS Mode)

Acronym	Definition
PROT	Protection
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Analysis
PTF	Permit-to-Fly
RMS	Root Mean Square
ROL	Roll Hold (AFCS Mode)
RPV	Remotely Piloted Vehicle
RTCA	RTCA Inc., formerly Radio Technical Commission for Aeronautics
SAE	Society of Automotive Engineers
SC	Special Condition
SED	System Environment Description
SILS	Software-in-the-Loop Simulation
SISO	Single-Input Single-Output
SL/SF	Simulink/Stateflow
SOI	System-of-Interest
SPD PIT	Speed by Pitch (AFCS Mode)
SPD THR	Speed by Thrust (AFCS Mode)
SRATS	Systems Requirements Allocated to Software
SRB	Safety Relay Box
SSA	System Safety Analysis
STC	Supplemental Type Certificate
STE	Single Turbine Engine
STPA	System-Theoretic Process Analysis
SW	Software
TECS	Total Energy Control System
THS	Trimmable Horizontal Stabilizer
TRK	Track (AFCS Mode)
TSN	Transition
TSO	Technical Standard Order
TUM-FSD	TUM Institute of Flight System Dynamics

Acronyms

Acronym	Definition
UAS	Unmanned Aircraft Systems
UAV	Unmanned Aerial Vehicle
V/S	Vertical Speed (AFCS Mode)
VERT	Vertical Plane Priority (AFCS Mode)
VNAV	Vertical Navigation (AFCS Mode)
VOR	Very High Frequency Omnidirectional Range Station
VPRIO	Speed Priority (AFCS Mode)
WGS84	World Geodetic System 84
YD	Yaw Damper

Nomenclature

This section contains an overview of the symbols, indices and abbreviations used throughout the thesis. Common symbols and indices are written in italics. Matrices and vectors are written in bold. Vectors with a physical interpretation in the three-dimensional Euclidean space are marked with an arrow on top of the symbol, \vec{x} ; the following index positions apply:

$$\left(\begin{array}{l} \vec{x} \text{Reference point} \\ \text{Type of motion or source of force/moment} \end{array} \right) \begin{array}{l} \text{Reference frame} \\ \text{Notation frame} \end{array}$$

Symbols are sorted as Latin and Greek capital and small letters, respectively.

Latin Capital Letters

Symbol	Description	Unit
$A(x)$	Decoupling matrix	[...]
A	System matrix of a state space model on the form $\dot{x} = Ax + Bu, y = Cx + Du$	[...]
B	Input matrix of a state space model on the form $\dot{x} = Ax + Bu, y = Cx + Du$	[...]
C	Dimensionless aerodynamic force or moment coefficient	-
C	Output matrix of a state space model on the form $\dot{x} = Ax + Bu, y = Cx + Du$	[...]
D	Drag force	N
D	Feedthrough matrix of a state space model on the form $\dot{x} = Ax + Bu, y = Cx + Du$	[...]
E	Energy	J
$F(s)$	Feedforward transfer function	[...]
F^i, \vec{F}^i	Force magnitude acting on point i , force vector acting on point i	N
F	Dynamics of a feedback linearized system from input vector u to vector $[y_i^{(r_i)}]$ of outputs y_i of relative degree r_i , respectively	[...]

Symbol	Description	Unit
$G_{ij}(s)$	Linear transfer function from input j to output i	-
H^i, \vec{H}^i	Angular momentum magnitude around point i , angular momentum vector around point i	$\text{kg} \cdot \text{m/s}^2$
I	Identity matrix	-
\mathbf{I}^P	Inertia tensor with respect to a point P	$\text{kg} \times \text{m/s}^2$
$K(s)$	Linear controller transfer function	-
$L(s)$	Closed loop transfer function	-
L_i	Length scale for axis i	m
L	Lie derivative	-
L	Lift force	N
L	Propulsive power	W
L	Rolling moment	Nm
M^i, \vec{M}^i	Moment magnitude around point i , moment vector around point i	Nm
M	Pitching moment	Nm
\mathbf{M}_{ij}	Transformation matrix from coordinate frame j into coordinate frame i	-
N	Yawing moment	Nm
Q	Side force	N
$S(s)$	Sensitivity transfer function	-
$T(s)$	Complimentary sensitivity transfer function	-
T	Time constant	s
T	Thrust force	N
V, \vec{V}	Velocity magnitude, velocity vector	m/s
$W(s)$	Weighting function, frequency distribution of uncertainty	-
X	Force along x -axis in a given coordinate system	N
Y	Force along y -axis in a given coordinate system	N
Z	Force along z -axis in a given coordinate system	N

Latin Small Letters

Symbol	Description	Unit
f	Specific force; mass-specific force, i.e., F/m	m/s^2

g	Gravitational constant	m/s ²
h	Altitude	m
k_i	Controller gain; different indices	[...]
m	Aircraft mass	kg
p, \vec{p}	Linear momentum magnitude, linear momentum vector	kg · m/s ²
p_s	Static pressure	Pa
r^i, \vec{r}^i	Vector component from origin to point i , vector from origin to point i	m
r^{ij}, \vec{r}^{ij}	Vector component from point i to point j , vector from point i to point j	m
s	Laplace variable	-
t	Time	s
u, \mathbf{u}	Control variable, control vector	[...]
\bar{q}	Dynamic pressure	Pa
x, \mathbf{x}	State variable, state vector	[...]

Greek Capital Letters

Symbol	Description	Unit
Δ	Model error; deviation between real system and model dynamics	[...]
Δ	Incremental force: force component related to path curvature, reduced by the component related to gravitational force	N
Φ	Aircraft bank angle	rad
Ψ	Aircraft heading	rad
Θ	Aircraft pitch angle	rad

Greek Small Letters

Symbol	Description	Unit
α	Angle of attack	rad
β	Angle of sideslip	rad
χ	Course angle	rad
δ_T	Throttle position	-
η	Elevator deflection	rad

Nomenclature

Symbol	Description	Unit
γ	Flight path angle	rad
λ	Longitude	rad
μ	Flight path bank angle	rad
μ	Latitude	rad
ν, \mathbf{v}	Pseudo control variable / Pseudo control vector	[...]
$\omega, \vec{\omega}$	Angular velocity, angular velocity vector	rad/s
ω_0	Natural frequency of a second order linear system	rad/s
ρ	Density	kg/m ³
ξ	Aileron deflection	rad
ζ	Rudder deflection	rad

1 Introduction

1.1 Background and Motivation

Accurate control of the aircraft flight path, i.e., the climb and track angles, as well as the airspeed along the flight path, are central functional elements of most modern automatic flight control systems. Depending on the level of automation, the flight path control loops may execute direct medium-level flight path commands from for example a pilot or remote operator, or, as in a traditional, cascaded automatic flight control system structure, be used as the basis for higher-level functions like trajectory control, waypoint navigation, and approach and landing modes, including flare.

Key developments and enabling technologies have over the last decades opened new opportunities for low-cost avionics approaches, and modular, easily configurable, and certifiable flight guidance and control algorithms. These developments have spurred an increased digitalization, availability, and affordability of avionics and automatic flight control systems for regional, business, and General Aviation (GA) aircraft, as well as the expanding civil and military Unmanned Aircraft System (UAS) segments:

- Advances in Commercial-off-the-Shelf (COTS) embedded computer technologies outside the aerospace sector have led to a rapid increase in microprocessor speed, performance, memory, and miniaturization, with embedded computers getting faster, smaller, and cheaper, yet robust, affordable, and reliable for aerospace applications
- Reliable and easy-to-configure data bus and communication protocols with low-cost hardware controllers adopted from the automobile, IT, and automation industry, and extended for use in aerospace applications
- Micro-Electro-Mechanical System (MEMS) gyros and accelerometers as well as satellite navigation are main drivers of the rapidly expanding miniature UAS market but have also found use in certified flight control systems
- Compact and high-powered electro-mechanical actuators
- Affordable and integrated software development tools and environments for modern software development techniques (e.g., model-based) supporting safety-critical applications
- Recognition of technological advances by regulating authorities with corresponding revision of certification approaches and regulations to support the development and integration of new and affordable systems in order to enhance safety

The technological evolutions and revised certification strategies have made new functionalities for the low-end manned and unmanned civil and military aircraft segments possible. These segments are subject to other boundary conditions regarding costs, complexity, size, weight, and power consumption, as well as safety requirements, compared to commercial airliners, which rely on safety, reliability, and integrity through extensive redundancy and dissimilarity in both hardware and software, with associated costly and time-consuming development processes and tools. Affordable hardware together with modern model-based system and software development methods and tools allow rapid prototyping and early validation of system and algorithm design and tailoring to the specific needs of the lower-end segment, without the need to transfer high-end system designs and mitigation strategies.

With increased automation of GA and unmanned aircraft, and increasing control authority and bandwidth, new challenges arise. Classical automatic flight path control structures, with independent Single-Input/Single-Output (SISO) control of each flight path variable, are inherently prone to dangerous control objective conflicts, and energy mismanagement. Tracking of arbitrary speed and flight path targets is impossible with saturated energy rate control, i.e., thrust, and arbitrary curvatures of vertical and lateral flight paths are precluded with saturated path perpendicular force control. From a control system perspective, GA autopilots have been characterized by low control authority, i.e., narrow permissible control surface deflection ranges, and low control bandwidths, i.e., limited control surface deflection rates, sufficient for the typical operational scenario with slow and segmented path changes, and large settling times. This has allowed sufficient time for the pilot to react to any unsafe conditions. The higher the bandwidth of the flight path controller, the more dangerous the decoupled control of each variable becomes, since the aircraft energy rate quickly redistributes, compromising the energy integrity of the aircraft, and risking a loss of control. Especially for unmanned applications, robust, high-authority, and high-bandwidth flight path control is of interest, fully exploiting the physical capabilities of the plant and the available control power, in order to allow more aggressive maneuvering and disturbance rejection, going beyond that of traditional low-bandwidth commercial autopilots. Hence, automatic control objective conflict resolutions and envelope protections are required, providing deterministic behavior and robustness against competing or senseless higher-level commands, and ensuring the energy integrity of the aircraft, without conservative margins, allowing transient maneuvering outside stationary maintainable performance limits.

Taking advantage of these opportunities and challenges, the Institute of Flight System Dynamics at the Technische Universität München (TUM-FSD) has developed a highly modular Flight Guidance and Control System (FGCS), principally illustrated in Figure 1.1. The modular FGCS is designed to be easily portable to new application platforms, providing the full or partial range of typical and beyond state-of-the-art automated flight functionalities for a wide variety of aircraft, including full envelope, high-authority, high-bandwidth flight path control.

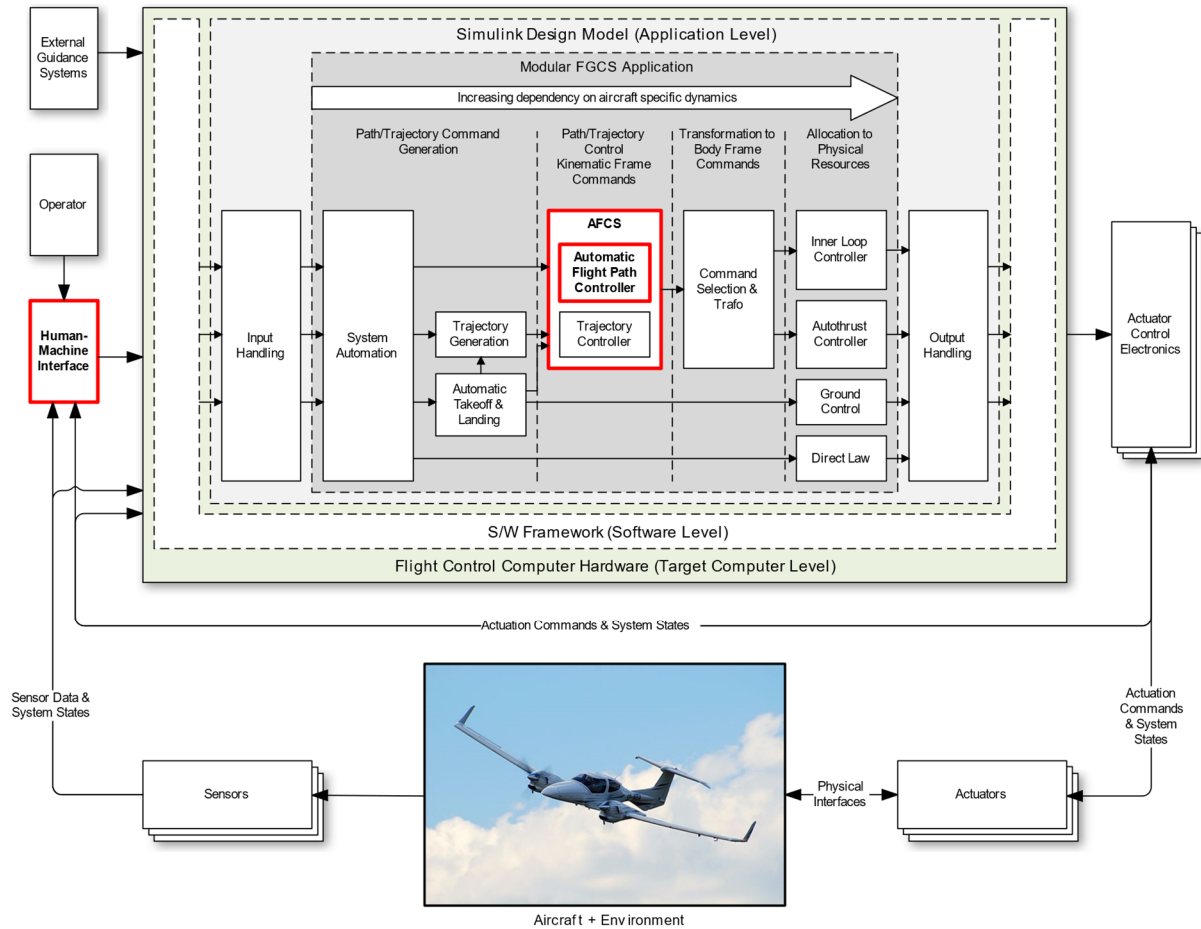


Figure 1.1: Modular FGCS overview with the main contributions of the author and the scope of this thesis (the development of the AFCS module with the automatic flight path controller, and the HMI) highlighted in red.

The research at TUM-FSD focuses on the model-based development of functional algorithms for guidance, navigation, and control applications, model-based safety assessment as well as the development of safety-critical avionics system architectures and components. Together with small- and medium-sized companies, specializing in each of the key technological developments described above, through partnerships and collaborative research projects, a modular avionics platform has been developed, that includes flight control computers, data concentrator units, sensing, and actuation systems. For verification and validation of functional algorithms, equipment, and component technologies for manned and unmanned applications, the TUM-FSD has modified a Diamond Aircraft Industries DA42 M-NG (Multi-Purpose Platform, Next Generation) to serve as flying testbed, the DA-42 OE-FSD, see Section 1.4.1.

The thesis presents the design, development, safety assessment, and verification of the automatic flight control system (AFCS) part of the modular FGCS, with a specific focus on the flight path controller and the mode control and monitoring interfaces, which have been the main research focus of the author.

1.2 Literature Study / State-of-the-Art

This section discusses the state-of-the-art of the application, design, development, and safety aspects of modern automatic flight path control. The terminology used varies among literature, regulations, and standards. Section 1.2.1 gives an overview of commonly used terms, their general usage, and how they are used in this thesis. Section 1.2.2 addresses classical and modern design approaches for automatic flight control systems and discusses their respective advantages and drawbacks. Automatic flight control is a system-level function that is realized by a set of hardware and software components, implemented according to a certain system architecture. The automatic flight control algorithms, i.e., the control laws, are implemented on the software level, but require verification both on the software (for example correct implementation) and on the system level (for example correct behavior and adequate performance). Section 1.2.3 gives an overview of current regulations and standards for the development of civil automatic flight control systems on the system as well as software level. Section 1.2.4 discusses the safety challenges inherent to automation with increasing authority and bandwidth, and Section 1.2.5 gives a broader overview of systems engineering standards and handbooks of relevance for the development of flight control systems.

1.2.1 Terminology

Throughout certification regulations, means of compliance, standards, literature, and descriptions of operational systems, different overlapping and interchangeable terms and acronyms associated with automatic flight control systems are used, for example, *autopilot*, *automatic pilot*, *AFGCS*, and *auto flight system*. An overview of commonly used terms and their usage within the scope of this thesis is given in Table 1.1.

Table 1.1 Overview of common automatic flight control system terms.

Acronym	Definition	General usage	Usage in thesis
AFCS	Automatic Flight Control System	Generic term that is used interchangeably with AP, AFGCS, AFCS or AFS	Refers to the AFCS module and functions of the modular FGCS
AFGCS	Automatic Flight Guidance and Control System	Refers to an integrated system of autopilot, flight director, yaw damper, and autothrottle functions and functional equipment AFGCS is the term used in regulations like TSO-C198 and MOPS like DO-325 and DO-335	When referencing regulations and standards using the term or when referring to a generic AFGCS
AFS	Auto/Automatic Flight System	Refers to an integrated system of autopilot, flight director, and autothrottle functions and functional equipment AFS is the term used in the ATA numbering system (ATA	Not used

Acronym	Definition	General usage	Usage in thesis
		22 for Auto Flight), and thus used by, e.g., Airbus and Boeing to refer to their integrated systems for automatic flight	
AP	Autopilot or Automatic Pilot	Refers to functions and functional equipment for one-, two- or three-axis automatic control of the aircraft Together with for example flight director, yaw damper and autothrottle part of an AFGCS	Used in the generic sense, not referring to a specific flight control system, when referencing regulations and standards using the term, for example
ATHR or A/T	Autothrottle or Autothrust	Refers to functions and functional equipment for automatic control of throttle levers or thrust setting.	Refers to the ATHR module and functions of the modular FGCS
FCS	Flight Control System	Generic term that encompasses everything from control interceptors, computing equipment, software, cabling, physical linkage, and sensors to control surfaces May or may not include autopilot/AFCS functions	Used in the generic sense, and not referring to a specific flight control system or implementation
FD	Flight Director	Refers to functions and functional equipment for providing guidance to the pilot for manual flight, or an autopilot for automatic flight	Refers to vertical and lateral guidance cues on MCMD
FGCS	Flight Guidance and Control System	Generic term that refers to the integrated FCS and Flight Management System (FMS)	Refers to the functions and functional equipment of the modular FGCS developed at TUM-FSD, of which the AFCS is part

1.2.2 Automatic Flight Control – Manned and Unmanned Applications

The level of automation and augmentation in modern manned and unmanned aircraft has increased over time, as have the number and complexity of automation modes [1, 2]. This development has had both positive impacts on aviation, with increasing capabilities and efficiencies, as well as negative impacts, with many accidents attributed to automation errors, lack of mode awareness, or pilot understanding of automation [2].

GA aircraft typically have mechanical flight control systems. Commercial autopilot systems interact directly with the mechanical controls through actuation servos and interfacing linkages. The actuation servos are typically of low bandwidth and control authority, i.e., they only move

at limited rates and with limited authority of control surface deflections. They are also typically possible to overpower by the pilot in case of unintended movements, or physically disengaged using overload clutches. Common autopilot functions, such as attitude, vertical and lateral path control, altitude capture and hold, as well as radio and waypoint navigation, are typically designed as cascaded loops around attitude- or rate-based inner loop controllers. Examples of such generic, cascaded SISO autopilot control structures for the longitudinal and lateral motion can be found in the standard literature on aircraft control, e.g. [3, 4, 5].

A Fly-by-Wire (FBW) flight control system employs digital control of the control surface deflections via flight control computers, data buses, and actuation systems, without any mechanical linkages between stick/control column and control surfaces. The commanded control surface deflections may be a direct mapping from stick/control column to deflection, a so-called *direct law*, or include stability augmentation or automation functionalities into the algorithms that govern the control surface deflection. As there is no mechanical control system to move the control surfaces, an FBW flight control system will have full control authority and be able to move the control surfaces at full bandwidth. Authority and bandwidth may be software-limited for different functions (e.g. autopilot and direct law). FBW flight control systems can be found on modern commercial airliners, as well as unmanned aircraft. For example, detailed descriptions of the Airbus FBW autopilot modes and control laws are given in [3].

Traditional autopilots suffer from inherent flight path control objective conflict problems, as arbitrary flight path and speed targets cannot be maintained with band-limited or saturated energy rate control. This is usually mitigated by the low control authority and bandwidth, i.e., limited surface deflections and rates, and limited operational envelope, allowing the pilot to intervene well before a critical flight condition arises. The Total Energy Control System (TECS) approach, initially presented in [6, 7], and refined in [8, 9], was developed to counter the inherent control objective conflict problems and disadvantages of the decoupled path and speed control, by coupling the control of energy flow, i.e. thrust, and energy rate distribution, i.e. pitch, based on flight path angle and acceleration errors. Efficient decoupling of speed and flight path control, however, requires pitch and thrust dynamics of similar bandwidth; with thrust typically much slower than the inherent pitch dynamics, the intentionally low control bandwidth is sufficient for manned autopilot applications, as demonstrated in [7], but lesser so for higher-bandwidth dittos. Furthermore, the linear interpretation of the path dynamics is only suitable for a limited envelope.

With increasing operational envelope, control authority, and bandwidth, nonlinearities of the aircraft dynamics and subsystems become more visible, the controlled variables become strongly coupled, and command saturations and control objective conflicts arise. Although linear control approaches predominate the practical applications, especially for manned, civil automatic flight control systems, several nonlinear strategies have been proposed and tested for the control of highly maneuverable manned and unmanned aircraft. Nonlinear Dynamic Inversion (NDI), also known as feedback linearization, is a well-known method for the control of nonlinear systems in general [10, 11]. NDI has found many flight control applications, both for control of the faster aircraft rate and attitude dynamics [12, 13], especially of highly

maneuverable aircraft [14, 15], as well as the path dynamics [16], for both manned and unmanned aircraft. One major challenge when employing NDI for practical control applications has been the relatively low robustness with respect to uncertainties in the model and its parameters used for the inversion. This is the case, especially for the control of the faster moment dynamics, which is dependent on the correct estimation of aircraft parameters. Adaptive control approaches, which are outside the scope of this thesis, aim at countering those uncertain and changing aircraft parameters by online parameter estimation and control law updates. For the control of the path dynamics, where the translational equations of motions are well-known and independent of aircraft-specific parameters, the only uncertainties in the inversion come from the measurement of the flight path states and forces.

Many of the control designs employing NDI are reference model based, i.e., a command filter is used to produce the corresponding pseudo-control trajectories for desired linear input-output dynamics. Faster dynamics, e.g., actuation dynamics, is typically neglected in the inversion if the time scale separation between the loops is sufficient. Time scale separation implies that the outer loop reference models are sufficiently slow, thereby maintaining achievable reference values and thus minimizing actuation saturations and other nonlinearities from the reference state to plant response error signal, avoiding integrator windups or unbounded weights in adaptive control approaches. The Pseudo-Control Hedging (PCH) approach [17, 18], which has received much attention in the context of dynamic inversion-based flight control, was originally introduced as a method for hiding such actuation nonlinearities from the adaptive elements. PCH has later also been employed as a means of propagating limitations in the dynamics and reaction deficits of nested control loops to the respective next outer loop [19].

The flight path controller that is the topic of this thesis utilizes an acceleration- or specific force-based inner loop controller for directly commanding the desired path curvature, whereas traditional autopilots are rate- or attitude-based. Acceleration controllers are commonly used in missile applications [20], but their application to aircraft maneuver flight control is not as common. Concepts for such acceleration-based maneuver autopilots for UASs, where the aircraft is reduced to a point mass with a steerable acceleration vector, have been presented for example by Peddle [21] and Boyle et al [22]. The speed loop of the flight path controller analogously utilizes an autothrust inner loop controlling the acceleration along the flight path, and thus influencing the total energy state of the aircraft. Concepts for longitudinal acceleration and total energy control (for enhanced manual control), utilizing active thrust, spoiler, and speed brake control, have for example been proposed and evaluated by Schreiter et al [23].

Specific force-based inner loop control allows for high bandwidth disturbance rejection, where control action is taken before disturbances propagate into position, velocity, and attitude errors. High bandwidth inner loop control further allows for efficient robust control that hides aircraft-specific uncertainties from the path loop dynamics [19, 21]. However, with an inner loop controlling the body axis specific forces, i.e. $f_{z,B}$ and $f_{y,B}$ as well as the bank angle Φ , as measured by the gyros and accelerometers of an Attitude and Heading Reference System (AHRS), extensive transformations are necessary between the different coordinate frames in order to successfully control the flight path and the aerodynamic speed along the flight path (measured by the inertial reference system and air data sensors). Specific force sensors have

successfully been used for feedback control in both military and civilian aircraft applications; however, incomplete or linearized attitude compensation, high noise-to-signal ratio, and pick up of non-minimum phase control responses have been negative side effects [21].

1.2.3 Regulatory Framework and Guidance Material

The design, development, and life cycle management of automatic flight guidance and control systems for manned and unmanned applications underlie extensive certification regulations, and associated standards provide acceptable means of showing compliance with the regulations. The regulation of the development, production, and installation of systems consisting of electronic hardware and software for airborne applications has the goal of ensuring an acceptable level of safety in the usage of such systems. Depending on the desired functionalities of a system at hand, their possible failure conditions and effects on the aircraft, crew/operator, and passengers, as well as the size and type of the aircraft, the acceptable level of risk per failure condition is quantified. Standardized "best practice" development processes, issued by private organizations but jointly developed between industry and authorities, define an agreed-upon minimum level of planning, development, and verification activities, and serve as acceptable means of showing compliance with the regulations. Additional standards define minimum operational performance requirements for specific systems or components.

Figure 1.2 gives a principal overview of the regulatory structure in the United States and European Union for the certification of manned civil aircraft and systems. The regulating authorities of the United States and the European Union, the Federal Aviation Administration (FAA) and the European Aviation Safety Agency (EASA), respectively, are tasked by their executive bodies to (among other things) research, draft, implement, and monitor safety rules, give type-certification of aircraft and components, as well as approve of organizations involved in the design, manufacture and maintenance of aeronautical products [24]. In the United States, the FAA issues the first three volumes of Title 14 of the Code of Federal Regulations (CFR), together known as the Federal Aviation Regulations (FAR). The FAR are divided into parts dedicated to a specific topic, e.g. Part 21 - Certification Procedures for Products and Parts (procedures for obtaining type certificates, supplemental type certificates, production certificates, etc.) [25], Part 23 - Airworthiness Standards: Normal Category Airplanes (including most of the general aviation aircraft) [26], or Part 25 - Airworthiness Standards: Transport Category Airplanes (including most large commercial aircraft) [27]. EASA issues Certification Specifications (CSs), largely adopting the topics and numbering of the FAR codification, e.g. CS-23 Normal-Category Aeroplanes [28].¹ The certification procedures corresponding to FAR Part 21 are specified in a separate appendix to the Basic Regulation 748/2012 [29]

¹ In a joint 2016 reorganization and harmonization effort [119], the FAA and EASA have revised the Part 23 and CS-23 (Amendment 5) to performance-based regulations, with revised aircraft classifications. New, design-independent, and objective requirements replace the previous design-oriented paragraphs, which are instead included as acceptable means of compliance. Aircraft classes have changed from *normal*, *utility*, *aerobatic*, and *commuter category* airplanes with associated seats and Maximum Takeoff Weight (MTOW) limits, to just normal category airplanes, comprising all aircraft with up to 19 seats and MTOW of 8618 kg, with certification requirements instead dependent on seat-oriented *aircraft levels* and maximum speed-oriented *performance levels*.

(conveniently named Part 21). For any modifications, exemptions, or additions to or from the certification specifications for new or innovative designs or for capabilities that are not explicitly regarded in the regulations, so-called Special Conditions (SCs) are negotiated between the manufacturer and authority and captured in Certification Review Items (CRIs) if an Equivalent Level of Safety (ELOS) can be proven. Examples of SC are Special Condition: Static Directional, Lateral, and Longitudinal Stability and Low Energy Awareness [30] and Special Condition: Flight Envelope Protection [31], part of the Airbus A350 type certificate [32].

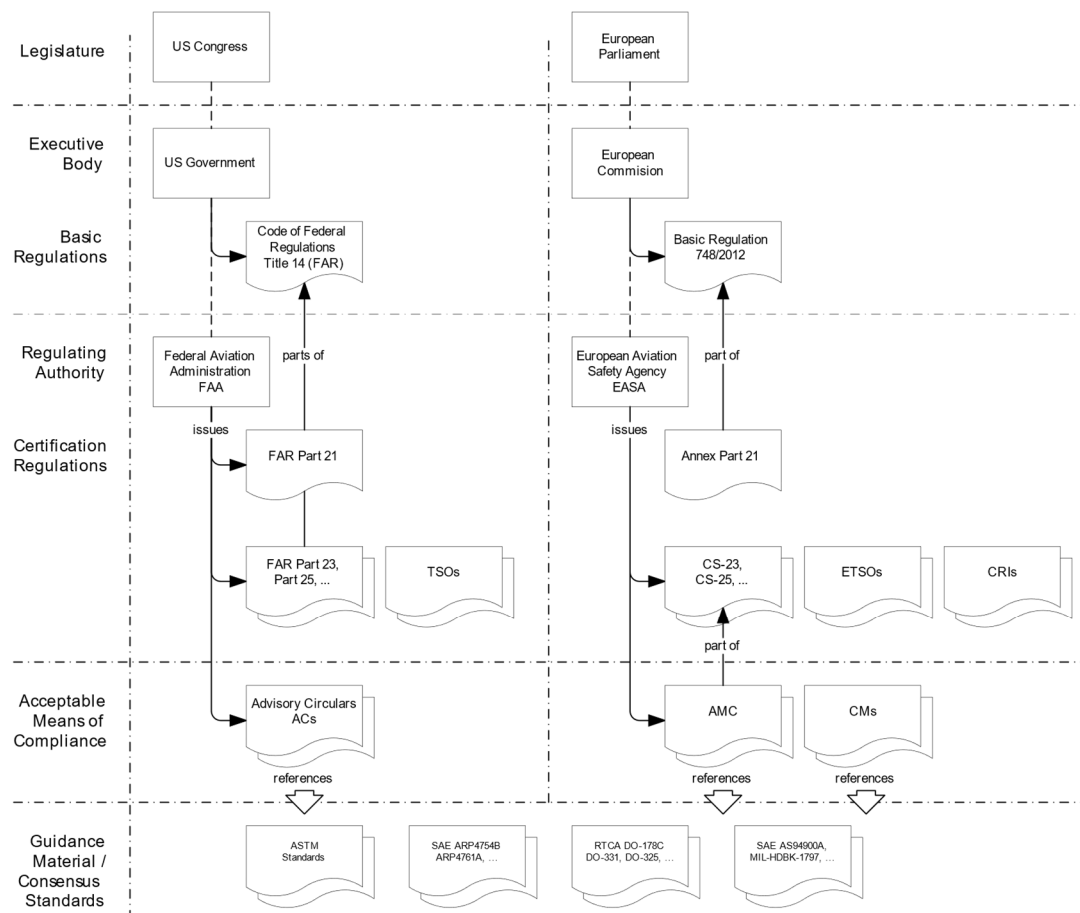


Figure 1.2: Regulatory structure in the United States and European Union for the certification of manned civil aircraft and systems.

Equipment, parts, or subsystems meeting certain Minimum Operational Performance Standards (MOPSS) can be approved for use in type-certified aircraft via a Technical Standard Order (TSO), issued by FAA, e.g. TSO-C198: Automatic Flight Guidance and Control System (AFGCS) Equipment [33], or an equivalent European Technical Standard Order (ETSO), issued by EASA, e.g. ETSO-C198 [34], referencing the DO-325 Minimum Operation Performance Standards (MOPS) for Automatic Flight Guidance and Control Systems and Equipment [35].

The applicable revisions of FARs, CSs, CRIs, TSOs, and possible additional guidance material are defined in a *certification basis* for the aircraft or system to be developed, which is included

in the type certificate. When a type certificate is temporarily invalid or not issued yet, a Permit-to-Fly (PTF) may be issued in order to operate the aircraft for specific approved flight conditions. A PTF is often issued for the operation of experimental aircraft, or for testing of aircraft with experimental systems installed.

Guidelines on how to interpret and meet regulatory requirements, so-called Acceptable Means of Compliance (AMC), are by the FAA issued as separate Advisory Circulars (ACs), e.g. AC 23-17C Systems and Equipment Guide for Certification of Part 23 Airplanes and Airships [36], and by the EASA as part of the specification, see Figure 1.2. The AMC in turn refer to industry consensus standards issued by standardization organizations, containing accepted best practices and guidance to show compliance with certification regulations.

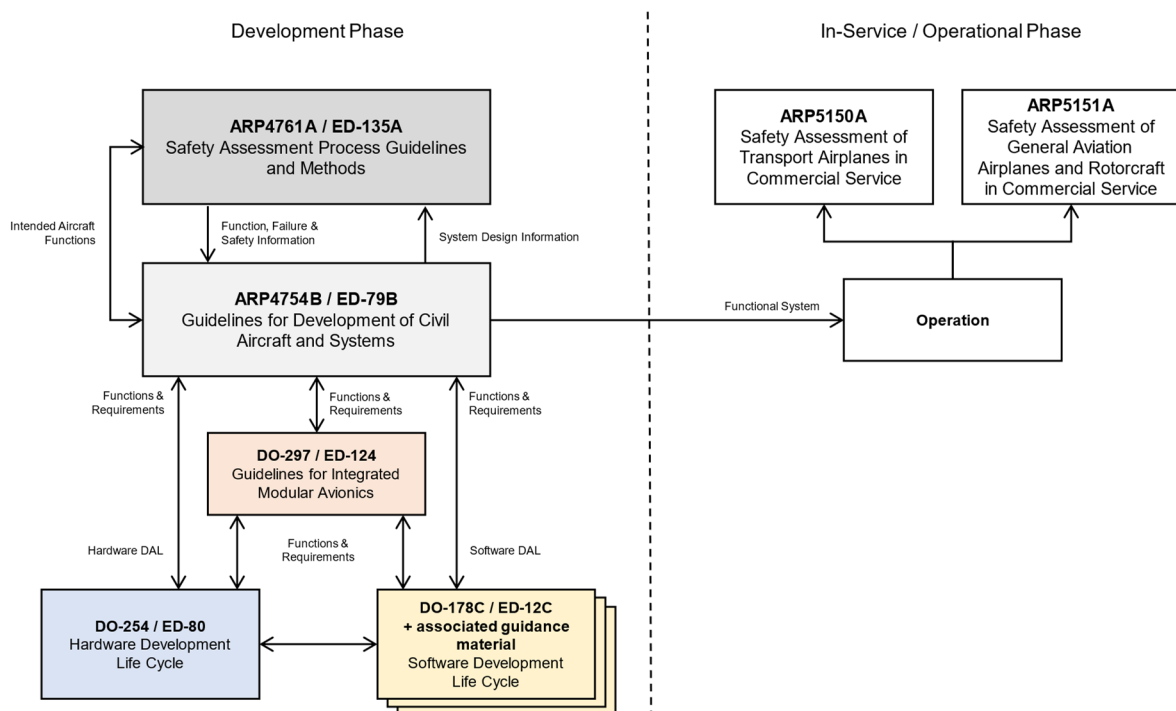


Figure 1.3: Interrelations of principal guidance material for civil aircraft and systems, their safety assessment, and hardware and software development.

The regulatory AMC identify several domain-specific standards for guidance on aircraft systems and software development. The main document providing guidance on the development of civil aircraft and systems is the Aerospace Recommended Practice (ARP) 4754 Guidelines for Development of Civil Aircraft and Systems, currently in revision B [37], issued by the Society of Automotive Engineers (SAE). The ARP4754B defines an overall development process model for aircraft and systems that implement aircraft functions. The development process model consists of a planning process, a set of sequential development processes, e.g., aircraft function development, system architecture development, and implementation, and a set of integral processes (i.e., parallel and coupled activities), e.g., safety assessment, configuration management, and process assurance processes. The safety assessment process and associated methods are further detailed in the related recommended

practice ARP4761A [38]. The software and electronic hardware processes are defined in the documents DO-178, revision C [39], DO-254 [40], and DO-160, revision G [41], all issued by RTCA Inc. Design guidance and certification considerations for integrated modular avionics are provided in RTCA document DO-297 [42], and guidance for in-service safety assessment of transport and general aviation airplanes as well as rotorcraft are found in ARP5150A [43] and ARP5151A [44], respectively. Standards related to the design and installation of automatic flight guidance and control systems include DO-335: Guidance and Installation of Automatic Flight Guidance and Control Systems (AFGCS) for Part 23 Airplanes [45], ARP419B Automatic Pilot Installations [46], and ARP5366A Autopilot, Flight Director and Autothrust Systems [47]. Many of the standards issued by RTCA and SAE are jointly developed together with the European Organisation for Civil Aviation Equipment (EUROCAE) and published as EUROCAE documents as well. The aforementioned guidance material and their relations are visualized in Figure 1.3, with their corresponding EUROCAE document numbers.

Modern software development techniques, such as model-based design and automatic code generation, have long been a regulatory limbo. The release of DO-178C and associated material clarified the use of several modern techniques within the aviation context, e.g., DO-330 addressing tool qualification [48], DO-331 addressing model-based development methods [49], DO-332 addressing object-oriented programming [50], as well as DO-333 addressing formal methods [51]. Rierson [52] provides a good overview of the relevant standards for the development of safety-critical software for aerospace applications. Rierson furthermore lists a number of potential benefits to model-based development and verification, including more focus on requirements and early validation, reduced development time and cost through automation and improved customer interaction, as well as growing support for additional verification tools and formal methods. However, Rierson also points out potential risks, such as the blurring of the traditionally well-defined system and software development processes and associated roles, the separation of specification and design, i.e., the mixing of "what" with "how", traceability between higher-level requirements and models, and uncertain certification credits from simulation.

1.2.4 Safety Assessment and Challenges in Automation Safety

Safety can be regarded as an emergent property of a system, i.e., not an inherent property of any of its components, but a characteristic of the integrated system operating in its intended environment. Modern aircraft are complex systems, with many software-intensive subsystems, feedback mechanisms, and human-machine interactions. Depending on the relations of the system components and the operational environment of the system, inadvertent component behavior or unforeseen interactions may cause unsafe system states. Every aircraft system is thus subject to an assessment of its safety.

The safety assessment process is a detailed analysis of a system's functional failure modes, their outcomes and associated failure conditions on an aircraft level, and subsequent classification of its functions depending on the severity of those conditions. The classification, combined with the type of aircraft, yields requirements on the system architecture, as well as the depth and rigor of development and verification activities for hardware and software, so-

called *development assurance levels* (DAL). For manned aircraft, failure conditions are classified based on their impact on crew, passengers, and aircraft. The safety assessment process and its relations to the overall development process are defined in ARP4754B [37], with ARP4761A [38] detailing the applicable analysis methods and tools.

The safety assessment process is based on analysis methods and tools such as Functional Hazard Assessment (FHA), Fault Tree Analysis (FTA), and Failure Modes and Effects Analysis (FMEA). As the system and automation complexity grow, traditional safety analysis based on these methods tends to lose in applicability, as many accidents are caused not by failing components (i.e., they were operating according to specification), but by unpredicted interactions between system elements, operators, and/or the system environment. With increasing levels and authority of flight control automation, especially in the GA and UAS segments, new challenges in automation safety arise. Many accidents can be attributed to missing safety considerations in the design and insufficient requirements. Lambregts [8] discusses some of these automation issues: airplane stall, i.e. no low-speed protection or awareness cues to the pilot; traditional decoupled automatic control with axis-by-axis control and overlapping modes; automation considered "non-flight critical", leaving to the flight crew to recognize and safely handle failures or inadvertent function of modes; partial automation, i.e. automatic control of flight path and manual control of airspeed, leading to a number of energy mismanagement incidents, see e.g. Asiana Airlines Flight 214 [53] and Turkish Airlines Flight 1951 [54]. Increased path control authority further allows meeting higher performance requirements for flight phases, or parts thereof, otherwise manually flown, such as approaches with lower decision heights. Alternative safety assessment approaches have been suggested, such as Levesons' System-Theoretic Process Analysis, STPA [55]. STPA assumes that accidents, in addition to component failures, can be caused by unsafe interactions of non-failing system components.

Display concepts and different mode control and monitoring strategies for increased automation and aircraft energy state awareness have received much research focus. The standard Primary Flight Display (PFD) layout with an artificial horizon, speed, altitude, and vertical speed tapes and limits, as well as Flight Path Vector (FPV) and energy angle indicators are well established, with the general design, symbology, colors, and formats regulated in AMC such as AC 23.1311C Installation of Electronic Display in Part 23 Airplanes [56], and industry standards such as ARP4102 Flight Deck Panels, Controls, and Displays [57], AS8008A Flight Director Equipment [58], and AS420B Flight Directors (Reciprocating Engine Powered Aircraft) [59].

A balance is required between a decluttered PFD for readability and necessary information density for sufficient situational and system state awareness. A too-unorthodox display concept would risk much energy required for interpretation at the cost of situational awareness. Evans et al. [60] evaluated different indicators for energy-related problems, such as conventional speed cues, synthetic vision with flight path vector, speed error, and acceleration cues, as well as aural alerts, with strong support for flight path-based symbology. Adami et al. [61] suggest a synthetic vision display concept for GA aircraft with enhanced flight path markers and a "tunnel in the sky", including potential flight path as an acceleration cue, and

maximum/minimum potential flight path for maximum/idle thrust to make the pilot aware of safety margins. Lambregts has in conjunction with the TECS concept developed a somewhat alternative *Energy Management PFD Concept* [62] that incorporates flight path acceleration and vertical speed together with appropriate scaling and formatting of indicators to visualize the aircraft energy error and rate, as well as the distribution of the energy rate.

1.2.5 Systems Engineering Standards and Handbooks

A plethora of standards and handbooks provide guidance and best practice on the development and life cycle management of complex, technical systems, but are not formally recognized as accepted means of compliance for civil aircraft and systems. However, they may still provide valuable guidance and best practice. Many of them are non-domain specific, providing high-level generic process frameworks for life cycle management of engineered systems, during conceptualization, development, production, operation, and support phases. A common framework for complex systems development and life cycle management is the ISO/IEC/IEEE 15288 [63], with a large collection of associated standards providing guidance on system-level planning [64], system requirements engineering [65], and system architecting [66]. At the software level, ISO/IEC/IEEE 12207 [67] and associated standards provide a corresponding life cycle framework. The International Council on Systems Engineering (INCOSE) provides a detailed handbook with additional guidance on ISO/IEC/IEEE 15288 and systems engineering in general [68].

Standards for technical documentation of aircraft systems provide a common documentation structure for operation and maintenance manuals, as well as related certification documents. ATA chapters [69] are a widely used structure, which has evolved for conventional commercial aircraft and typical systems. The ATA structure with its partitioning by system functions is less suitable for novel aircraft system architectures with highly integrated hardware and software functionalities. S1000D [70] is a non-domain-specific technical documentation standard for systems intended to provide a more flexible modular structure with information stored in data modules and its presentation generated for specific technical documentation purposes.

1.3 Scope, Research Objectives and Delimitations

The work presented in this thesis originated from the research question of how automatic flight path control functionalities can be designed to ensure a safe and deterministic control objective prioritization and protection of aircraft energy integrity for a broad range of operational scenarios for manned and unmanned, fixed-wing platforms of different sizes, envelopes, and regulatory environments.

1.3.1 Scope

Figure 1.1 illustrates the modular flight guidance and control system developed at TUM-FSD. The main responsibility and contribution of the author, as well as the scope of this thesis, has been the development of the following functional modules:

- the Automatic Flight Control System (AFCS) module which contains the flight path control modes as well as additional state-of-the-art autopilot control modes and associated mode control logic
- the Human-Machine-Interface (HMI) for AFCS mode control and monitoring, i.e., the design of a Mode Control Panel (MCP) and Mode Control and Monitoring Display (MCMD) for manual operation of the AFCS

Development activities include the definition of an appropriate development process, analysis of the applicable regulatory framework, specification, and safety assessment of desired functionalities, controller architecture development, implementation and integration into the overall application module, flight control system architecture, and system environment as well as verification and validation at increasing levels of system maturity, including flight testing.

The modules in focus have many interfaces to other parts of the overall flight control system; the interfaces and modules are described to an extent necessary for the understanding of the contributions of the author but are not described in detail. The delimitations and related works are further described in Section 1.3.3.

The work described in this thesis was mainly performed within the scope of one publicly funded research project (FLYSMART, LUFO-IV/4) and two application projects: the development and integration of an Experimental Digital Autopilot on the Dornier Do-228 D-CODE of the Deutsches Zentrum für Luft- und Raumfahrt (DLR) (led by RUAG Aerospace Services) and the SAGITTA Research Demonstrator (led by Airbus Defense and Space). The thesis focuses on the development work and validation of the AFCS concept within the FLYSMART research project. The DA42 OE-FSD serves as the main reference configuration for the presentation of system requirements, controller design, safety assessment, implementation, and integration as well as verification and validation throughout the thesis. Parts of the work related to the flight path controller design, control objective prioritization strategy, and initial flight test results presented in this thesis have previously been published in [71, 72, 73, 74].

1.3.2 Research Objectives

From the research question, a set of research objectives were defined, that the work presented in this thesis has concentrated on.

Objective 1: A modular control system architecture for reconfigurability and testability

The automatic flight path control functionalities shall enable flexible application to a broad range of application platforms. A modular controller architecture and distributed mode logic that allows configuration and adaptation of functionalities for new application platforms without extensive retesting is desired. The design principles and algorithms shall be straightforward, comprehensible, and reasonable. The controller architecture shall allow flexible integration of state-of-the-art automatic control modes to enable a broad range of operational scenarios and maximize the number of possible use cases of the system on different application platforms. The methodology is to be portable, and the modular architecture shall allow the easy configuration and verification of the desired functionalities, with retesting concentrated to application platform-specific functionalities and performance.

A modular architecture with controlled interfaces allows for the necessary concurrent development of the automatic flight path control functionalities with the development of the inner loop functionalities, trajectory controller, and higher-level system automation functionalities, as well as software framework.

Objective 2: High-authority automatic flight path control for dynamic mission profiles

The automatic flight path control functionalities shall enable a broad range of operational scenarios and mission profiles of manned and unmanned aircraft. For standard autopilot functionalities, limited control authority and bandwidth will suffice. For more dynamic mission profiles like reconnaissance patterns or steep departure or approach profiles, higher control authority and bandwidth will be required. The automatic flight path control functionalities shall meet operational and performance requirements over the entire operational envelope, i.e., the full speed and altitude range of the aircraft, by taking full advantage of the permissible load factor/bank angle/attitude envelope and fully utilizing the available bandwidth of the inner loop controller and propulsion systems over the envelope for maneuvering and disturbance rejection.

Objective 3: A certification-oriented design, implementation, and verification framework for real-world applications

The automatic flight path control functionalities shall enable easy configuration and adaptation of the controller structure and parameters for new application platforms from a broad range of manned and unmanned platforms of different sizes, envelopes, and regulatory environments. The controller design, implementation, and verification framework shall consider relevant system development process elements to support the qualifiability of the generated software and thus certifiability of the system, according to applicable certification regulations and development standards. Aspects of model-based software development shall be considered, in conjunction with requirements development and management, and their traceability to implementation and test cases, as well as verification on model level up to flight testing.

Important sub-objectives are the generic design and portability of the requirements and implementation structure to new aircraft configurations, and automation of parameter design and system analysis.

Objective 4: Dynamic control objective conflict resolution for “carefree” operation

A larger control bandwidth and actuation authority to allow for more dynamic mission profiles requires consideration of the flight path variable couplings in order to tackle arising control objective conflicts. The AFCS shall ensure safe and deterministic behavior, protection of the energy integrity of the aircraft, and robustness against non-achievable flight path commands. The AFCS shall also ensure smooth and achievable commands to the inner loops, without conservative margins. Higher-level automation functionalities or a manual operator shall be able to operate the automatic flight path control functionalities in a “carefree” manner, i.e., with an automatic and deterministic resolution of arising control object conflicts and ensuring envelope and energy integrity of the aircraft.

For the modular FGCS, a control objective prioritization and protection strategy according to Table 1.2 is desired. The control objective prioritization and protection strategy is such that the implementation of a prioritization or protection is located closer to the actuator commands the higher the priority, i.e., the angle of attack protection is downstream from the maneuver bandwidth prioritization and thus has the ability to override upstream commands.

Whenever a prioritization becomes active, the resulting limitation of control variables shall be smooth and transient-free.

Table 1.2: Control objective prioritization strategy for the modular FGCS.

Prio	Objective	Means	Protected Variables	Instance
1	Aerodynamic Integrity No Departure	Departure resistance by means of aerodynamic angle protections. Departure resistance more important than departure recovery capabilities (recovery not a considered function)	$\alpha_{min} \leq \alpha \leq \alpha_{max}$ $\beta_{min} \leq \beta \leq \beta_{max}$	Inner Loop
2	Structural Integrity No Overload	Limitation of load factor commands No overload unless temporarily required to ensure aerodynamic integrity	$n_{z,B,min} \leq n_{z,B} \leq n_{z,B,max}$ $n_{y,B,min} \leq n_{y,B} \leq n_{y,B,max}$	AFCS / Inner Loop Interface
3	Energy Integrity Stay in Envelope	Active limitation of thrust and flight path commands at edges of the envelope, in order to ensure airspeed limits	$V_{IAS,min} \leq V_{IAS} \leq V_{IAS,max}$	AFCS
4	Attitude Integrity No Upset Attitude	Possible to include for aircraft with limited nominal attitude range (i.e., passenger aircraft), for passenger and crew comfort and reduction of risk for departure Limitation of load factor command at edges of pitch attitude envelope Bank angle command limitation	$\theta_{min} \leq \theta \leq \theta_{max}$ $\Phi_{min} \leq \Phi \leq \Phi_{max}$	AFCS
5	Maneuver Integrity Maneuvering in Prioritized Control Plane	Energy rate saturation: Prioritization between path or speed tracking	$\gamma_{min,prio} \leq \gamma \leq \gamma_{max,prio}$ $\dot{V}_{min,prio} \leq \dot{V} \leq \dot{V}_{max,prio}$	AFCS
		Transverse force saturation: Prioritization between lateral or vertical plane path curvature	$\dot{\gamma}_{min,prio} \leq \dot{\gamma} \leq \dot{\gamma}_{max,prio}$ $\dot{\chi}_{min,prio} \leq \dot{\chi} \leq \dot{\chi}_{max,prio}$	AFCS

Objective 5: HMI with intuitive mode control and annunciation for high system awareness

Automated flight requires system state and behavior awareness. The automatic flight path control functionalities shall enable a broad range of operational scenarios and mission profiles of manned and unmanned aircraft, including direct command by higher-level system of flight management functionalities, as well as direct command inputs of desired targets and functions from a pilot or remote operator.

An HMI for AFCS mode control and annunciation shall be intuitive and configurable, avoid mode confusion, and support system state and behavior awareness during different stages of

flight control system testing. The HMI shall also allow adaptation and integration to application platform-specific environments.

1.3.3 Delimitations and Related Works

Development of a flight control system for real-world applications is a large undertaking and truly a team effort, and many colleagues and students at TUM-FSD have over several years contributed to the development and successful flight testing of the modular flight guidance and control system. Only through the contributions of all team members have the results in this thesis been made possible.

Summaries of, or references to, the following related works by team colleagues to the overall system development are included in the thesis for the understanding of the development and verification context of the AFCS but are not the principal work of the author.

Flight Guidance and Control Modules

The AFCS module has extensive interfaces to the trajectory controller [75, 76], trajectory planning and generation module [77, 78], as well as higher-level system automation [79, 80] and automatic takeoff and landing controller modules [81, 82, 83, 84]. The responsible colleagues and the author jointly developed those interfaces.

The platform-specific inner loop controllers and their interfaces, around which the flight path controller was validated in simulation and flight test, were developed specifically for the application platforms. The inner loop for the DA42 is described in [85]; this is also the inner loop design that was the basis for the Do-228. The SAGITTA inner loop design is presented in [86].

The Autothrust (ATHR) control module, which controls the acceleration along the longitudinal axis through the thrust setting of the aircraft (coupled with the control of the vertical plane), has been one of the developments of the author; however, this module is only briefly covered in this thesis.

Aircraft, Subsystem, and Environment Models

The AFCS design and verification were performed for multiple aircraft using different instantiations of a generic, high-fidelity Flight Dynamics Model (FDM), configurable for different aircraft, developed at TUM-FSD. The implementation and verification of the flight dynamics model, as well as environment models for static and dynamic atmosphere, and terrain, is described in [87].

The implementation and verification of models of different fidelity for inertial, air data, and other sensors, is also described in [87]. The actuation system is likewise modeled at different levels of fidelity. A high-fidelity, modular simulation model of the entire actuation system control chain including electro-mechanical actuators, mechanical systems, and actuation sensors, including effects of thermal behavior and stiff transitions for example in backlashes and clutches is described in [88].

Integration and Verification Environment

The integrated flight control system was initially verified and validated in model-in-the-loop desktop simulations. The main integration model for model-in-the-loop, hardware-in-the-loop, and subsequent flight testing was implemented mainly by Simon Schatz and Alexander Zollitsch, see [87]. A reduced integration model for verification and validation of the flight path controller, not containing interfacing control modules of which the flight path controller was not directly dependent on for implementation and verification, was implemented by the author and utilized for results presented in this thesis, see section 7.1.5. The hardware-in-the-loop simulation setup in the laboratory at TUM-FSD is presented in [89].

Aircraft Integration and Flight Testing

The presented AFCS was integrated and flight tested on multiple platforms. The principal reference configuration for the development and flight testing validation was the DA42 OE-FSD flying testbed. An overview of the testbed platform, aircraft integration activities, safety system concept, aircraft-in-the-loop simulation, flight test instrumentation, and experimental flight control system commissioning is given in [90]. The verification results included in this thesis are from this reference configuration, see section 8.1.

The AFCS was configured and integrated on the DLR Do-228 research aircraft D-CODE as an experimental autopilot system to support DLR research projects. The integration concept for the Do-228 D-CODE experimental autopilot is described in [91].

The AFCS was also integrated and flight tested on the unmanned SAGITTA Research Demonstrator. Configuration-specific system design aspects, aircraft integration, and first flight test activities are described in [92, 93].

Model-Based Software Development Environment

The AFCS design model implementation was performed following a detailed model-based software development process and modeling framework developed at TUM-FSD, see [94]. The verification of the algorithms was performed by the author at the design model level. The model-based software development environment at TUM-FSD, based on *MathWorks MATLAB/Simulink* tool suite, with automated code generation using the *Embedded Coder* from MathWorks and software integration aspects, are described in Hochstrasser et al. [94, 95].

Safety Assessment

The safety assessment described in this thesis concentrates on the system-level FHA part of the safety assessment process, where desired system functionalities and their potential hazards are analyzed, regardless of their implementation, and what that means for the development assurance of the system. The physical system architecture that implements the system functions is highly dependent on the application scenario and platform. The required functional and hardware/software development assurance levels resulting from the FHA may be achieved through different system architectures of varying levels of redundancy, dissimilarity, and independence. Research at TUM-FSD is focused on model-based approaches for safety assessment, with the simulation of nominal and failure behavior of

integrated avionics systems, software, hardware, sensors, and actuators, based on the MATLAB tool chain. This analysis does not require, but may be enhanced by, modeling of the system and its components in the presence of faults to predict system behavior. Modeling of candidate and implemented system architectures, components, and component failures with quantification of failure probabilities and their propagation to system-level failure effects, for example using FMEA or FTA is more of the focus in the System Safety Assessment (SSA) process, which is only briefly discussed in this thesis.

Both the function-centered, implementation-agnostic FHA and architecture-centered SSA are essential parts of the safety assessment process: the FHA generates correct system development and architecture safety and redundancy requirements, and the model-based analysis quantitatively assesses and verifies the required level of safety of the system as implemented. The different system architectures in the AFCS application platforms are discussed in the thesis, but a complete safety assessment is not part of this work.

1.4 Demonstration Platforms

The AFCS that is the topic of this thesis was developed and validated with the Diamond DA42 OE-FSD flying testbed as the reference platform. The AFCS has furthermore been configured and integrated on additional platforms, as part of the Experimental Autopilot System on the DLR Dornier Do-228 D-CODE, and as part of the flight control system on the SAGITTA Research Demonstrator. This section introduces the three demonstration platforms and summarizes key technical data.

1.4.1 Diamond DA42 OE-FSD

The Diamond DA42 OE-FSD is a research aircraft platform owned by the TUM-FSD, based on the DA42 M-NG by Diamond Aircraft Industries. The DA42 is a Part 23 aircraft with a maximum take-off weight (MTOW) of under 2000 kg, a purely mechanical flight control system with pushrods for elevator and aileron control, cables to the rudder, and Bowden cables for trim surfaces, see technical data in Table 1.3.

The DA42 OE-FSD is equipped with a specially developed flight test instrumentation, with high-quality reference sensors such as a navigation grade inertial navigation system, multiple GNSS receivers, multiple air data booms, laser and radar altimeters, and data links. An experimental FBW system based on electromechanical rotary actuators back driving the existing mechanical controls enables direct control of elevator, aileron, rudder, throttles, and trim surfaces. A multi-stage safety system allows research and testing of experimental components and functional algorithms by ensuring a safe disconnect of the experimental FBW system and reversion to manual controls in the event of faults in the experimental system.

The first flight testing of the modular FGCS functionality, including the AFCS, was successfully performed in January 2016. The DA42 OE-FSD has since been used for flight testing of functional algorithms and various demonstration missions, including fully automatic takeoff and automatic landing without intervention by the safety pilot.



Figure 1.4: DA42 OE-FSD at Wiener Neustadt airport in January 2016 (photo by the author).

Table 1.3: Diamond DA42 OE-FSD technical data.

Diamond DA42 OE-FSD Technical Data	
Wingspan	13.55 m
Length	8.56 m
Height	2.49 m
Propulsion	2 x Austro Engine AE300 (123.5kW)
MTOW	1999 kg
Seats	3 (safety pilot in front left, test pilot in front right, and flight test engineer rear left)
Service ceiling	18,000 ft
Max. cruise speed	197 kts
Stall speed	62 kts (landing configuration)
Class	Part 23, Class II ²

² Class II according to AC 23.1309-1E [92] (Multiple Reciprocating Engine (MRE) at 6,000 pounds or less). Corresponds to level 2 low-speed aircraft (2-6 seats, max 250 kts CAS) according to the revised Part 23/CS-23 specifications, or normal category airplane (max. 9 seats, MTOW 5670 kg) according to the old Part 23/CS-23 specifications.

1.4.2 Dornier Do-228 D-CODE

The Dornier Do-228 D-CODE of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR), Figure 1.5, is a universal research aircraft that has been in operation for more than 30 years supporting a broad spectrum of research activities [96, 97]. The D-CODE is a twin-engine turboprop aircraft with a mechanical flight control system and a maximum take-off weight of 5980 kg, see technical data in Table 1.4.



Figure 1.5: The DLR Dornier Do-228 D-CODE prior to first flight tests in Oberpfaffenhofen, August 2016 (photo by the author).

Table 1.4: Dornier Do-228 D-CODE technical data (adopted from [97, 91]).

Dornier Do-228 D-CODE Technical Data	
Wingspan	16.97 m
Length	15.03 m
Height	4.86 m
Propulsion	2 x Garret AiResearch TPE 331-5-252D engines 533 kW, 5-bladed composite propellers
MTOW	5,980 kg
Seats	15 (fitted with nine seats for DLR research purposes)
Service ceiling	25,000 ft
Maximum speed	200 kts
Stall speed	74 kts
Class	Part 23, Class IV ³

³ Class IV according to AC 23.1309-1E [92] (commuter category aircraft). Corresponds to level 4 low-speed aircraft (10-19 seats, max 250 kts CAS) according to the revised Part 23/CS-23 specifications, or

The D-CODE has an analog three-axis autopilot (Bendix King KFC 250) installed that provides basic autopilot functionalities. However, the analog autopilot can only receive inputs from the built-in avionic systems and has no capability to receive or process commands from external guidance systems, such as autopilot modes and flight guidance from DLR's Experimental Flight Management System (EFMS). In order to provide an interface to the flight control system of the D-CODE, an *Experimental Autopilot System* was developed and installed in parallel to the analog autopilot [91]. The purpose of the remote-control capabilities is to enable the D-CODE to work as a demonstrator for Remotely Piloted Vehicle (RPV) research activities and RPV systems.

The Experimental Autopilot System comprises a Safety Relay Box (SRB), a Flight Control Computer (FCC), actuators, and Actuator Control Electronics (ACE) including clutches interfacing with the main control surfaces, as well as an interface to the electrical pitch trim system. The integration concept allows for the Experimental Autopilot System to be installed but electrically set inoperative under a Supplemental Type Certificate (STC) and activated and operated under a PTF for defined flight test conditions. This allows the D-CODE to operate unrestricted under its type certificate with the Experimental Autopilot System inactive, for example during ferry flights or sensor missions. Due to the experimental status of the Experimental Autopilot System, software modifications may be performed and flight tested without a full recertification. The first automatic flight with the Experimental Autopilot System was performed in August 2016.

The Experimental Autopilot System may be operated by a test pilot via the MCP or by the EFMS and monitored via the MCMD. The Experimental Autopilot System software consists of the AFCS module and the inner loop module of the modular FGCS, a dedicated pitch trim functionality, as well as application-specific input and output handling.

1.4.3 SAGITTA Research Demonstrator

The SAGITTA Research Demonstrator is part of the Airbus Defence and Space *Open Innovation* initiative and was developed in cooperation with universities and research institutes across Germany. SAGITTA is a downscaled, jet-powered UAV technology and capability demonstrator in a diamond-shaped flying wing configuration, with a maximum takeoff weight of 150 kg, and a length and wingspan of 3 m.

The operational concept includes a Flight Operator (FO) at a ground control station providing high-level commands and automation mode switching, for example, automatic takeoff and landing modes and waypoint navigation. An External Pilot (EP) serves as a backup, ready to fly the aircraft manually via low-level attitude control in case of a failure or unintended behavior of the automatic flight control system. The first flight took place in July 2017 at the Denel Overberg Test Range in South Africa [92].

commuter category airplane (max. 19 seats, MTOW 8618 kg) according to the old Part 23/CS-23 specifications.



Figure 1.6: SAGITTA Research Demonstrator mockup on display at TUM (photo by the author).

TUM-FSD was responsible for the development of the digital flight control system, from inner loops [86] to high-level system automation functionalities [80]. The AFCS provides medium-level control of flight path and airspeed, including energy protection. The nominal AFCS module is for the SAGITTA extended by a backup law with simplified control laws for reduced sensor set (only AHRS and air data sensors).

Table 1.5: SAGITTA Research Demonstrator technical data.

SAGITTA Research Demonstrator Technical Data	
Wingspan	3 m
Length	3 m
Propulsion	2 x 300 N turbines
MTOW	150 kg
Class	Research prototype

1.5 Contributions

This section describes the contributions of the thesis beyond the current state-of-the-art. They concern:

- Tailoring of system-level process elements from applicable guidance material for the purpose of automatic flight control system development
- The development, implementation, and flight test demonstration as an application example of NDI-based flight path control
- A concept for active control objective prioritization integrated into the flight path control architecture, demonstrated in simulation and flight test
- A customizable mode control and monitoring interface for flight control algorithm validation during development, as well as in-flight and remote control AFCS operation

1.5.1 Contribution 1: Tailored System Development Process

The first contribution is a tailored system-level implementation of a subset of the ARP4754B development process model for the purpose of a model-based flight control law development.

The proposed AFCS development process aligns with and complements the modular model-based software development process developed at TUM-FSD and described by Hochstrasser [98], which covers software-level aspects of DO-178C and its DO-331 supplement. The modular model-based software development approach [98] utilizes the DO-331 workflow where typical software-level activities relating to the control law development, such as software requirements and design are performed at the system level, through the direct implementation of a design model from system-level requirements. This workflow makes them subject to the ARP4754B processes which the AFCS development process addresses.

The system development process contributes with the following system-level perspectives:

- The definition of a generic, principal automatic flight control system architecture with generic architecture elements, which provides a structure for the definition of system functions and requirements
- An approach for the definition of sets of application-generic and application-specific system functions and requirements with references to applicable performance standards for automatic flight control systems
- A functional hazard assessment application, with references to applicable safety assessment standards for automatic flight control systems
- A set of system architecture considerations in an automatic flight control system context, with development assurance level assignment strategies based on three physical reference architectures representing different levels of functional and item independence

1.5.2 Contribution 2: NDI-Based Flight Path Control Application

The second contribution is an application example of a reference model architecture for NDI-based flight path control with design, implementation, and flight-test demonstration.

A coupled reference model architecture for control of the vertical and lateral kinematic flight path, as well as aerodynamic speed along the flight path, is presented. The basic theory of reference model-based NDI is not new, and many flight control applications have been studied. The contribution beyond the current state-of-the-art consists of the application of existing methods for automatic path control in a new context, more specifically the large scope of functional integration and application platforms, as well as experimental flight test results that further validate the application of the used methods.

The architecture allows dynamic coupling of the flight path reference dynamics for control objective prioritization and additional control mode integration, to suit a large number of possible use cases of the system on different application platforms, in terms of functions, envelope, and maneuvering bandwidth. In contrast to typical limited authority autopilots the presented architecture utilizes the full maneuvering potential of the plant over the envelope. The full envelope and maneuvering bandwidth of the flight path controller is made possible by:

- Utilizing the full available performance of a specific force-based inner loop controller, by directly commanding the equivalent path curvature, independent of the aircraft attitude
- Decoupling of the flight path tracking and error dynamics through the use of second-order reference models for the flight path states, while ensuring smooth and transient-free pseudo control commands to the inner loops
- Not making the reference models unnecessarily slow by adhering to time scale separation criteria and worst-case inner loop actuation dynamics but designing the reference model time constants based on available inner loop performance over the envelope and using a pseudo-control hedging approach to form a closed loop reference dynamics accounting for the actual inner loop dynamics
- Controller stability and robustness assessments explicitly considering the non-standard feedback structures arising with active pseudo-control hedging and active control objective prioritizations

The coupled architecture with cross-feeds and internal limiters allows for the easy integration and configuration of control objective prioritization strategies and additional control modes such as speed by pitch, flare, and direct flight path rate modes.

1.5.3 Contribution 3: Active Control Objective Prioritization Concept

The third contribution is the design, implementation, and flight test demonstration of an active control objective prioritization concept based on saturated energy rate and force control constraints.

The control objective prioritization concept is defined and integrated into the reference model architecture of the NDI-based flight path controller. Holzapfel et al. [19, 99, 100] lay out the basic principles for an energy-based prioritization of speed or flight path, as well as a force-based prioritization of vertical or lateral path curvature. The approach derives a set of separate constraints on the acceleration and flight path in order to achieve speed priority (flight path limited to achieve desired acceleration) or flight path priority (acceleration limited to achieve

desired flight path), as well as constraints on vertical and lateral curvature to enable maneuvering in the desired control plane.

This thesis presents an extension of the basic principles proposed in [19, 99, 100] beyond the binary speed versus path prioritization and vertical versus lateral control plane maneuvering to an integrated, mixable prioritization, with the energy rate distribution authority fully or partly assigned to speed or flight path control, and force distribution fully or partially assigned to the vertical or lateral control plane. The energy- and force-based prioritizations are implemented and integrated into an overall modular control application and demonstrated in simulations as well as flight tests, which constitutes a contribution beyond the state-of-the-art.

The dynamic control objective prioritization ensures deterministic, smooth, and transient-free interventions in case of energy or force control saturations, as well as at the edges of the envelope, without sacrificing maneuvering bandwidth at safe operating points. This goes beyond the state-of-the-art of integrated path control architectures, which typically employ statically coupled control of the flight path states or conservative margins.

The control objective prioritization strategy is made possible by:

- The use of reference models with cross-coupled internal limiters for desired flight path states and their derivatives
- Dynamic distribution of calculated energy rate and path perpendicular force budgets, in order to momentarily
 - fully prioritize vertical path or speed control, by prioritizing acceleration along or inclination of the flight path, or a specific acceleration and flight path angle combination
 - fully prioritize maneuvering in the vertical or lateral plane, or some combined maneuvering plane priority
- Automatically prioritizing airspeed at the edges of the envelope, through enforcement of dynamic speed/acceleration phase-plane-based acceleration, flight path angle and throttle limits, in order to ensure aircraft energy integrity

The control objective prioritizations act as protection against adverse input command combinations, reducing the required command plausibility efforts at higher-level control loops or system automation as well as ensuring smooth and trackable output commands to the inner loop controllers, i.e. up- and downstream "carefree" operation.

1.5.4 Contribution 4: Mode Control and Monitoring Interface

The fourth contribution is the development of a compact, single-display mode control and monitoring interface, consisting of a Mode Control Panel (MCP) and a Mode Control and Monitoring Display (MCMD), for in-flight AFCS operation, remote pilot control, and state awareness of the experimental flight control system. The contribution beyond the state-of-the-art is the architecture and design of the MCP and MCMD to allow configuration, installation, and operation on multiple application platforms, as well as the experimental validation of the MCP and MCMD in flight test and remote operation.

The MCP and MCMD layouts share design principles with and combine elements of typical state-of-the-art autopilot mode control interfaces, Primary Flight Displays (PFD), and system status monitoring found in airliners and commercial avionics suites for general aviation aircraft. This is important to ensure familiarity and intuitive handling for test pilots and operators. The MCP and MCMD, however, include design elements, information density, and form factors driven by the experimental nature of the flight control system.

The compact single panel, single display design allows MCP and MCMD installation into an application aircraft for test pilot control of the AFCS functionalities in-flight, as demonstrated on the Do-228 D-CODE, or configuration as a UAV or OPV ground station providing a remote operator with AFCS control and awareness via data link, as demonstrated on the DA42 OE-FSD.

MCP and MCMD functions and elements beyond state-of-the-art of comparable commercial and experimental systems include:

- Dynamic indications of flight path limits and targets when control objective prioritization or protections become active
- Separated target selection, synchronization and confirmation logic and annunciation to increase target selection awareness
- Individual engagement/disengagement control and status of actuation resources for each axis, together with control surface commands and current deflections

The interface layout and functionality were initially developed as executable desktop applications, interfacing with the model-in-the-loop simulation environment, greatly supporting control law prototyping and early validation of system operation. The design was iteratively improved based on test pilot feedback before hardware manufacturing and hardware-in-the-loop integration. Display software and mode control logic were fine-tuned in the laboratory setup and aircraft integration environments and validated in flight tests and remote-controlled operation.

1.6 Thesis Structure

The thesis structure is illustrated in Figure 1.7. After the introductory chapter, the development process, flight dynamics, and control theory preliminaries are given over three chapters.

- Chapter 2 describes the AFCS development process in relation to the regulatory framework, applicable regulations, and acceptable means of compliance, specifically the ARP4754B and DO-178C, and the intertwining of the system and model-based software development processes.
- Chapter 3 provides the necessary flight system dynamics preliminaries, including the derivation and analysis of the aircraft equations of motion, with a focus on the flight path dynamics. Further, the energy and force constraints causing the typical control path control objective conflicts are analyzed in detail.
- Chapter 4 presents the control theory preliminaries upon which the controller is designed. The theory of dynamic inversion is briefly presented, including reference models, error dynamics and error control, pseudo-control hedging, and aspects of

stability and performance of systems with uncertainties subsequently used for robustness analysis of the controller.

The development activities that comprise the main work of the thesis are described in the subsequent four chapters, covering the AFCS requirements, design, safety assessment, implementation, and verification.

- Chapter 5 describes the AFCS system definition:
 - The elicitation of the needs, design objectives, and system requirements of the controller
 - The design of the controller architecture, i.e., the control laws and mode control logic designed to meet the requirements
 - The design of the controller gains and parameters
 - The design of the HMI, i.e., the Mode Control Panel and the Mode Control and Monitoring Display of the AFCS
- Chapter 6 describes the AFCS safety assessment activities, and the resulting safety-related requirements and design constraints.
- Chapter 7 describes the AFCS system realization, i.e., the implementation of the AFCS design model, and its integration into the overall flight control system application software, hardware architecture, and aircraft.
- Chapter 8 presents the AFCS verification and validation and the successive levels of prototype to final system flight testing, and summarizes the verification and validation results, demonstrating the validity of the selected approach.

Chapter 9 provides a concluding discussion of the results and an outlook.

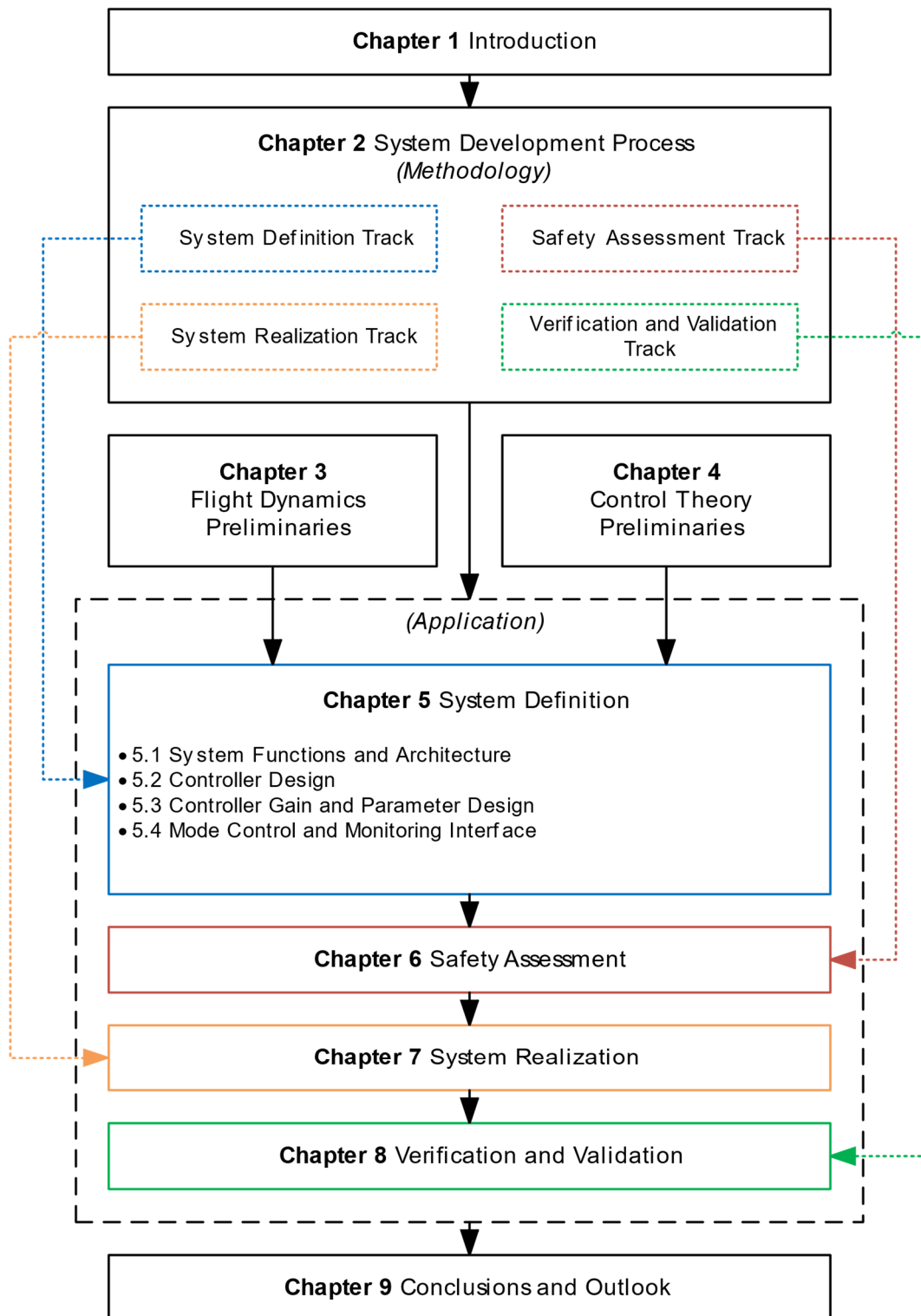


Figure 1.7: Thesis structure and chapters.

2 System Development Process

This chapter describes a system development process defined and implemented for the AFCS. Development of a flight control system, including the application software, i.e., the flight control algorithms and logic, is a chain of sequential and iterative activities both at the system and at the software level. The introductory literature study gives an overview of the main applicable regulations, guidance material, and development standards for flight control systems in general, and for autopilots in particular, see section 1.2.3.

The AFCS development process introduces a tailored system-level implementation of a subset of the ARP4754B development process model for the purpose of a model-based flight control law development. It contributes with system-level perspectives on requirements, functional safety assessment, system architecture, and development assurance levels and thus complements the modular model-based software development process developed at TUM-FSD and described by Hochstrasser [98], which covers software level aspects of DO-178C and its DO-331 supplement. The modular model-based software development approach takes advantage of the DO-331 workflow where typical software-level activities relating to the control law development, such as software requirements and design are performed at the system level, through the direct implementation of a design model from system-level requirements. This workflow makes them subject to ARP4754B processes which the AFCS development process addresses. Software-level aspects of DO-178C and its DO-331 supplement such as automatic source code generation from the design model, integration and verification of the software framework, and software coding and requirements standards are only briefly covered in the thesis.

The subsequent chapters of the thesis focus on parts of one or more of the AFCS development process elements. This chapter shall give an understanding of their role and relation to each other, as well as help the reader follow the "thread" through the thesis.

This chapter is organized as follows. First, Section 2.1.1 discusses the system-of-interest for the development process. Thereafter, in Section 2.1.2 the principal system life cycle models and their relations for the AFCS development and its applications are presented, together with a discussion on the relation between the life cycle model and life cycle processes. The applicable system- and software-level guidance materials are presented in more detail, with elements relevant for the AFCS development process. In section 2.3, the developed AFCS development process and its main activities are presented, with the interrelations of the

standards and their scope. The process elements and activities are also related to the corresponding chapter of the thesis. Section 2.3.3 describes the safety assessment process. The safety assessment process is an integral process with many interactions with the system development activities.

2.1 Development Process Context

2.1.1 Modular Flight Guidance and Control System

The modular FGCS developed at TUM-FSD and introduced in Section 1.1 can be viewed from a structural as well as a functional perspective. A structural view considers the systems, subsystems, and components in a hierarchy of system levels, see Figure 2.1. A functional view focuses on the system behavior and breaks down functions into functional blocks and information flows.

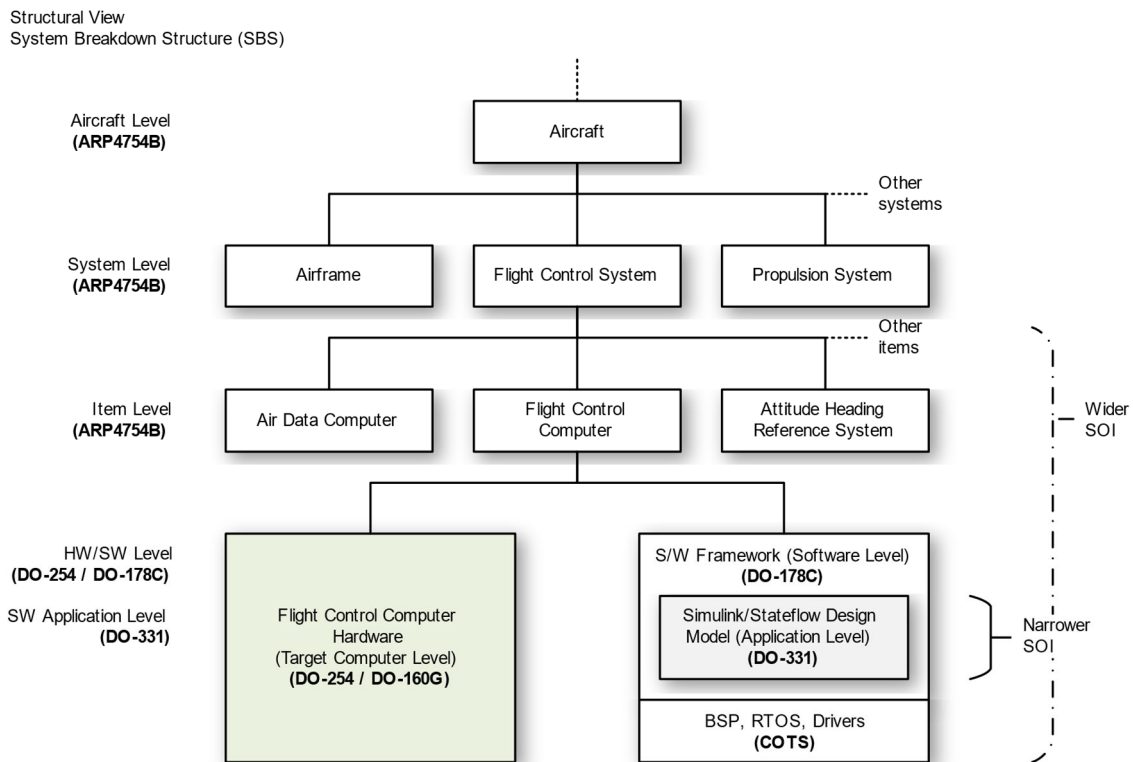


Figure 2.1: A hierarchical view of the modular FGCS, with principal system levels and the narrower and wider SOI indicated.

Automatic flight can be regarded as an aircraft-level function allocated to an automatic flight control system, which consists of multiple items, such as flight control computer, sensors, and actuation components. System-level functions for an automatic flight control system typically include automatic control of the vertical or lateral flight paths, flight director functions, mode control functions, and engagement/disengagement functions. The flight control algorithms for the specific automatic control functions that govern the aircraft behavior are implemented as software functions on the item level.

Viewed from a structural perspective, the modular FGCS may be divided into multiple levels, illustrated in Figure 2.1:

- *Application level*, which encompasses the flight control algorithms and their interfaces, such as the AFCS module, inner loop controller, trajectory controller, system automation, sensor filters, and monitoring functions
- *Software level*, which includes the application software, the software framework, and the hardware-close drivers and interface handling
- *Item level (target computer or hardware-software integration level)* is where the software is deployed on the embedded hardware, with a microcontroller, physical interfaces etc. affecting computing performance and accuracy
- *System level*, which includes sensors, mechanical interfaces, and actuation system properties
- *Aircraft level*, which includes the airframe and the aircraft dynamics, propulsion system dynamics, and their interactions with the atmosphere and ground

The structural perspective may be extended to higher system levels, including ground stations, data links and other systems interacting or interfacing with the aircraft.

Software Components

The flight control algorithms are implemented within the software application, following the model-based software development process according to DO-331 outlined by Hochstrasser [98]. The application software is embedded in a handwritten software application framework, developed in a conventional manner according to DO-178C. The software application framework design philosophy and implementation details are described by Nürnberger and Hochstrasser in [89].

Additional hardware-close software components like the Real-Time Operating System (RTOS), Board Support Package (BSP), and device drivers are COTS products, provided with the necessary documentation and artifacts to support qualification and application in a safety-critical context.

Flight Control Computer Hardware

The Flight Control Computer (FCC) is the target hardware for the developed software. The FCC has been specified by TUM-FSD and designed and manufactured by AEE Aircraft Electronics Engineering GmbH in close cooperation with TUM-FSD. The FCC was developed under consideration of applicable standards for airborne hardware, DO-254 [40], and environmental conditions and test procedures, DO-160 [41].

The FCC design philosophy and architecture are detailed by Nürnberger and Hochstrasser in [89]. The design objective was a platform that provides high computational performance and a large number of various digital interfaces common in avionics systems, in order to be adaptable and configurable for a larger number of aircraft application scenarios and flight control system integration environments. Thus, a multi-processor architecture was chosen. Two ARM Cortex-

M3 processors provide the external interfaces (referred to as the input-output or I/O processors). Each I/O processor provides the following external interfaces:

- 2 ARINC 825 CAN bus interfaces
- 4 serial interfaces
- 1 ARINC 429 input and 1 ARINC 429 output

An MPC8349 serves as the main processor executing the flight control algorithms. The main processor communicates with the I/O processors via Ethernet and discrete interfaces.

2.1.2 AFCS as System-of-Interest

System development activities are performed with respect to a specific “system,” defined by a set of interacting system elements, a system boundary, and a set of interfaces. The “system” under consideration is referred to as the *system-of-interest* (SOI) [63]. System elements may be regarded as systems in their own right, and every system is from some perspective part of a larger system or system-of-systems [68]. A flight control system consists of many interacting elements, such as sensors, actuators, computing hardware, software, and mechanical interfaces. Many of these elements may be regarded as complicated systems on their own and be further broken down into parts or subsystems. Development activities at different levels may concern different systems-of-interest – what constitutes the “system” in any given situation is a matter of definition. It is important that the definition is made in order to avoid confusion and contradicting views regarding what the “system” is, what the interfaces are, and what constitutes the system environment.

Narrower and Wider System-of-Interest

Flood and Carson [101] extend the concept of a SOI to a *narrower SOI* and a *wider SOI*. The narrower SOI is the system of direct concern of the observer, driven by the associated scope and authority of control. This scope may not capture all elements related to the narrower SOI. Thus, a wider SOI is defined with a logical system boundary that encompasses the elements necessary to understand the system behavior. The observer may not have the authority of control over all those elements but is able to establish and define the relationships and dependencies between the wider SOI and the narrower SOI.

For the purpose of the work presented in this thesis, the AFCS module at the control application level is regarded as the narrower SOI and is the subject of the presented development process. The modular FGCS at the control system level is regarded as the wider SOI.

System Environment

The modular FGCS was developed and validated with the DA42 OE-FSD as the reference platform, but with multiple other aircraft as candidate application platforms. With the aircraft and propulsion systems as the typical integration environment for the modular FGCS, the control system level is where the design degrees of freedom mainly exist, for example regarding computing, sensor, and actuation system architecture and associated development assurance requirements. Aircraft level and higher system levels are regarded as the system

environment, i.e., the physical and operational environment into which the control system is integrated, verified, and operated. For some application platforms, however, already present sensor and actuation systems would be utilized by the FGCS and thus rather be regarded as parts of the system environment. As stressed above, what constitutes the SOI in a given situation is a matter of definition.

2.1.3 AFCS and Application Life Cycles

Every SOI has a *life cycle*, i.e., different phases that the system progresses through, for example, a concept phase, development phase, and operational phase [63]. Life cycles look different for different systems. The life cycle of a given SOI may be described by a corresponding *life cycle model*, with defined life cycle phases. A life cycle model is a tool to facilitate a better understanding of the system, as well as give structure to and support system development and life cycle management activities. System maturation and the progression through the life cycle is supported by *life cycle processes*, for example, technical processes such as requirements, architecture, implementation, verification, and validation processes, and supporting processes such as configuration management. The life cycle phases are typically sequential, whereas the life cycle processes are overlapping with parallel activities. The context-specific scope, structure, and dependencies of the life cycle processes are given by applicable development standards and regulations, for example, ARP4754B [37] and DO-178C [39]. Standards and regulations provide principal life cycle models and process frameworks with generic descriptions of and requirements for activities and outcomes, which need to be tailored to the system-specific development assurance, level of complexity, and organizational aspects. The process models of the ARP4754B and DO-178C are described in Sections 2.2.1 and 2.2.2, respectively.

A principal life cycle model of the modular FGCS is illustrated in Figure 2.2. The life cycle is initiated when a basic stakeholder or mission need for a new system, or a system modification is identified. The life cycle starts with a concept phase where the system needs are further developed into an operational concept and a conceptual system solution emerges. The system environment is analyzed, and interfaces, installation requirements, and other boundary conditions for the system-of-interest are captured. The regulatory framework is analyzed, a suggested certification basis and means of compliance are defined, and the development phase is planned.

The development phase is the phase where the system is defined and realized; that is where system requirements and system architecture are developed, hardware and software are implemented and integrated, and the system is verified and validated, including flight testing. When the system is validated in its operational environment it is ready to support research activities as a flying testbed for novel functionalities, hardware, or operational scenarios. The system is continuously evolving with improved or added capabilities as part of or in order to support research activities. In parallel, the system is maintained and kept flight-worthy as per regulatory requirements. Once the system is obsolete, or unable to continue operation for some reason, the system is retired.

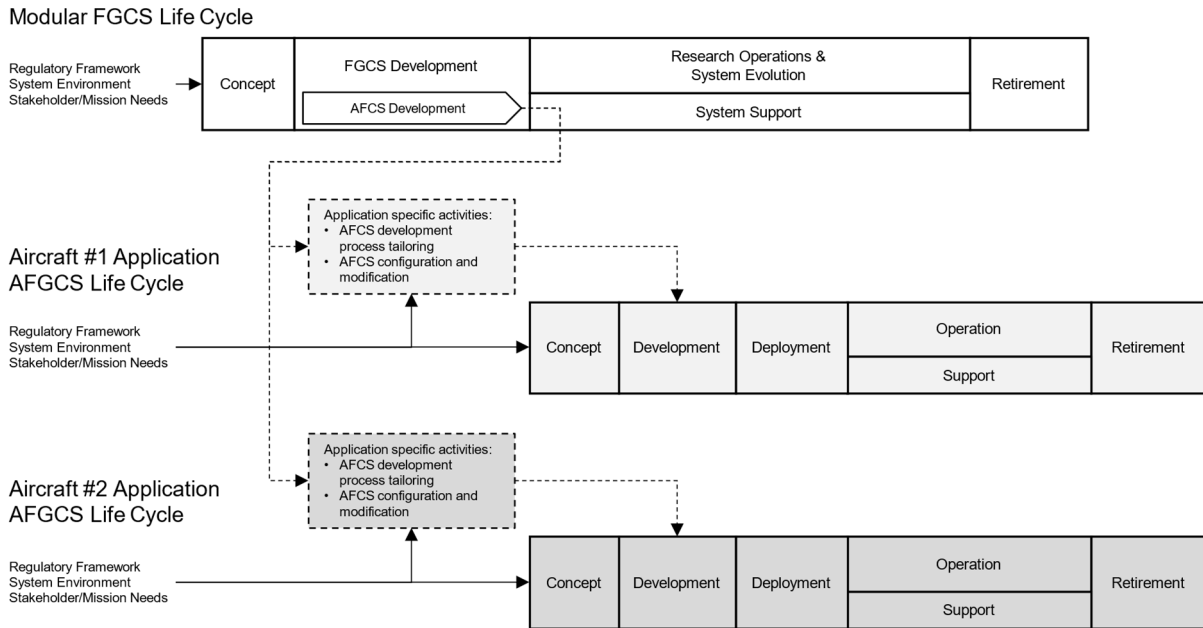


Figure 2.2: Principal Modular FGCS life cycle model with example AFGCS application life cycle models.

The modular FGCS is concurrently the basis for several application cases on other aircraft platforms, each with its own life cycle model. Figure 2.2 illustrates principal life cycle models for the applications.⁴ A similar concept phase identifies the needs and a conceptual system solution. A development phase details system requirements and system architecture, modifies and adds configuration-specific parts of the modular FGCS, implements, integrates, and verifies the final system. A deployment phase installs the system in its operational environment, performs final verification and validation activities, and obtains the necessary regulatory approvals before the system is transitioned to an operational state. The operational phase typically constitutes the main part of the life cycle, with a support phase where the system and its components are maintained, updated, and kept flight-worthy. At the end of its operational life, the system is retired, and its components perhaps recycled.

During the development, deployment, and operation of a flight control system, data is produced in order to describe the system and its functionality, for example, requirements and design descriptions, and to plan and collect evidence of activities performed, for example test plans and test results. This type of data is referred to as *life cycle data*. The life cycle data shall support software qualification and system certification, as well as future modifications of the developed system or software. The specific life cycle data to be produced for a given system

⁴ In Figure 2.2, the aircraft application life cycle models use the term *AFGCS* to denote the wider SOI subject to development. In this context, the *modular FGCS* refers to the specific system concept developed at TUM-FSD that is the basis for several aircraft applications, and the *AFCS* is one of the functional modules of the modular FGCS, see Figure 1.1. See Table 1.1 for a discussion of the various terms and their usage in this thesis.

is, similar to the specific life cycle process activities, determined by the applicable development standards and regulations, and what is agreed upon with the governing certification authority.

The thesis focuses on the development phase of the modular FGCS, illustrated in Figure 1.1, and the life cycle activities and life cycle data of the AFCS module, such as requirements, design model, test cases, and test procedures. The thesis briefly describes two application cases and discusses configuration-specific modifications and additions. The presented AFCS development process seeks to tailor and implement relevant subsets of the ARP4754B and DO178C/DO-331 to support certifiability of the AFCS concept. The AFCS development process also draws on best practice elements from the ISO/IEC/IEEE 15288 process framework, which is described in Section 2.2.3.

The objective of the thesis is to demonstrate a development and verification effort performed under the consideration of applicable system and software development standards and processes for "real world" applications, which is that the approach is suitable to support the certifiability of the conceptual solution. The AFCS, however, constitutes a subset of overall system functionalities and software implementation of the modular FGCS, and additional system and software level process elements and verification activities would be required to be tailored and implemented to enable qualification of an entire flight control software application and associated certification of an application system. A full set of life cycle data for certification of such a system is outside the scope of this thesis. Some of the system and software level life cycle data, such as generic plan documents, software model and requirements standards, model libraries, etc. have been produced as a team effort at TUM-FSD, and are briefly described or referenced, but not further detailed in this thesis.

2.1.4 Assumptions for the AFCS Development Process

The AFCS development process focuses on the AFCS functionalities as the narrow SOI, subject to the design and constraints of the flight control system as the wider SOI. The AFCS development process, as defined and implemented in the modular FGCS life cycle context, as well as for any aircraft-specific application life cycle, is subject to the following set of assumptions regarding the development of an overall AFGCS:

AS 1) Concept phase exists defining a basic operational concept, certification basis, and system environment

A concept phase may be:

- Part of the concept phase of the life cycle of a new aircraft design
- The concept phase for an AFGCS developed for integration into a legacy airframe

The concept phase is assumed to produce the following inputs to the AFCS development.

Certification Basis and Means of Compliance

A certification plan is the main plan to obtain agreement on with a certification authority before starting development, and should include:

- Certification basis for the system, specifying applicable certification regulations and guidance material
- A preliminary hazard assessment, identifying the dimensioning failure conditions and flight phases, and specifying the associated required functional development assurance level
- Outline of the means of showing compliance with the applicable certification regulations and guidance material

Operational Concept

The operational concept should capture the desired qualitative and quantitative functionalities, capabilities, and operational scenarios of the system to be developed from a user-oriented perspective. The operational concept serves as a common understanding of the system capabilities between the end user, developer, and other stakeholders and is the basis for the development of system requirements and the system architecture.

The operational concept should include:

- A description of the current system or situation, with justification for the proposed new or changed system, along with assumptions and constraints
- A conceptual description of the proposed system and its intended operational environment, modes of operation, user classes, and support environment
- Operational scenarios and mission profiles
- Analysis of impacts of the new system, benefits, disadvantages, and limitations

System Environment Description

The environment into which the system is to be integrated generates constraints that cannot be influenced by the system to be developed. These constraints shall be captured as external interface requirements and operational limits of the system. The system environment may be captured in the operational concept description, or in a separate, detailed System Environment Description (SED) document that serves as a common understanding between stakeholders of the technical constraints for the system development.

A SED should include descriptions of:

- The system-of-interest, system boundaries, and interfaces
- Aircraft flight dynamics properties such as aerodynamic data, mass and balance data, propulsion system performance per aircraft configuration
- Aircraft operational envelope, airspeed, and structural limits per aircraft configuration
- Interface requirements for subsystems
 - Sensors, including data bus formats with update frequency, resolution and accuracy, sensor installation points, software and hardware development assurance levels
 - Propulsion system
 - Mechanical flight controls and access points
- Available installation space and power for new components

- Integration strategy, for example, suitable subset integration, integration sequences, and critical dependencies

AS 2) Software development process is defined and compliant modeling blocks available

A DO-178C/DO-331-oriented process for model-based software development for AFCS and other functional module design model implementation, integration, and verification, as well as software framework development and integration, should include:

- Modeling standard and compliant block libraries
- Design model integration and verification activities
- Software requirements standard and software coding standard
- Design and implementation of software framework
- Automatic source code generation
- Software integration and verification activities

A detailed description of a modular software development process integrating model-based and traditional development according to DO-178C and DO-331 is given in [98].

AS 3) Process and tool for requirements capture and management exist

The AFCS Development Process puts focus on the elicitation and validation of system requirements from which the functional algorithms are developed and verified. The setup of a requirements management process and toolchain, with defined requirement and test artifact types, their attributes, and specific workflows is not within the scope of this thesis.

At TUM-FSD, *Polarion* was used as the tool for requirements capture and management. Exemplary workflows integrating *Polarion* into the modular model-based software development process is presented in [102].

AS 4) Process and tool for configuration management and change control exist

Development life cycle data such as system requirements, software components, and verification artifacts are considered configuration items subject to configuration and change control. The AFCS development process does not define a specific configuration management process implementation or tool for configuration control, as these would be project-specific.

For the modular FGCS development at TUM-FSD, *Git* has been the tool used for configuration control, with configuration items such as the AFCS module as separate *Git* repositories with local release control and automatic traceability between versions and baselines.

AS 5) Application aircraft-specific inner loop development exists or is developed concurrently with the AFCS

The AFCS is designed around an aircraft-specific inner loop controller that shall take the body axis normal specific force and bank angle as input commands, according to [85]. The inner loop development shall follow the same model-based software development process and use the same modeling standard to support design model integration, MILS and nonlinear

controller assessment, automatic source code generation, and hardware-software integration and testing.

The closed inner loop and actuation system dynamics shall exist as linear state space models over the envelope and aircraft configuration, preferably as generalized state space models with loop cuts implemented at the actuator commands to facilitate integration with the AFCS design and linear assessment toolchain.

2.2 System and Software Life Cycle Processes

2.2.1 ARP4754B Development Process Model

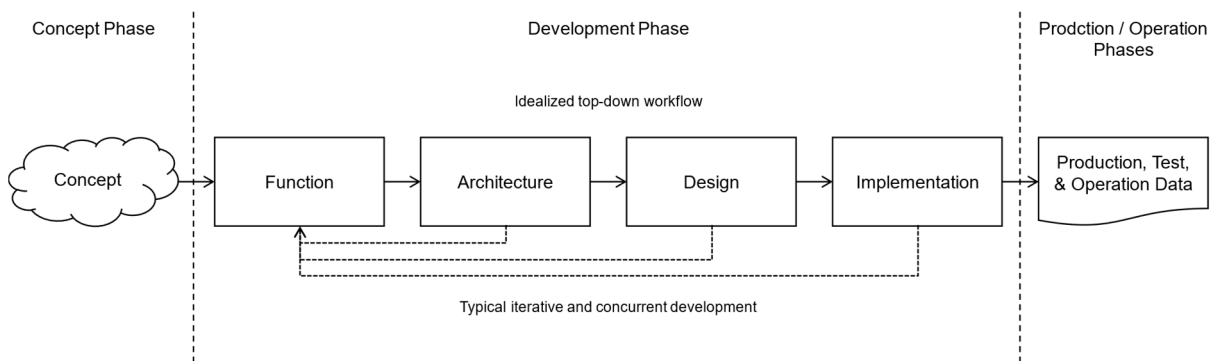


Figure 2.3: ARP4754B general life cycle model, adopted from [37].

The ARP4754B [37] is the principal guidance material on the development of civil aircraft and systems. It describes a general development life cycle, and an aircraft or system development process model, consisting of a planning process, a sequential, waterfall-like development process, and a set of integral processes that run parallel to and interact with the development process. The purpose of the process model is to provide a framework and guidance for process development, establish a common terminology, and set the expectations associated with systems development, not the definition of a normative process to be strictly adhered to. Neither does the ARP4754B imply a specific organizational structure. ARP4754B recognizes the typically iterative and concurrent nature of complex systems development, using both top-down and bottom-up strategies, and puts focus on the intent and applicability of the process model.

The ARP4754B development process is documented as a sequential waterfall model to illustrate the necessary links between aircraft safety and system development. The life cycle model is illustrated in Figure 2.3. The development process model is mainly concerned with the development phase of the life cycle model and provides no guidance on the conceptual design process for aircraft or systems, nor on the production or operation phases of the life cycle.

The ARP4754B defines objectives and a set of outputs for each process activity. The outputs are either recommended for certification, as negotiated, or not required, depending on the development assurance level and assigned one of two system control categories, SC1 or SC2.

These system control categories define the level of configuration management controls, for example, the establishment of baselines and configuration index, problem reporting, and change control, where SC1 requires more activities than SC2. The process model is illustrated in Figure 2.4.

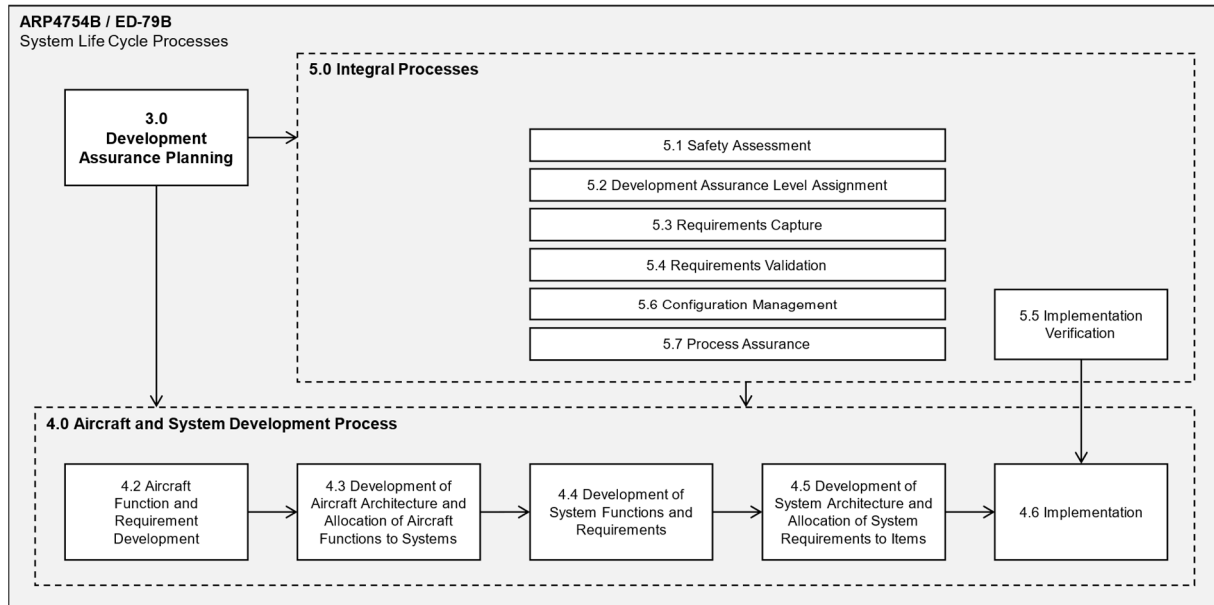


Figure 2.4: ARP4754B aircraft or system development process model, adopted from [37].

Development Assurance Planning

The planning process defines the means of producing an aircraft or system that will satisfy the aircraft and system requirements, and to provide the necessary level of confidence that airworthiness requirements are satisfied. The planning process defines a set of planning elements, rather than explicit plans, which corresponds to the development and integral processes of the process model: development, safety program, requirements management, validation, implementation verification, configuration management, process assurance, and certification. The planning elements may be captured in various formats such as integrated schedules or formally released planning documents.

The main plan to obtain agreement on with a certification authority before starting development would be a *Certification Plan*, defining the certification basis for the aircraft or system, a preliminary hazard assessment, and outlining the means of showing compliance with applicable certification regulations and guidance material.

Aircraft and System Development Process

The development process is described as a waterfall like sequence of activities from aircraft level down to item level. The process includes in a top-down manner the following activities:

- Aircraft Function and Requirement Development
- Development of Aircraft Architecture and Allocation of Aircraft Functions to Systems

- Development of System Functions and Requirements
- Development of System Architecture and Allocation of System Requirements to Items
- Implementation

Automatic Flight Control is mentioned as an example of an aircraft-level function. The aircraft-level functions are allocated to specific systems, for which candidate system architectures are developed and iteratively evaluated using functional and performance analyses, including safety assessment. The system architecture defines the structure and boundaries for the set of hardware and software items that implement the system functions to meet the established requirements.

Integral Processes

The integral processes are recursive in the sense that they apply parallel to the development processes on each level of aircraft function development, system development, and sub-system development. At the item level, the corresponding hardware and software life cycle processes apply, see Section 2.2.2. The integral processes are:

- Safety Assessment
- Development Assurance Level Assignment
- Requirements Capture
- Requirements Validation
- Implementation Verification
- Configuration Management
- Process Assurance

The requirements capture process is as mentioned above tightly coupled with the development process elements. The AFCS development implements process elements from the safety assessment, development assurance level assignment, requirements capture, and implementation verification processes, which are described in the following sections.

Safety Assessment

The goal of the safety assessment process, illustrated in Figure 2.5, is to show compliance with safety-related certification requirements, such as FAR Part 23, section 23.2510.

The Safety Assessment process consists of subprocesses at the aircraft and system levels:

- Aircraft-level Functional Hazard Assessment (FHA)
- Preliminary Aircraft Safety Assessment (PASA)
- Aircraft Safety Assessment (ASA)
- System-level FHA
- Preliminary System Safety Assessments (PSSA)
- System Safety Assessments (SSA)

Before a detailed safety assessment is developed, an FHA determines the depth and scope of the subsequent analysis. The FHA is performed in order to identify system hazards so that the necessary level of safety can be determined. The level of safety is determined by the

functions of the system and the effects of their failure modes on the aircraft, flight crew, occupants, environment, etc. An FHA may be performed on multiple levels, considering and classifying failure conditions associated with the functions of the specific system or subsystem level. Depending on the classification of the failure conditions, the subsequent analysis may require only qualitative or also quantitative assessments.

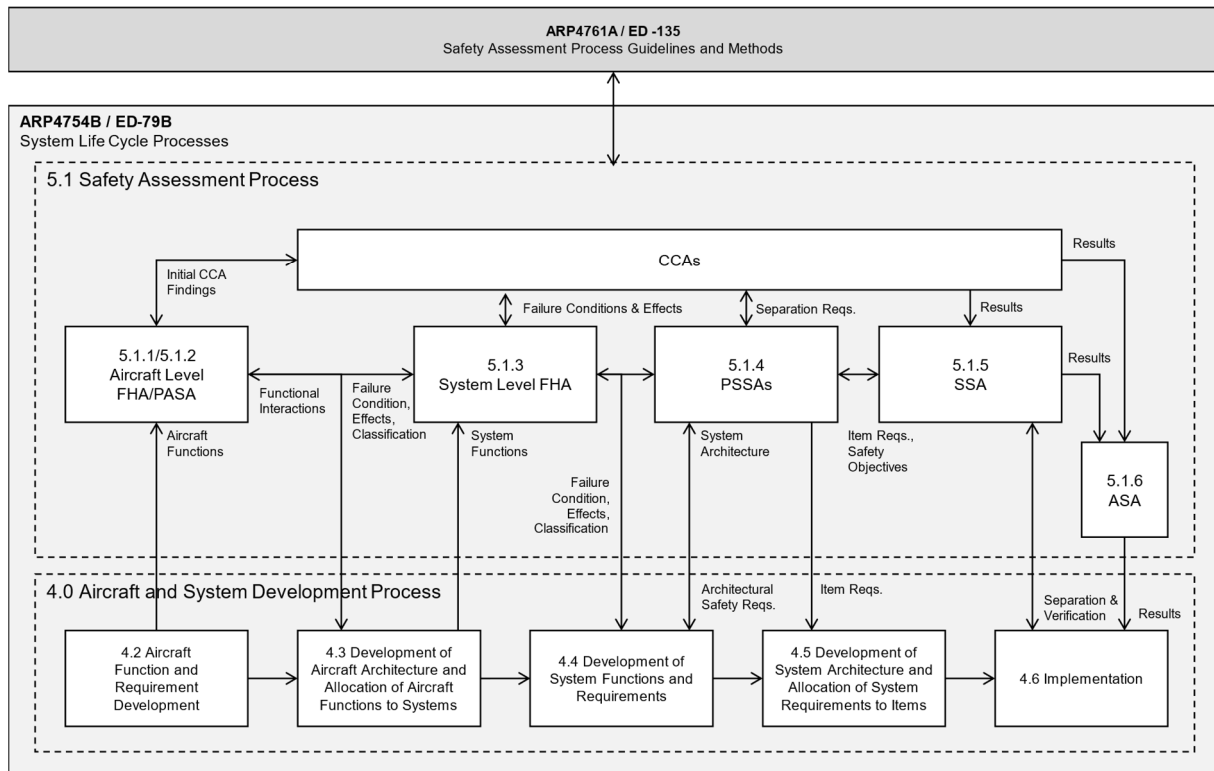


Figure 2.5: ARP4754B safety assessment process and interactions with the development process, adopted from [37].

Failure conditions are classified according to the effect on airplane, occupants, and flight crew, with the effect described as no safety effect, minor, major, hazardous, or catastrophic. Failure condition classifications are associated with qualitative probability requirements according to their assessed effects. The qualitative probability is quantified differently depending on the aircraft category.

The presented AFCS development process and associated safety assessment activities cover system requirements and system FHA process elements.

Development Assurance Level Assignment

The safety assessment process analyses the aircraft and system functions, their failure conditions, and possible effects, and classifies the failure conditions accordingly. Depending on the severity classification of the failure condition, a development assurance level is assigned, which is a requirement on the rigor of development activities associated with that function or items implementing the function. The idea is that a certain level of development

rigor shall assure that to a sufficient degree, development errors are mitigated. The more severe the failure condition, the more rigor is required for the associated functions and items.

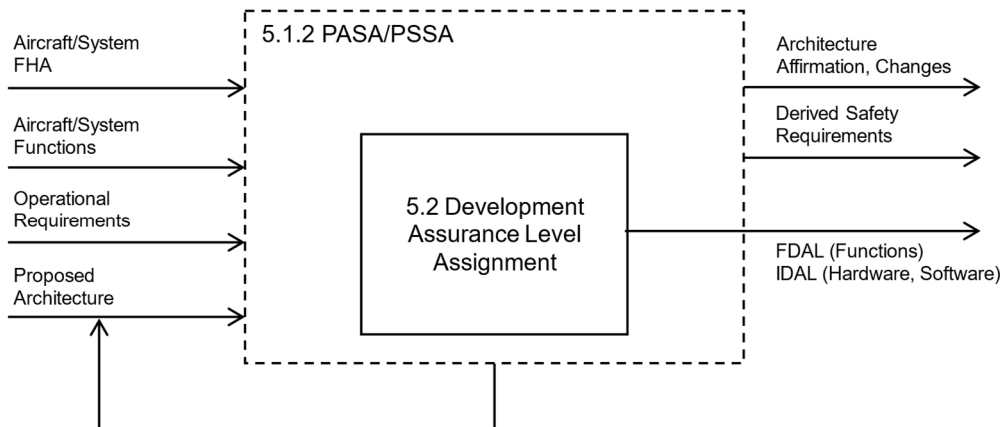


Figure 2.6: ARP4754B Function/Item Development Assurance Level (FDAL/IDAL) assignment process, adopted from [37].

For the top-level aircraft functions, a Functional Development Assurance Level (FDAL) is assigned, such that for catastrophic failure conditions, the function is assigned FDAL A, hazardous conditions FDAL B, major conditions FDAL C, minor conditions FDAL D, and if there is no safety effect FDAL E. If the aircraft function is allocated to a single system, the same FDAL would apply to the system functions (for example the automatic flight control system functions implementing the aircraft level function automatic flight control). If there is a set of independent system functions that implement an aircraft level function (such as wheel brakes and reverse thrust independently implement the aircraft level function retardation on ground), ARP4754B defines combinations of lower FDALs for the respective system functions that would be sufficient. Functional independence may be claimed where the functions are different such that the likelihood of a common requirement error or error in requirement interpretation is minimized.

From the system functions and their FDAL, hardware and software development assurance are specified as Item Development Assurance Level (IDAL), corresponding to software level in DO-178C. Depending on the system architecture and the independence of the hardware and software items that implement the system functions, analog to functional independence on the aircraft level, reduced IDALs may be possible. Item independence may be claimed where the items (hardware and software) are different such that the likelihood of a common development, design, or tool error is minimized. Examples of means of achieving item independence include different development teams and processes, different coding languages, different operating systems, and different microprocessors.

Requirements Capture

The development process is tightly coupled with the requirements capture integral process, which is recursively applied at every level (aircraft, system, item).

ARP4754B groups the system requirements into four main categories:

- Safety Requirements
- Functional Requirements
- Additional Certification Requirements
- Derived Requirements

Safety requirements specify minimum performance constraints for the availability and integrity of the system functions and are determined from the safety assessment process.

Functional requirements include customer requirements (e.g., desires, operation constraints, and desired features), operational requirements (e.g., actions, decisions, and information), performance requirements (e.g., accuracy, fidelity, range, and response times), physical and installation requirements, maintainability requirements, and interface requirements.

Additional certification requirements encompass functions, attributes, and implementations required by airworthiness regulations.

Derived requirements are those requirements originating from the design process itself, for example functional design or system architecture decisions, without an explicit relation to higher-level requirements.

System requirements may be categorized and grouped in different ways. There is no single standardized way of categorizing system requirements; each guidance material or standard differs, although many requirement types are common. Software development typically differentiates between functional requirements, defining *what* the system does, and non-functional requirements, defining *how* the system does it, including for example performance, security, and usability requirements. For the purpose of this thesis, the categorization according to ARP4754B is adopted.

2.2.2 DO-178C/DO-331 Model-Based Software Development Process Model

The DO-178C defines a set of software life cycle phases and a process framework with extensive process requirements for the development of software for airborne systems and equipment. DO-178C has many structural similarities with ARP4754B in terms of development vs. integral processes, process objectives depending on assurance levels, control categories, etc., and it assumes that the software development process is embedded in a systems development process according to ARP4754B.

The DO-178C does not define a specific software life cycle model, but rather a set of life cycle phases, such as Requirement, Design, Coding, and Integration, which may be combined in different sequences according to the needs and constraints of the specific development context. An implementation-specific life cycle model would depend on, among other things, the system and software functionalities and complexity.

The DO-178C process framework is structured similarly to that in ARP4754B, consisting of a planning process, a development process with a set of subprocesses, and a set of integral processes. The process framework is illustrated in Figure 2.7.

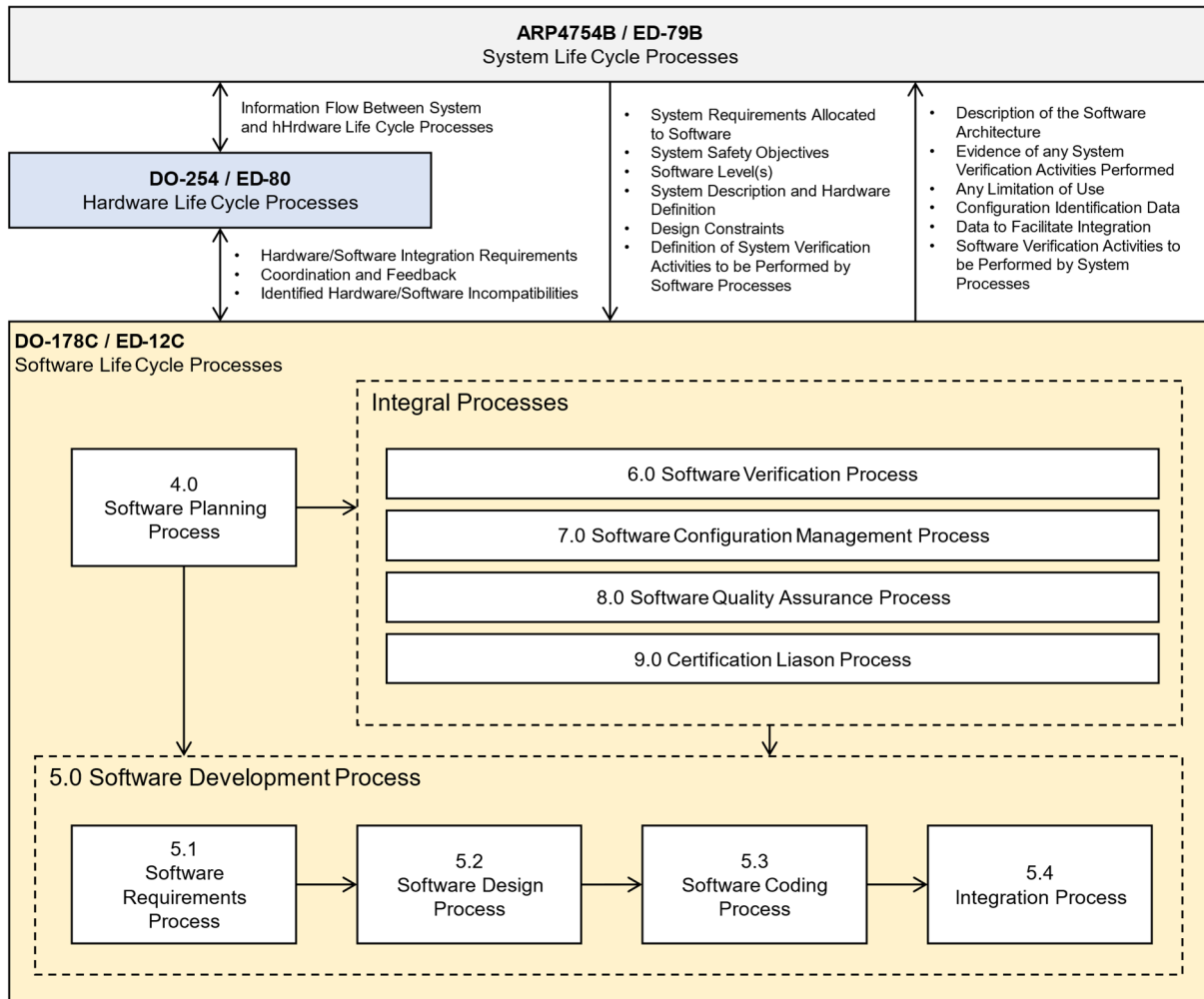


Figure 2.7: DO-178C software development process model.

Information Flow between System and Software Life Cycle Processes

The system is implemented as hardware and software, thus the respective life cycles for system, hardware and software development must exchange information and life cycle data to enable a successful and safe system to be realized. From the system to the software life cycle, systems requirements that are to be realized in software, the so-called Systems Requirements Allocated to Software (SRATS) flow. SRATS may include functional requirements, interface requirements, performance requirements, safety-related requirements such as design constraints and methods, security requirements maintenance requirements, and certification requirements. The system-level development assurance level assignment process determines the software level, which governs the software development and verification effort. Due to the iterative nature of system development, the software-level activities must analyze the allocated system requirements and refer any incorrect or inadequate requirements back to the system-level processes.

Software Planning Process

The software planning process produces the software development plans and other documents required to show how compliance with the standard is to be achieved. The Plan for Software Aspects of Certification (PSAC) is the main plan and the principal document agreed upon with the certification authorities before development begins. Additional required plans correspond to the software life cycle processes, and include a Software Development Plan, Software Verification Plan, Software Configuration Plan, and a Software Quality Assurance Plan. In addition to the plans, a set of development standards are required, corresponding to the software development phases: a Software Requirements Standard, a Software Design Standard, and a Software Coding Standard.

Software Development Process

The software development process consists of four subprocesses: software requirements process, software design process, software coding process, and integration process.

The software requirements process develops one or more levels of software requirements. High-level requirements are developed from analysis of the allocated system requirements and the system architecture. Typically, the high-level requirements are broken down into one or more lower levels of requirements. The software design process developed the software architecture and low-level requirements. The low-level requirements are the level from which source code can be implemented, in the software coding process. In the case that source code is implemented directly from the high-level requirements, these are regarded as low-level requirements as well, and the associated objectives apply. The integration process generates executable object code from the source code and loads it together with parameter data files onto the target hardware for hardware/software integration.

Software Verification Process

The software verification process is the technical assessment of the outputs of the other processes in order to provide evidence that allocated system requirements have been developed into high-level requirements, the high-level requirements have been developed into software architecture and low-level requirements, the software architecture and low-level requirements have been developed into source code, that the executable object code is robust, satisfies the requirements from which it was developed, and that any unintended functionality is absent. The software verification process may use a combination of different means of verification, such as reviews, analyses, test cases, and procedures.

Software Configuration Management Process

The software configuration management process shall provide a defined and controlled configuration of the software throughout its life cycle, repeatability of the generation of life cycle data and process activities, establishment and change control of configuration items and software baselines, and ensure the proper archiving, control, and recovery of configuration items.

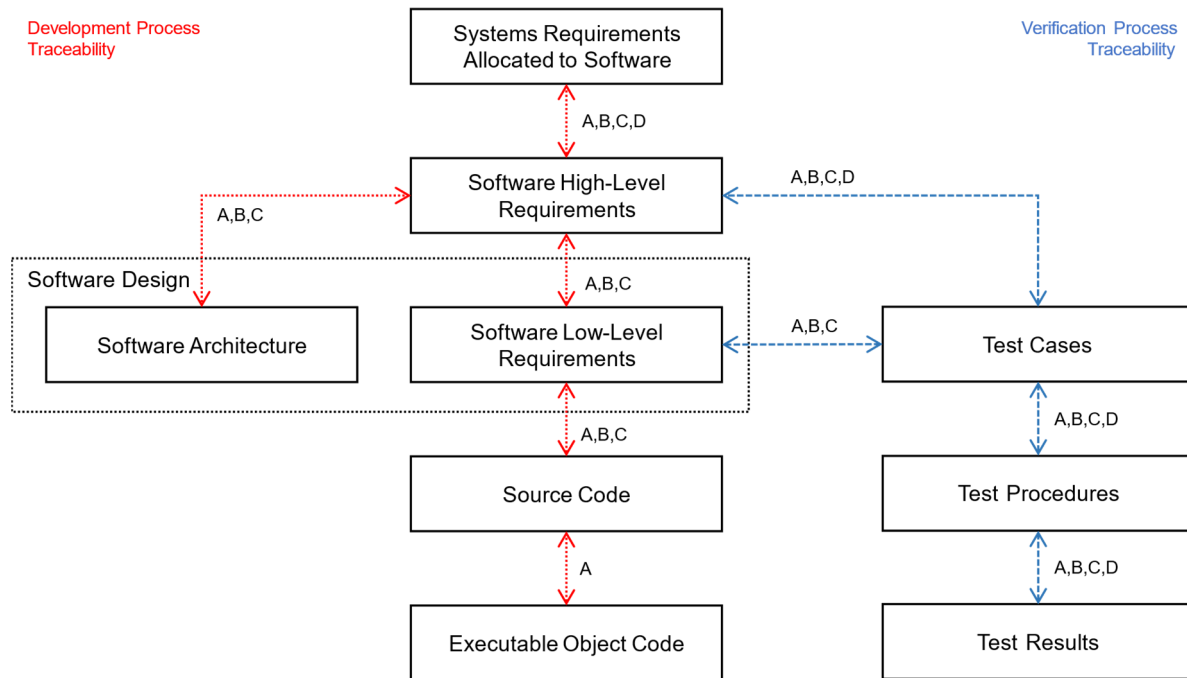


Figure 2.8: DO-178C development and verification traceability requirements depending on development assurance level.

Traceability

DO-178C puts much focus on traceability between life cycle data. *Development process traceability* is the required traceability between development artifacts such as high-level requirements, low-level requirements, and source code. *Verification process traceability* is the required traceability between requirements, test cases, and test results. *Configuration traceability* is the required change control and traceability between different baselines of the same configuration item. The level of required traceability requirements is dependent on the software development assurance level, with level A requiring traceability down to the executable object code. The traceability requirements depending on the software level are visualized in Figure 2.8.

Model-Based Software Development and DO-331

Models may serve many purposes in the development of systems as well as software. For airborne systems and software dependent on the aircraft dynamics, such as a flight control system, a *plant model* or *flight dynamics model* captures relevant aspects of the aircraft dynamics, together with relevant subsystem dynamics. A plant model can be trimmed and linearized to support analysis of the aircraft dynamics for control law development or be integrated into a simulation environment together with other *verification and validation models* or *test harness models* for real-time or accelerated simulation-based verification and validation activities. Control laws may be developed or rapidly prototyped using *algorithm design models*, for integration into test harness models and early validation of desired functionalities and performance. The algorithm design models are not the actual implementation of the control

laws but may be regarded as executable specification models. Control law and other functional algorithms may be implemented as *software design models* for automatic code generation. Textual requirements may be formalized in *requirement models*, supporting model-based verification and automated requirements checking.

The DO-331 supplement addresses the use of models in software development for airborne applications and distinguishes between *design models* and *specification models* as the only types of models that may replace software life cycle data according to DO-178C. A specification model replaces “high-level requirements that provide an abstract representation of function, performance, interface, or safety characteristics of software components” (DO-331 MB.1.6.2), whereas “a design model includes LLRs and/or software architecture” (DO-331 MB.1.6.2).

2.2.3 ISO/IEC/IEEE 15288 Life Cycle Process Model

The ISO/IEC/IEEE 15288 [63] is a commonly used standardized framework for complex systems development and life cycle management, across domains. It provides guidance for the entire life cycle of a system, not just system development. The standard is generic and does not prescribe a specific life cycle model or development approach. It serves as a toolbox of best-practice processes that are intended to be tailored to the specific system context, its complexity, novelty, and level of risk. The ISO/IEC/IEEE 15288 system life cycle process framework is shown in Figure 2.9. The process framework consists of four process groups, with many similarities with the ARP4754B and DO-178C process models. The four groups are summarized in the following.

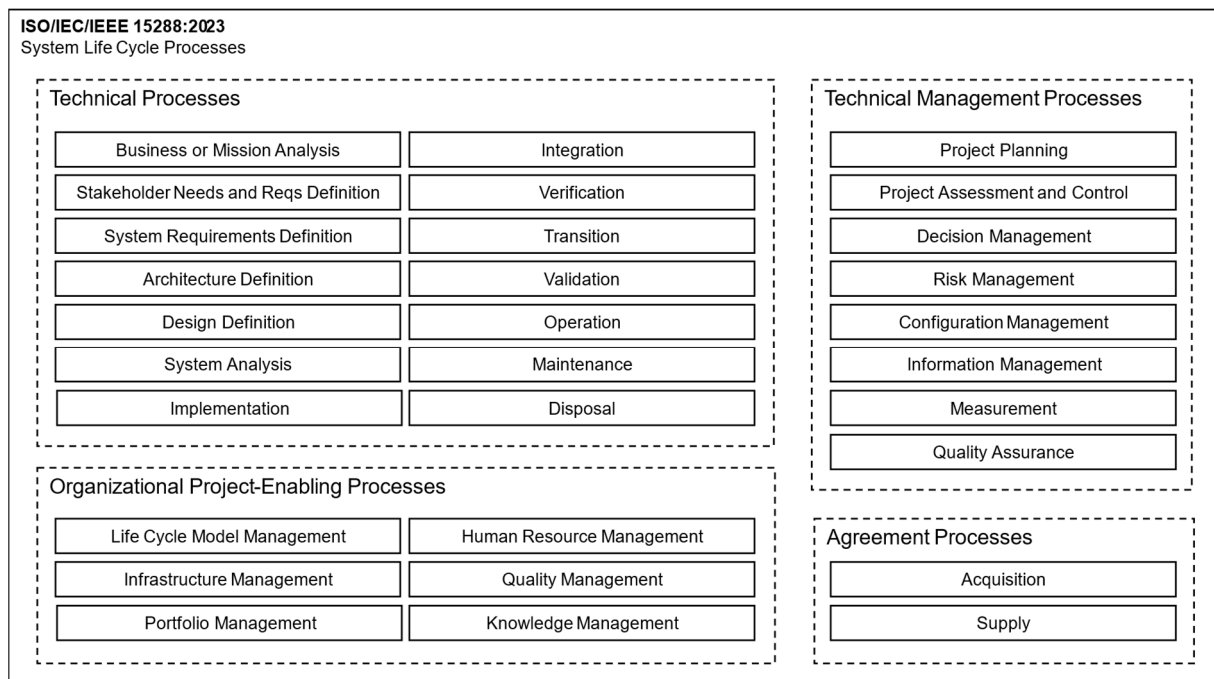


Figure 2.9: ISO/IEC/IEEE 15288:2023 system life cycle process framework.

- Technical Processes: mainly correspond to the development processes of the ARP4754B or DO-178C, however, covering the entire life cycle of the system including operation, maintenance, and disposal.
- Technical Management Processes: mainly correspond to the planning and integral processes of the ARP4754B or DO-178C.
- Organizational Project-Enabling Processes: consider organizational aspects of system development that are outside the scope of the ARP4754B or DO-178C.
- Agreement Processes: consider the relation between interacting organizations, projects, or other entities during system development, which need to be regulated in agreements.

Some differences between ISO/IEC/IEEE 15288, ARP4754B, and DO-178C are worth noting. ISO/IEC/IEEE 15288 does not include a specific safety assessment process. System safety engineering is regarded as a specialty engineering activity that is not the subject of any specific process, but rather an integrated part of the engineering effort, with safety-related requirements guiding the design, production, operation, support, and retirement of the system, but not detailing how this is achieved. The INCOSE Systems Engineering Handbook [68], which provides additional guidance on the ISO/IEC/IEEE 15288 system life cycle processes, explicitly refers to the safety assessment process of the ARP4754B and ARP4761A as best-practice within the aviation industry.

ISO/IEC/IEEE 15288 includes a dedicated process for risk management. In the ARP4754B, risk is associated with system safety, where risk is defined as *the combination of the frequency (probability) of an occurrence and its associated level of severity*, and safety is defined as *the state in which risk is acceptable*. The ARP4754B approach to risk and safety is concentrated on the system design, that the process outcome is a safe system. The risk management approach of the ISO/IEC/IEEE 15288 puts focus on risks associated with the development and life cycle management of the system, such as project risks, technological risks, and process risks. This view is missing in the ARP4754B view on safety assessment and risk.

2.3 AFCS Development Process

2.3.1 Overview

The AFCS Development Process is implemented as part of the modular FGCS life cycle in Figure 2.2 (see Section 2.1.3) with the following objectives:

- An ARP4754B-oriented development, to provide assurance that the concept developed aligns with standard means of compliance
- Provide a basis for tailoring and integration of development process elements into an aircraft-specific application context and life-cycle model according to Figure 2.2

The development process is illustrated in Figure 2.10, with corresponding chapters in the thesis as well as related ARP4754B process elements noted. This section summarizes the respective process activities.

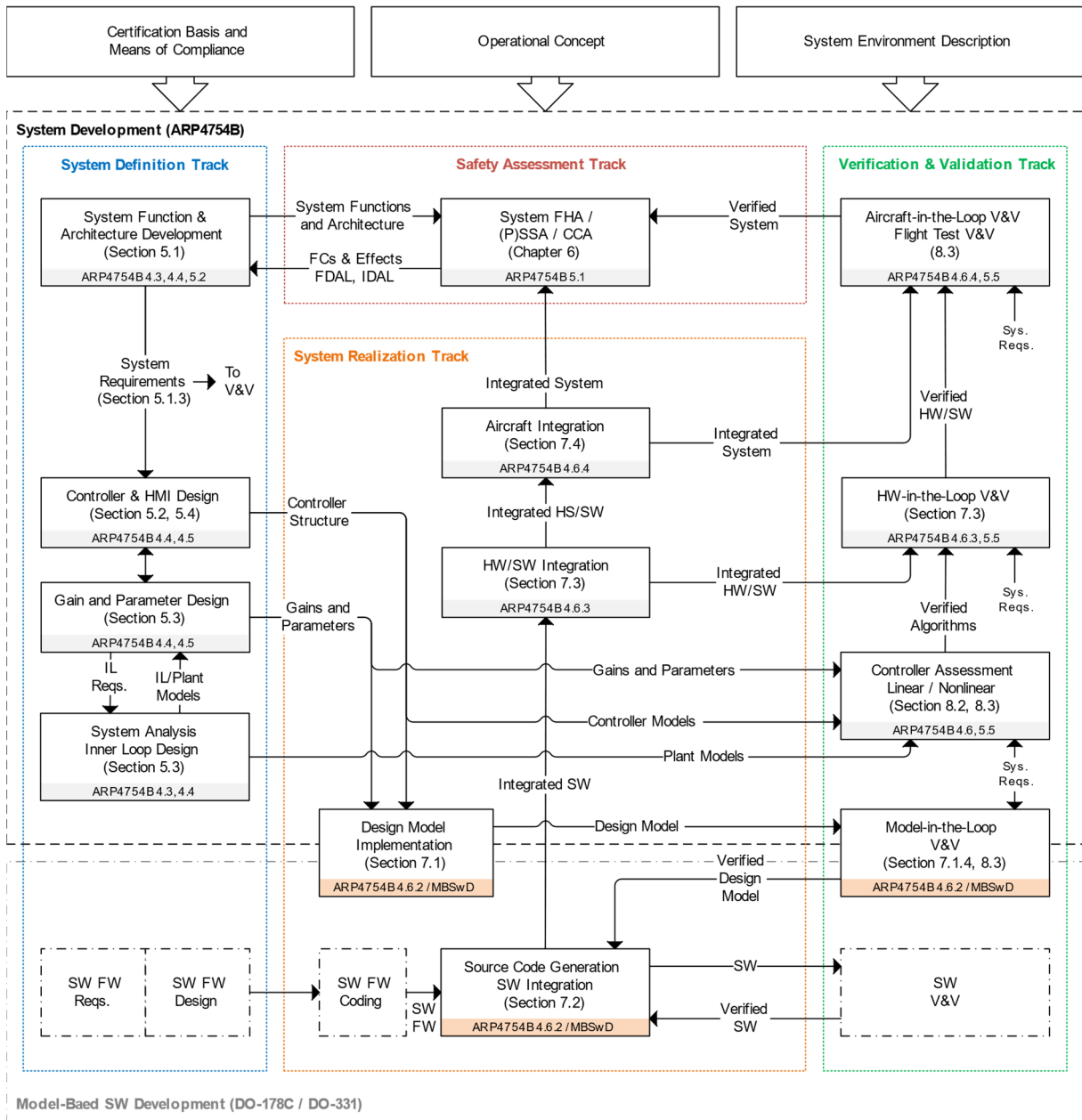


Figure 2.10: AFCS development process with related thesis chapters and sections.

The development process elements and activities are grouped into four main tracks, reflected in subsequent chapters in this thesis:

- *System Definition Track* (Chapter 5) which comprises the definition of requirements, architecture considerations, functional structure, and control law design
- *Safety Assessment Track* (Chapter 6) which comprises the system-level FHA, identification of safety-related requirements and required development assurance levels as input to the system architecture considerations in the System Definition phase
- *System Realization Track* (Chapter 7) which comprises the implementation and integration activities

- *Verification and Validation Track* (Chapter 8) which comprises the control law assessment and verification/validation activities at software and system level up to and including flight testing

Major inputs from a concept phase are summarized in Assumption 1 for the AFCS Development Process, Section 2.1.4:

- Certification basis and means of compliance
- Operational concept
- System environment description

2.3.2 System Definition

The system definition track encompasses the following activities:

- System Function and Architecture Development
- System Analysis / Inner Loop Design
- Controller Design
- Controller Gain and Parameter Design

The scope of the activities is described in the following. The systems definition for the AFCS is further presented in Chapter 5.

System Function and Architecture Development

The desired system functions are driven by the desired operational concept and mission needs, as defined in the concept phase. The system architecture is driven by the required development assurance level, as determined by the safety assessment process. The design and performance of both system functions and architecture are constrained by the system environment into which the system is to be integrated.

The main output of the system function and architecture development is the set of system requirements and a system architecture. For the AFCS, system functions and architecture are developed from a principal system architecture with generic architecture and functional elements, and allocation of system functions to items and interfaces, see Figure 2.11. The principal architecture and generic elements are adopted from the guidelines in DO-325 Minimum Operation Performance Standards (MOPS) for Automatic Flight Guidance and Control Systems and Equipment [35], which defines common characteristics of functional and architectural elements.

The development of the AFCS concept as part of the modular FGCS life cycle in Figure 2.2 focuses on the system function development and associated system requirements. The principal architecture and generic elements provide a common basis for reusability and parametrization of requirements for new AFCS applications but require application-specific tailoring. The physical system architecture is highly dependent on the aircraft application platform, with required development assurance level, level of redundancy of physical components, integration with legacy sensor and actuation systems, display and mode control interfaces, aircraft trim systems, and physical installation constraints.

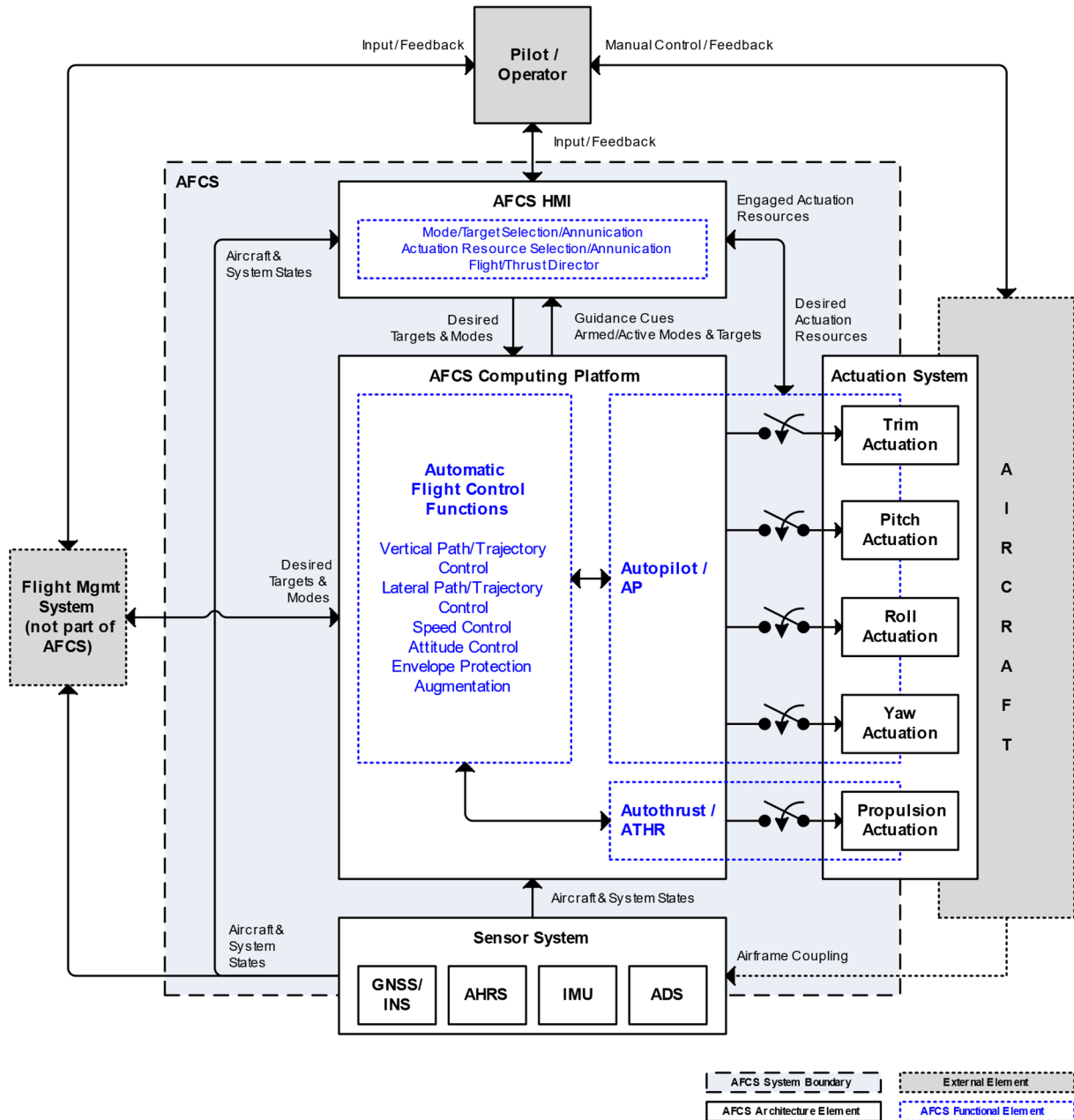


Figure 2.11: Principal AFCS architecture with generic architecture elements, adopted from [35].

General AFCS architecture elements include:

- Computing platform
- HMI for AFCS mode control and monitoring as well as guidance in selected AFCS operation
- FMS/MMS for external guidance in managed AFCS operation
- Actuation resources for pitch/roll/yaw and trim, as well as corresponding engagement/disengagement mechanisms
- Sensors for aircraft and system state data

Depending on the aircraft application platform, the AFCS sensor and actuation elements may be fully or partially realized by existing sensor suites and actuation systems. This is illustrated in Figure 2.11 by the sensors and actuation elements placed on the AFCS system boundary.

Together with the generic architecture elements, a set of functional elements with desired nominal system functions is defined, with a description of the intended system behavior. For the AFCS, the generic functional elements and set of nominal system functions comprise:

- Automatic Flight Control Functions (AFCF)
 - Vertical/lateral path/trajectory control functions
 - Speed control functions
 - Attitude control functions
 - Envelope protection functions
 - Augmentation functions (e.g., stability augmentation)
- Mode Control and Monitoring Functions
 - Mode and target selection and annunciation in selected AFCS operation (guidance from AFCS HMI)
 - Mode and target annunciation in selected and managed AFCS operation (guidance from FMS)
 - Selection of annunciation of actuation resources for automatic flight
 - Flight director cues for manual flight / automatic flight monitoring
 - System state annunciation for system monitoring
- Autopilot (AP) Functions
 - Automatic pitch/roll/yaw control
 - Automatic trim
 - Autopilot engagement/disengagement
- Autothrust (ATHR) Functions
 - Automatic thrust control
 - Autothrust engagement/disengagement
- Safety Functions
 - Input monitoring functions
 - Subsystem monitoring functions
 - Automatic AP/ATHR disengagement

The set of functions and functional descriptions are the main input to the system-level FHA together with the operational concept (operational scenarios, flight phases), and applicable certification regulations and guidance material (for example advisory circulars with failure condition classification guidance). The FHA generates the system-level FDAL based on the classification of identified failure conditions, together with any safety-related system functions and constraints. The system-level FDAL can be achieved in multiple ways, through different combinations of system architecture decisions and hardware and software IDALs, depending on the level of redundancy and item independence.

The set of system functions and safety constraints is the basis for the system requirements. The system requirements related to the automatic flight control functions and HMI functions

are grouped into general system requirements applicable to all functions, and function-specific system requirements.

General system requirements include:

- Functional Requirements
 - Operational envelope and aircraft configurations
 - Command limits
 - Vertical modes and transition criteria
 - Lateral modes and transition criteria
 - Control objective prioritization
 - Speed/path prioritization criteria
 - Vertical/lateral curvature prioritization criteria
 - Energy protection
 - General performance, e.g. stability and robustness requirements
 - Interface requirements
 - Command inputs for selected and managed operation
 - Sensor inputs (range, resolution, update rates)
 - Inner loop command interface
 - Monitoring interface
- Safety Requirements
 - Monitoring functions
 - Fail-safe behavior (e.g., automatic disengagement, annunciation)

System requirements that define function-specific behavior and performance, as well as function-specific variants of general system requirements, encompass:

- Functional requirements detailing intended behavior
- Function-specific envelope and aircraft configuration
- Function-specific/flight phase command limits
- Function-specific/flight phase desired/adequate performance and robustness

Desired performance is the design target and would typically be perfect following of a reference flight path. In reality, the desired performance for all functions under all conditions will not be achievable. Thus, an adequate performance is defined that must be achieved for all applicable conditions. Adequate performance may be directly specified from corresponding requirements in applicable guidance material such as AS94900A [103]. Requirements for desired/adequate performance include:

- Target overshoot and settling time
- Disturbance rejection
- Tracking accuracy in smooth air
- Tracking accuracy in turbulent air

In the case of design conflicts, where different regulations and standards have directly or indirectly contradictory requirements, a priority hierarchy is defined.

Performance requirements for the inner loops for the path controller to achieve desired or adequate performance are not explicitly related to the high-level system requirements, but *derived requirements* resulting from the design process.

A number of system requirements relate to the physical system architecture and items, such as hardware/software development assurance levels, physical and installation requirements, reliability and availability requirements, as well as maintainability requirements. These types of requirements are not further elaborated here.

System Analysis / Inner Loop Design

From the perspective of the flight control system, the aircraft and its dynamics are regarded as part of the system environment. For the design and assessment of the control laws, a Flight Dynamics Model (FDM) of the aircraft is developed, as well as models of relevant subsystems such as propulsion, sensors, actuators, and mechanical flight control system, see Section 1.3.3. The subsystem models may include failure states and their effects on the system dynamics, either to be compensated for by the flight control system or as inputs to the safety assessment process. The rigid-body equations of motion from which the FDM is developed, and dynamics of typical subsystems are discussed in Chapter 3, although the plant modeling process is not the focus of this thesis.

From the perspective of the AFCS development, the closed inner loop dynamics, including the dynamics of the control surface actuators, as well as the closed thrust control loop including the propulsion dynamics, are regarded as the path control actuation dynamics, i.e., the transfer functions from the commanded specific forces to the actual ones. The closed loop dynamics is analyzed on the basis of linearized models, for a set of trimmed, quasi-stationary flight conditions. Eigenvalues and -vectors for the full or reduced order system are analyzed, with frequency and damping of the characteristic aircraft modes.

For existing sensor suites or intended sensor systems, the sensor dynamics are analyzed with respect to data range, resolution, update rates and error characteristics, as input to the design of sensor filters for the controller design.

Controller Design

The controller design process develops the system requirements into a functional design, i.e., the control laws, limiters, control mode logics, and initialization logics. The controller design process includes the selection of appropriate feedback structures, reference model dynamics, error dynamics, and input signal filtering.

The path controller design process is coupled with the inner loop and thrust loop design processes. The path loop design includes rapid prototyping and validation, using simplified point mass models with generic transfer functions for estimated inner loop and actuation dynamics with representative frequencies and damping. Additional requirements on inner loop and thrust control dynamics are derived, in order to achieve outer loop performance, and fed to the inner loop design process. Achievable inner loop dynamics are fed back to the path control loop process.

For aircraft-specific application cases, the configuration of the path controller structure is limited to the omission of non-desired modes or mode transitions, sensor-specific signal filtering, and aircraft-specific automatic trim control functions.

Gain and Parameter Design

With the controller structure defined, the gain and parameter design process generates the numerical values of all controller parameters, for example, reference model time constants, error controller gains, internal limiters, and filter dynamics.

The aircraft-specific parameters are captured as data structures, referenced by the generic design model implementation during initialization.

Reference dynamics time constants, error dynamics gains as well as command and internal limiters are scheduled over the aircraft envelope and aircraft configurations in order to utilize the full available performance of the inner loop and thrust loop dynamics. The scheduling grid and scheduling variables are determined from the scheduling grid and variables of the aircraft-specific inner loop.

2.3.3 Safety Assessment

The safety assessment process track for the AFCS focuses on the *system-level FHA*, according to the ARP4754B/ARP4761A system safety assessment process, Figure 2.5, see Section 2.2.1. The FHA is function-oriented and architecture-agnostic. The physical system architecture that implements the system functions is dependent on the application scenario and aircraft platform, with different valid system architecture solutions of varying levels of redundancy, dissimilarity, and independence satisfying required functional and hardware/software development assurance levels resulting from the FHA.

The objective of the safety assessment track is to show how the AFCS functions may be assessed with respect to safety and provide the necessary inputs to an application life cycle and the development of a system architecture meeting the requirements of the specific operational scenario and regulatory environment.

The scope of the AFCS safety assessment is described in the following. The safety assessment activities performed are detailed in Chapter 6.

The system-level FHA is performed in order to identify system hazards and determine the necessary level of safety. The main inputs are the set of system functions from the system definition track together with the operational concept (operational scenarios, flight phases), and applicable certification regulations and guidance material (such as advisory circulars with classification guidance).

The FHA activities include:

- Identification and description of failure conditions
- Determination of failure condition effects
- Classification of failure conditions
- Assignment of requirements to failure conditions

- Identification of method for compliance verification

The FHA produces *worksheets*, tables with rows for each assessed function and columns representing the outcome of each activity above. The FHA worksheet columns are briefly explained in Table 2.1. A detailed description of the FHA worksheet columns with reference to analysis support and guidance material is given in Appendix E, Table E.1.

Identification and Description of Failure Conditions

Functional failure conditions should be considered over the entire flight envelope and for abnormal operating conditions Both detected and undetected loss of functions and malfunctions are considered. Typical failure conditions include:

- Loss of function
- Unannounced loss of function
- Partial loss of function
- Malfunction/erroneous behavior

Operational flight phases to be considered for each failure condition include:

- Climb / Descent
- Cruise
- Maneuvering Flight
- Approach

Table 2.1: FHA worksheet columns and explanation.

Column	Explanation
Function	Function name
Failure Condition	Description of failure condition
Phase	Flight phase or condition affecting the effect of the assessed failure condition
Effect of Failure Condition on Aircraft/Crew	Description of the effect of the failure condition at aircraft level
Classification	Resulting qualitative classification according to the failure effect
FDAL	Functional Development Assurance Level associated with the failure condition classification
Probability/FH	Maximum Probability per Flight Hour (FH) for failure condition occurrence
Reference to supporting material	Reference to acceptable means of compliance, industry standards, test reports or other documentation justifying the failure effect and/or classification
Verification	Method of showing compliance with classification and related probability requirements

Determination of Failure Condition Effects

Failure conditions are analyzed with respect to their effect on aircraft, crew, occupants, and environment for the various flight phases. Factors considered when assessing the failure condition effects include:

- The impact of the full or partial loss of function or malfunction/erroneous behavior on the structural integrity of the aircraft
- Implications of the aircraft response in terms of attitude, speed, accelerations, and flight path, as well as the impact on the occupants and on pilot performance
- Degradation in the stability or other flying qualities of the airplane
- The duration of the condition
- The airplane configuration
- The airplane motion cues that will be used by the pilot for recognition
- Availability, level and type of alerting provided to the pilot
- Expected pilot corrective action on detection of failure

Where the failure condition effects differ between flight phases, the effects and their severity classification should be explicitly listed according to flight phase.

The severity of a failure condition may be dependent on whether the aircraft is flown manually or by the AFCS. When flown by the AFCS, the pilot reaction time in case of a failure condition is dependent on the pilot attentiveness, which depends on flight phase and associated duties.

The pilot may detect a failure condition through airplane motion cues or by cockpit flight instruments and alerts. Malfunction or erroneous behavior may result in so-called “hardover” or “softover” effects. “Hardover” effects, with sudden resulting aircraft motion or guidance cues are typically significant and immediately detectable by the pilot. “Softover” effects, on the other hand, are typically not immediately detectable by the pilot, with gradual departures from the intended flight path which may only be detected when a significant path deviation has occurred.

Assumptions regarding pilot reaction time before recovery action is initiated, depending on flight phase are listed in Table 2.2.

Table 2.2: Pilot reaction times, adopted from [104].

Flight Phase	Pilot Reaction Time [s]
Climb / Descent	3
Cruise	3
Maneuvering Flight	1
Approach	1

Classification of Failure Conditions

Based on the analysis of their effects, the failure conditions are classified as catastrophic, hazardous, major, minor, or as having no safety effect, with an associated qualitative probability of occurrence. Table 2.3 summarizes the classifications and qualitative probabilities as function of failure condition effect on airplane, occupants, or flight crew.

Table 2.3: Classification of failure conditions and associated qualitative probability requirements, adopted from [104].

Classification of Failure Conditions	No Safety Effect	Minor	Major	Hazardous	Catastrophic
Allowable Qualitative Probability	No Probability Requirement	Probable	Remote	Extremely Remote	Extremely Improbable
Effect on Airplane	No effect on operational capabilities or safety	Slight reduction in functional capabilities or safety margins	Significant reduction in functional capabilities or safety margins	Large reduction in functional capabilities or safety margins	Normally with hull loss
Effect on Occupants	Inconvenience for passengers	Physical discomfort for passengers	Physical distress to passengers, possibly including injuries	Serious or fatal injury to an occupant	Multiple fatalities
Effect on Flight Crew	No effect on flight crew	Slight increase in workload or use of emergency procedures	Physical discomfort or a significant increase in workload	Physical distress or excessive workload impairs ability to perform tasks	Fatal Injury or incapacitation

The AC 23.1309-1E [104] provides additional guidance on the classification of certain failure conditions related to AFCS. The following relevant catastrophic, hazardous, and major failure conditions for the AFCS functions are to be considered.

Catastrophic failure conditions:

- Unrecoverable loss of flight path control
- Exceedance of V_D/M_D (demonstrated flight dive speed/demonstrated flight dive Mach)
- A temporary loss of control (for example stall) where the pilot is unable to prevent contact with obstacles or terrain
- Deviations in flight path from which the pilot is unable to prevent contact with obstacles, terrain, or other airplane

Hazardous failure conditions:

- Exceeding of an airspeed halfway between V_{MO} (maximum operating limit speed) and V_D or a Mach number halfway between M_{MO} (maximum operating limit Mach) and M_D
- A stall, even if the pilot is able to recover safe flight path control
- A load factor less than zero
- Bank angles more than 60 degrees
- A flight path deviation that requires a severe maneuver to prevent contact with an obstacle, terrain, or other airplane

Major failure conditions:

- A flight path deviation, including the required recovery maneuver, which may result in passenger injuries

The most critical failure condition for an automatic flight control system would be a malfunction in a critical flight phase, i.e., during approach. The criticality of an unannounced autopilot hardover is dependent on the number of axes controlled by the autopilot and its control authority, as described in Table 2.4.

Table 2.4: Classification of AFCS hardover depending on authority [104].

Autopilot authority	Classification of hardover malfunction without warning
Single-axis, limited authority	Major
Multi-axis, limited authority	Hazardous
Multi-axis, unlimited authority	Catastrophic

Assignment of Requirements to Failure Conditions

The failure condition classification determines a qualitative probability of occurrence. For CS-23 airplanes, the corresponding quantitative probability and development assurance levels are depending on the aircraft class, as summarized in Table 2.5.

The FHA also identifies safety requirements needed to limit the effects that determine the failure condition classification, for example, design constraints, annunciation of failure conditions, recommended flight crew, or maintenance actions.

Identification of Method for Compliance Verification

Depending on the classification of the failure condition and the type of system, the method and depth of the subsequent safety analysis and verification of safety requirement compliance are different. The required depth of analysis for different failure condition classifications and type of system combinations is given in Appendix E, Table E.2, which summarizes the depth of analysis requirements in AC 23.1309-1E [104].

System types that affect compliance verification methods encompass:

- Simple and conventional installations
- Systems similar in relevant attributes and failure conditions to other already certified systems
- Complex and non-redundant systems
- Redundant systems

Qualitative verification methods include:

- Design and installation appraisals or analysis to claim similarity in attributes and effects to previously certified systems, or sufficient independence of redundant systems
- Qualitative FTA, dependency diagrams (DD), or Markov analysis (MA) to analyze failure propagation and robustness against single failures

Quantitative verification methods include:

- Quantitative FTA, DD, MA, or functional FMEA to analyze failure propagation and probabilities, as well as robustness against single failures

The System Safety Assessment shall provide documentation that the required analysis has been performed and the results of that analysis.

Table 2.5: Relationship between airplane classes, probabilities, failure condition severity, and IDAL requirements for primary (P) and secondary (S) items (adopted from [104] Figure 2).

Classification of Failure Conditions	No Safety Effect	Minor	Major	Hazardous	Catastrophic
Class I (Typically SRE 6,000 pounds or less)	No Probability Requirement	$<10^{-3} / \text{FH}$	$<10^{-4} / \text{FH}$	$<10^{-5} / \text{FH}$	$<10^{-6} / \text{FH}$
	No IDAL Requirement	P = IDAL D	P = IDAL C S = IDAL D	P = IDAL C S = IDAL D	P = IDAL C S = IDAL C
Class II (Typically MRE, STE or MTE 6,000 pounds or less)	No Probability Requirement	$<10^{-3} / \text{FH}$	$<10^{-5} / \text{FH}$	$<10^{-6} / \text{FH}$	$<10^{-7} / \text{FH}$
	No IDAL Requirement	P = IDAL D	P = IDAL C S = IDAL D	P = IDAL C S = IDAL C	P = IDAL C S = IDAL C
Class III (Typically SRE, STE, MRE, and MTE greater than 6,000 pounds)	No Probability Requirement	$<10^{-3} / \text{FH}$	$<10^{-5} / \text{FH}$	$<10^{-7} / \text{FH}$	$<10^{-8} / \text{FH}$
	No IDAL Requirement	P = IDAL D	P = IDAL C S = IDAL D	P = IDAL C S = IDAL C	P = IDAL B S = IDAL C
Class IV (Typically Commuter Category)	No Probability Requirement	$<10^{-3} / \text{FH}$	$<10^{-5} / \text{FH}$	$<10^{-7} / \text{FH}$	$<10^{-9} / \text{FH}$
	No IDAL Requirement	P = IDAL D	P = IDAL C S = IDAL D	P = IDAL B S = IDAL C	P = IDAL A S = IDAL B

2.3.4 System Realization

The system realization track encompasses the following activities:

- Design Model Implementation and Integration
- Source Code Generation and Software Integration
- Hardware-Software Integration
- Aircraft Integration

The systems realization for the AFCS is presented in Chapter 7.

Design Model Implementation and Integration

The design model from which the source code is directly generated is implemented directly from the functional design at the system level and in accordance with modeling guidelines and

library blocks from the modular model-based software development process. The AFCS design model is integrated with interfacing control application modules to a complete design model into different test harness models for model-level verification and validation.

Source Code Generation and Software Integration

Source code is automatically generated from the design model using MathWorks *Embedded Coder*. Alternative approaches for automatic code generation exist, with the generation of code from an integrated design model with all functional models included, or the generation of code from each design model with code integration after generation [98].

Hardware-Software Integration

The hardware-software integration includes the generation of executable object code from the integrated software and deployment on the target computer hardware. The target computer is integrated with interfacing systems such as the actuation system and sensor models in a hardware-in-the-loop laboratory setup.

The purpose of the hardware-in-the-loop laboratory setup is to enable testing of complete system behavior, communication, timing, bus loads, wiring, etc. with the intended physical components and interfaces. The hardware-in-the-loop setup enables real-time simulations, or “virtual flight demonstrations” for increased confidence before aircraft integration and flight testing.

Aircraft Integration

Physical installation of integrated and hardware-in-the-loop verified hardware and software in the aircraft, interfacing with intended systems and mechanical structures.

2.3.5 Verification and Validation

The verification and validation track encompasses the following activities:

- Control Law Assessment, Linear and Nonlinear
- Model-in-the-Loop (MIL) Verification and Validation
- Hardware-in-the-Loop (HIL) Verification and Validation
- Aircraft-in-the-Loop (AIL) Verification and Validation
- Flight Test Verification and Validation

The verification and validation of the AFCS are presented in Chapter 8.

The objective of the *verification* is to confirm that the system has been correctly implemented and that the requirements have been satisfied, i.e., that the system has been “built right” (with respect to requirements/applicable guidelines). The objective of the *validation* is to confirm that the system satisfies the needs of the stakeholders, i.e., that “the right system” has been built (with respect to the needs).

Verification and validation are performed continuously during the entire development process and at multiple levels with different purposes (cf. discussion on modular FGCS levels in Section 2.1.1):

- At the application level, to validate appropriate algorithm and logic design and their correct implementation
- At the system level using linearized controller, aircraft, and subsystem dynamics to verify stability and robustness
- At the system level using model-in-the-loop simulation to validate the functionality of the closed-loop system in operational scenarios and to verify functional aspects of interfaces
- At the system level using hardware-in-the-loop simulations to verify closed loop performance, computation and timing aspects, as well as physical interfaces
- At the aircraft level to validate and verify system functionalities and their performance in flight

The verification and validation process uses a combination of the common methods of test, analysis, demonstration, and inspection/review, at different degrees of automation to ensure that the implementation corresponds to and satisfies the requirements. The means of verification and validation with example uses are given in Table 2.6.

Table 2.6: Means of verification and validation.

Means	Characteristics	Example use in a verification context	Example use in a validation context
Test	Quantitative evaluation of the system outputs resulting from a defined set of inputs in real or simulated, controlled conditions Special test equipment or instrumentation to obtain accurate quantitative data to be analyzed	Verification of performance requirements in linear and nonlinear time domain assessment Verification of functional requirements in automated model-in-the-loop tests Verification of functional requirements in flight tests	Validation of system behavior and performance against actual operational scenarios
Analysis	Application of mathematical calculations to a system model under defined conditions to show theoretical compliance with specific requirements	Verification of stability and robustness requirements in closed-loop system frequency domain assessment	Validation of proposed system architecture with respect to system environment and interfaces requirements

Means	Characteristics	Example use in a verification context	Example use in a validation context
Demonstration	Quantitative or qualitative evaluation of the system operation against desired operational and observable characteristics No or little test equipment or instrumentation	Qualitative assessment by test pilot of flight test performance in a representative operational scenario	"Virtual flight testing" of the control system and its components using model-, hardware-, and operator-in-the-loop simulations to build confidence in the proposed design
Inspection/ Review	Examination of non-functional system or component attributes, or their representation, e.g., physical dimensions	Model review for verification of architecture requirements, and modeling guideline requirements	Review of system requirements or architecture candidates with respect to operational concept intentions

Hardware-in-the-Loop (HIL) Verification and Validation

HIL demonstrations validate the system operation in the integrated laboratory setup for various operational scenarios.

Aircraft-in-the-Loop (AIL) Verification and Validation

AIL simulation demonstrations validate the system operation in its integrated form for various operational scenarios.

Flight Test Verification and Validation

The purpose of the flight testing is to verify the performance and correct functionality of the system with respect to the system-level requirements, as well as to validate the system operation through demonstration in its intended operational environment.

Function-specific flight tests sequentially verify the different control loops. First, the correct functionality and performance of the inner loop controller is verified with preprogrammed command signals. After basic verification of the inner loops, the automatic flight path control loops are stepwise engaged and verified. System-level tests verify mode changes, target captures, and external command tracking.

Flight demonstrations validate the system operation in its operational environment under various operational scenarios.

3 Flight Dynamics Preliminaries

This chapter presents the aircraft, environment and subsystem dynamics fundamentals that form the basis for the development of the control strategy in Chapter 5.

The control of the speed, vertical, and lateral flight path is based on a dynamic inversion of the aircraft translational dynamics, which is derived in detail, along with a brief derivation of the rotational dynamics which produces the flight path transverse forces. External forces and moments acting upon the aircraft are also derived. Relevant models of static and dynamic atmosphere are discussed, as well as their coupling to the flight dynamics. The chapter includes the principal dynamics of relevant parts of the flight control system itself, such as the dynamics of the actuation, sensor, and propulsion systems. From the point of view of the flight path controller, the closed inner loop and actuator dynamics together form the actuation system. This actuation system and its dynamics are highly aircraft-specific, and this chapter discusses principal models capturing relevant dynamics for control law prototyping and requirements derivation. The aircraft-specific parameter design and closed loop assessment must be based on models of the actual inner loop, aircraft as well as subsystem dynamics, which are not detailed in this chapter.

Furthermore, an analysis of the energy and force constraints that preclude arbitrary flight path inclination, acceleration, and curvature is presented, along with derived vertical and lateral flight path control constraints to enable deterministic prioritization and tradeoffs in the control of the flight path.

3.1 Aircraft Equations of Motion

This section develops the six degree of freedom equations of motion for a rigid-body aircraft. The equations are formulated as a set of first-order nonlinear ordinary differential equations on the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3.1)$$

with the states describing the motion of the aircraft divided into groups defining the translational and rotational motion, as well as the aircraft's attitude and position. For the design of a flight path controller, the primary dynamics of interest is the translational dynamics; however, the faster rotational dynamics, by which the necessary flight path transverse forces are achieved, is also derived and analyzed. Although of minor importance for the flight path control, the

absolute position of the aircraft (except for the density, i.e., altitude dependency of aerodynamic parameters) and its propagation are included summarily for sake of completeness.

Multiple formulations of the equations of motion are possible, and a form may be chosen that best suits the task at hand. Detailed derivations may be found in [3, 4, 5].

Conservation of Linear and Angular Momentum

The differential equations describing the motion of the aircraft may be derived from the second law of Newton, according to which the rate of change of the linear momentum $\vec{\mathbf{p}} \in \mathbb{R}^3$ of an object in an inertial frame is equal to the net force $\sum \vec{\mathbf{F}} \in \mathbb{R}^3$ acting upon it,

$$\sum \vec{\mathbf{F}} = \left(\frac{d}{dt}\right)^I (\vec{\mathbf{p}}(t))^I = \left(\frac{d}{dt}\right)^I \int_m (\vec{\mathbf{V}}^P(t))^I \cdot dm \quad (3.2)$$

where $(\vec{\mathbf{V}}^P(t))^I \in \mathbb{R}^3$ is the velocity of a point P with mass dm , relative to the inertial frame I . Analogously, the rate of change of angular momentum $\vec{\mathbf{H}}^O \in \mathbb{R}^3$ about a point O is equal to the applied net torque $\sum \vec{\mathbf{M}}^O \in \mathbb{R}^3$ with respect to that point,

$$\sum \vec{\mathbf{M}}^O = \left(\frac{d}{dt}\right)^I \vec{\mathbf{H}}^O = \left(\frac{d}{dt}\right)^I \int_m \vec{\mathbf{r}}^P(t) \times (\vec{\mathbf{V}}^P(t))^I \cdot dm \quad (3.3)$$

where again $(\vec{\mathbf{V}}^P(t))^I \in \mathbb{R}^3$ is the velocity relative to the inertial frame I of a point P with mass dm at a position $\vec{\mathbf{r}}^P(t) \in \mathbb{R}^3$.

Assumptions

The derivation of the equations of motion underlies the following assumptions.

Assumption 3.1 (Rigid Body). The aircraft is considered a rigid body, i.e., the mass distribution of the aircraft is assumed to be quasi-stationary,

$$\left(\frac{d}{dt}\right)^B (\vec{\mathbf{r}}^{RP}) = (\dot{\vec{\mathbf{r}}^{RP}})^B = \vec{\mathbf{0}}, \quad (3.4)$$

where $\vec{\mathbf{r}}^{RP}(t) \in \mathbb{R}^3$ denotes the relative position of any point P to the aircraft reference point R . Hence, any effects of aircraft mass elements moving relative to the reference point are neglected.

Assumption 3.2 (Stationary Mass). The rate of change of the mass is negligible compared to the total mass of the aircraft, i.e., the system is assumed to have a quasi-stationary mass m ,

$$\left(\frac{d}{dt}\right)m = \dot{m} = 0. \quad (3.5)$$

Hence, only changes in the linear momentum of the system due to forces applied to the system are considered.

Assumption 3.3 (Inertial Frame Validity). The Earth-Centered Inertial (ECI) frame (see Appendix B) is considered a valid inertial (nonaccelerating) reference frame, i.e. the residual accelerations due to the Earth's orbit around the Sun are neglected.

Assumption 3.4 (Constant Earth Rotation). The angular velocity $\vec{\omega}^{IE} \in \mathbb{R}^3$, describing the rotation of the Earth-Centered Earth-Fixed (ECEF) frame with respect to the ECI frame, is assumed to have a constant value and direction, i.e. the small variations in the Earth's rotation rate and rotational axis are neglected,

$$\left(\frac{d}{dt}\right)^I (\vec{\omega}^{IE}) = (\dot{\vec{\omega}}^{IE})^I = 0. \quad (3.6)$$

3.1.1 Translational Equations of Motion

Under the Assumption 3.2, Eq. (3.2) can be written

$$\sum \vec{F} = \int_m (\vec{a}^P)^{II} \cdot dm, \quad (3.7)$$

with $(\vec{a}^P)^{II}$ being the absolute acceleration of a point P at position \vec{r}^P with infinitesimal mass dm . Considering such an arbitrary point P , at a relative position \vec{r}^{RP} to the aircraft reference point R (moving with the velocity $(\vec{V}^R)^I$ with respect to the inertial frame I), the absolute acceleration is given by differentiating the velocity vector $(\vec{V}^P)^I$ with respect to the inertial frame,

$$\begin{aligned} (\vec{a}^P)^{II} &= \left(\frac{d}{dt}\right)^I (\vec{V}^P)^I \\ &= \left(\frac{d}{dt}\right)^I [(\vec{V}^R)^I + (\vec{V}^{RP})^I] \\ &= \left(\frac{d}{dt}\right)^I [(\vec{V}^R)^E + (\vec{\omega}^{IE}) \times (\vec{r}^P) + (\vec{V}^{RP})^B + (\vec{\omega}^{IB}) \times (\vec{r}^{RP})] \\ &= \left(\frac{d}{dt}\right)^I [(\vec{V}^R)^E + (\vec{\omega}^{IE}) \times (\vec{r}^P)] + \left(\frac{d}{dt}\right)^B [(\vec{\omega}^{IB}) \times (\vec{r}^{RP})] \\ &\quad + (\vec{\omega}^{IB}) \times [(\vec{\omega}^{IB}) \times (\vec{r}^{RP})] \\ &= (\dot{\vec{V}}^R)^{EI} + (\vec{\omega}^{IE}) \times (\vec{V}^R)^E + (\vec{\omega}^{IE}) \times [(\vec{\omega}^{IE}) \times (\vec{r}^R)] \\ &\quad + (\dot{\vec{\omega}}^{IB})^B \times (\vec{r}^{RP}) + (\vec{\omega}^{IB}) \times [(\vec{\omega}^{IB}) \times (\vec{r}^{RP})] \\ &= (\dot{\vec{V}}^R)^{EO} \\ &\quad + (\vec{\omega}^{EO}) \times (\vec{V}^R)^E \\ &\quad + 2 \cdot (\vec{\omega}^{IE}) \times (\vec{V}^R)^E + (\vec{\omega}^{IE}) \times [(\vec{\omega}^{IE}) \times (\vec{r}^R)] \\ &\quad + (\dot{\vec{\omega}}^{IB})^B \times (\vec{r}^{RP}) + (\vec{\omega}^{IB}) \times [(\vec{\omega}^{IB}) \times (\vec{r}^{RP})]. \end{aligned} \quad (3.8)$$

On the right-hand side, the terms may be interpreted as follows:

- the first term describes the rate of change of the velocity of the reference point R over the surface of the Earth with respect to the North-East-Down (NED) system O ,
- the second term is the acceleration of the reference point R due to the transport rate $(\vec{\omega}^{EO})$,
- the third and fourth terms are the Coriolis and centrifugal accelerations of the reference point R due to the rotation of the Earth $(\vec{\omega}^{IE})$, and
- the fifth and sixth terms are the relative acceleration and centrifugal acceleration at the point P due to the angular acceleration $(\dot{\vec{\omega}}^{IB})^B$ and angular velocity $(\vec{\omega}^{IB})$ of the rigid body.

Inserting the result from Eq. (3.8) into Eq. (3.7), splitting the position vector of the point P into its components relative to the center of gravity G ,

$$(\vec{r}^{RP}) = (\vec{r}^{RG}) + (\vec{r}^{GP}), \quad (3.9)$$

and utilizing the definition of the center of gravity,

$$\int_m (\vec{r}^{GP}) \cdot dm = 0, \quad (3.10)$$

Eq. (3.7) can be written as

$$\begin{aligned} \sum \vec{F} = m \cdot & \left[(\dot{\vec{V}}^R)^{EO} + (\vec{\omega}^{EO}) \times (\vec{V}^R)^E + 2 \cdot (\vec{\omega}^{IE}) \times (\vec{V}^R)^E \right. \\ & + (\vec{\omega}^{IE}) \times [(\vec{\omega}^{IE}) \times (\vec{r}^R)] + (\dot{\vec{\omega}}^{IB})^B \times (\vec{r}^{RG}) \\ & \left. + (\vec{\omega}^{IB}) \times [(\vec{\omega}^{IB}) \times (\vec{r}^{RG})] \right]. \end{aligned} \quad (3.11)$$

For the analysis of the flight path dynamics, a formulation of the translational equations of motion considering the kinematic frame K is desired. Hence, the acceleration relative to the NED frame O is split up to include the kinematic acceleration due to flight path dynamics,

$$(\dot{\vec{V}}_K^R)^{EO} = (\dot{\vec{V}}_K^R)^{EK} + (\vec{\omega}^{OK}) \times (\vec{V}_K^R)^E. \quad (3.12)$$

$(\dot{\vec{V}}_K^R)^{EK}$ is the kinematic acceleration relative to the kinematic frame K , and $(\vec{\omega}^{OK}) \times (\vec{V}_K^R)^E$ is the acceleration due to a change in flight path, i.e. a rotation of the kinematic frame K with respect to the NED frame O . The kinematic motion of the reference point R is now given by

$$\begin{aligned} & (\dot{\vec{V}}_K^R)^{EK} + (\vec{\omega}^{OK}) \times (\vec{V}_K^R)^E \\ & = \frac{1}{m} \cdot \sum \vec{F}^R - (\vec{\omega}^{EO}) \times (\vec{V}_K^R)^E - 2 \cdot (\vec{\omega}^{IE}) \times (\vec{V}_K^R)^E \\ & \quad - (\vec{\omega}^{IE}) \times [(\vec{\omega}^{IE}) \times (\vec{r}^R)] - (\dot{\vec{\omega}}^{IB})^B \times (\vec{r}^{RG}) \\ & \quad - (\vec{\omega}^{IB}) \times [(\vec{\omega}^{IB}) \times (\vec{r}^{RG})]. \end{aligned} \quad (3.13)$$

The Coriolis and centrifugal accelerations only have a significant influence on the aircraft dynamics for higher supersonic speeds or when accurate long-distance navigation simulation is required [5]. For the analysis and simulation of subsonic flight in a sufficiently small area, which is applicable to the flight path controller developed in this thesis, it can be assumed that the Earth is non-rotating, i.e. $(\vec{\omega}^{IE}) = \vec{0}$, and flat, i.e. $(\vec{\omega}^{EO}) = \vec{0}$. Furthermore, if the center of gravity is chosen as the reference point, i.e. $\vec{r}^{RG} = \vec{0}$, Eq. (3.13) is reduced to the translational equations of motion for a point mass aircraft,

$$(\dot{\vec{V}}_K^G)^{EK} + (\vec{\omega}^{OK}) \times (\vec{V}_K^G)^E = \frac{1}{m} \cdot \sum \vec{F}^G. \quad (3.14)$$

Multiple choices are possible for the translational states and notation frame. Using a Cartesian notation in the NED frame, the translational states become the north $u_K^G \in \mathbb{R}$, east $v_K^G \in \mathbb{R}$ and vertical $w_K^G \in \mathbb{R}$ velocity components,

$$(\vec{V}_K^G)_O^E = \begin{bmatrix} u_K^G \\ v_K^G \\ w_K^G \end{bmatrix}^E. \quad (3.15)$$

Using a polar notation in the NED frame, the translational states become the kinematic velocity magnitude $V_K^G \in \mathbb{R}$, the course angle $\chi_K^G \in \mathbb{R}$ and the flight path angle $\gamma_K^G \in \mathbb{R}$,

$$(\vec{V}_K^G)_O^E = \begin{bmatrix} V_K^G \\ \chi_K^G \\ \gamma_K^G \end{bmatrix}^E, \quad (3.16)$$

with

$$V_K^G = \|(\vec{V}_K^G)_O^E\|_2 = \sqrt{[(u_K^G)_O^E]^2 + [(v_K^G)_O^E]^2 + [(w_K^G)_O^E]^2}, \quad (3.17)$$

$$\chi_K^G = \tan^{-1} \left(\frac{(v_K^G)_O^E}{(u_K^G)_O^E} \right), \quad (3.18)$$

$$\gamma_K^G = -\tan^{-1} \left(\frac{(w_K^G)_O^E}{\sqrt{[(u_K^G)_O^E]^2 + [(v_K^G)_O^E]^2}} \right). \quad (3.19)$$

The course angle χ_K^G and flight path angle γ_K^G are the rotation angles between the NED frame O and the kinematic frame K . Noted in the kinematic frame, Eq. (3.14) is given by

$$\begin{bmatrix} \dot{V}_K^G \\ 0 \\ 0 \end{bmatrix}_K^{EK} + \begin{bmatrix} -\dot{\chi}_K^G \sin \gamma_K^G \\ \dot{\gamma}_K^G \\ \dot{\chi}_K^G \cos \gamma_K^G \end{bmatrix} \times \begin{bmatrix} V_K^G \\ 0 \\ 0 \end{bmatrix}_K^E = \frac{1}{m} \cdot \begin{bmatrix} X_T^G \\ Y_T^G \\ Z_T^G \end{bmatrix}_K, \quad (3.20)$$

where $X_T^G \in \mathbb{R}$, $Y_T^G \in \mathbb{R}$, and $Z_T^G \in \mathbb{R}$ are the total forces at the center of gravity G . Expanding the left-hand side of Eq. (3.20) gives

$$\dot{V}_K^G = \frac{1}{m} X_T^G, \quad (3.21)$$

$$\dot{\chi}_K^G V_K^G \cos \gamma_K^G = \frac{1}{m} Y_T^G, \quad (3.22)$$

$$-\dot{\gamma}_K^G V_K^G = \frac{1}{m} Z_T^G \quad (3.23)$$

3.1.2 Rotational Equations of Motion

Analog to the derivation of the translational equations of motion, using the simplifying assumptions above and selecting the center of gravity as the reference point, Eq. (3.3) is reduced to

$$\sum \vec{\mathbf{M}}^G = (\vec{\mathbf{I}}^G)_{BB} \cdot (\dot{\vec{\omega}}^{IB})^B + (\vec{\omega}^{IB}) \times [(\vec{\mathbf{I}}^G)_{BB} \cdot (\vec{\omega}^{IB})] \quad (3.24)$$

where $(\vec{\mathbf{I}}^G)_{BB} \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of the aircraft with respect to its center of gravity G ,

$$(\vec{\mathbf{I}}^G)_{BB} = \begin{bmatrix} I_{xx}^G & -I_{xy}^G & -I_{xz}^G \\ -I_{xy}^G & I_{yy}^G & -I_{yz}^G \\ -I_{xz}^G & -I_{yz}^G & I_{zz}^G \end{bmatrix}_{BB}. \quad (3.25)$$

Solving the Eq. (3.24) for the rate of change of $\vec{\omega}^{IB}$ yields the rotational equations of motion,

$$(\dot{\vec{\omega}}^{IB})^B = (\vec{\mathbf{I}}^G)_{BB}^{-1} \cdot \left[\sum \vec{\mathbf{M}}^G - (\vec{\omega}^{IB}) \times [(\vec{\mathbf{I}}^G)_{BB} \cdot (\vec{\omega}^{IB})] \right], \quad (3.26)$$

or

$$\begin{bmatrix} \dot{p}_K \\ \dot{q}_K \\ \dot{r}_K \end{bmatrix}_B^B = (\vec{\mathbf{I}}^G)_{BB}^{-1} \cdot \left[\begin{bmatrix} L^G \\ M^G \\ N^G \end{bmatrix}_B - \begin{bmatrix} p_K \\ q_K \\ r_K \end{bmatrix} \times (\vec{\mathbf{I}}^G)_{BB} \cdot \begin{bmatrix} p_K \\ q_K \\ r_K \end{bmatrix}_B \right], \quad (3.27)$$

here with components noted in the body-fixed frame B . In equation (3.27), L denotes the body-axis roll moment, M the pitching moment, and N the yaw moment about the center of gravity G , where p , q , and r denotes the body-axis roll, pitch and yaw rates, respectively.

3.1.3 Attitude Propagation

The attitude of the aircraft with respect to the NED system is described by the three Euler angles; the bank angle Φ , the pitch angle Θ , and the heading or azimuth angle Ψ . The rate of change or the propagation of the aircraft attitude can be derived from the strapdown equation [5]

$$(\boldsymbol{\Omega}^{OB})_{BB} = \mathbf{M}_{BO} \cdot \dot{\mathbf{M}}_{BO}^T, \quad (3.28)$$

with $(\boldsymbol{\Omega}^{OB})_{BB}$ containing the roll, pitch, and yaw rate components of the aircraft angular velocity vector,

$$(\boldsymbol{\Omega}^{OB})_{BB} = \begin{bmatrix} 0 & -r_K & q_K \\ r_K & 0 & -p_K \\ -q_K & p_K & 0 \end{bmatrix}_{BB}, \quad (3.29)$$

$\mathbf{M}_{BO}(\Psi, \theta, \Phi)$ being the transformation matrix between the body-fixed and NED systems, and $\dot{\mathbf{M}}_{BO}(\dot{\Psi}, \dot{\theta}, \dot{\Phi}, \Psi, \theta, \Phi)$ its derivative. By comparing the left- and right-hand side of Eq. (3.28), the attitude equations of motion are given by

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \frac{\sin \Phi}{\cos \theta} & \frac{\cos \Phi}{\cos \theta} \end{bmatrix} \cdot \begin{bmatrix} p_K \\ q_K \\ r_K \end{bmatrix}_B. \quad (3.30)$$

Eq. (3.30) has a singularity for pitch angles $\theta = \pm\pi/2$. This problem could be solved by describing the attitude of the aircraft using quaternions [5].

3.1.4 Position Propagation

Depending on the application, the position of the aircraft \mathbf{r}^G can be given in a number of different ways:

- With respect to the ECEF frame E either in
 - Cartesian coordinates $[x, y, z]_E^T$, or
 - World Geodetic System 84 (WGS84) coordinates as the geodetic latitude μ , longitude λ and altitude h ;
- With respect to a local Cartesian navigation frame N with coordinates $[x, y, z]_N^T$.

The aircraft position changes with the kinematic velocity

$$(\vec{\mathbf{V}}_K^G)_O^E = \begin{bmatrix} u_K^G \\ v_K^G \\ w_K^G \end{bmatrix}_O^E = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_O^E = \begin{bmatrix} V_N \\ V_E \\ -\dot{h} \end{bmatrix}_O^E = V_K^G \cdot \begin{bmatrix} \cos \chi_K^G \cos \gamma_K^G \\ \sin \chi_K^G \cos \gamma_K^G \\ -\sin \gamma_K^G \end{bmatrix}. \quad (3.31)$$

The rate of change of the position of the aircraft with respect to the ECEF frame is hence, in Cartesian coordinates, given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_O^E = \begin{bmatrix} u_K^G \\ v_K^G \\ w_K^G \end{bmatrix}_O^E, \quad (3.32)$$

and in WGS84 coordinates by

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\mu} \\ \dot{h} \end{bmatrix}_O^E = V_K^G \begin{bmatrix} \cos \chi_K^G \cos \gamma_K^G \\ (N_\mu + h) \cos \mu \\ \frac{\sin \chi_K^G \cos \gamma_K^G}{M_\mu + h} \\ \sin \gamma_K^G \end{bmatrix}_O^E, \quad (3.33)$$

with

$$N_\mu = \frac{a}{\sqrt{1 - e^2 \sin^2 \mu}}, \quad (3.34)$$

being the radius of curvature in the prime vertical, and

$$M = N_\mu \cdot \frac{1 - e^2}{1 - e^2 \sin^2 \mu}, \quad (3.35)$$

being the meridian radius of curvature, computed from the length of the semi-major axis a and the first eccentricity e of the reference ellipsoid.

For local small-extent navigation purposes, a Cartesian navigation frame can be derived from the NED frame, with its origin at a fixed point on the Earth's surface and rotated about the NED z -axis by an angle χ_N , giving the position equations

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_N^E = \mathbf{M}_{NO} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_O^E = \mathbf{M}_{NO} \begin{bmatrix} u_K^G \\ v_K^G \\ w_K^G \end{bmatrix}_O^E = \mathbf{M}_{NO} \begin{bmatrix} V_K^G \cos \chi_K^G \cos \gamma_K^G \\ V_K^G \sin \chi_K^G \cos \gamma_K^G \\ -V_K^G \sin \gamma_K^G \end{bmatrix}, \quad (3.36)$$

with the transformation matrix \mathbf{M}_{NO} between the NED and navigation frames given by

$$\mathbf{M}_{NO} = \begin{bmatrix} \cos \chi_N & \sin \chi_N & 0 \\ -\sin \chi_N & \cos \chi_N & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.37)$$

3.2 External Forces and Moments

The derivation of the translation and rotational equations of motion only considered the total net force $\sum \vec{\mathbf{F}}_T^G$ and moment $\sum \vec{\mathbf{M}}_T^G$ acting on the system, according to the second law of Newton, without further consideration of their respective composition and sources. In this section, models of different fidelity for the external forces and moments acting upon the aircraft are discussed, with the appropriate fidelity required for the design of the flight path controller analyzed in more detail.

The relevant forces are divided into the gravitational force (index G), aerodynamic forces (index A) and propulsive forces (index P),

$$\sum \vec{\mathbf{F}}_T^G = \vec{\mathbf{F}}_G^G + \vec{\mathbf{F}}_A^G + \vec{\mathbf{F}}_P^G. \quad (3.38)$$

The moments are divided into aerodynamic and propulsive moments (indices analog to forces),

$$\sum \vec{\mathbf{M}}_T^G = \vec{\mathbf{M}}_A^G + \vec{\mathbf{M}}_P^G. \quad (3.39)$$

Other forces and moments that are either neglectable, e.g., aerostatic forces, or temporarily acting on the aircraft, e.g., landing gear forces and moments, are excluded here, since they are of no relevance for the design of the flight path controller. Further, according to Assumption 3.1, the aircraft is considered a rigid body, thus neglecting any structural dynamics and aeroelastic forces.

3.2.1 Gravitational Force

The gravitational force acts at the center of gravity G of the aircraft,

$$\vec{\mathbf{F}}_G^G = m \cdot \vec{\mathbf{g}}^G, \quad (3.40)$$

where m is the aircraft mass and $\vec{\mathbf{g}}^G$ the gravitational vector. The gravitational vector may be modeled either as a constant vector in the NED frame, as a function of the geopotential altitude, or using more complex approaches taking the ellipsoid form of the Earth into account, see e.g. [5]. For the operational envelope of the aircraft configurations considered for this thesis, modelling the gravitational force as constant is sufficient, i.e.

$$(\vec{\mathbf{F}}_G^G)_o = m \cdot g_0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_o, \quad (3.41)$$

where g_0 is the gravitational constant.

3.2.2 Aerodynamic Forces and Moments

The aerodynamic forces and moments result from the airflow around the aircraft and are acting upon and about the aerodynamic reference point A , located at a position $\vec{\mathbf{r}}^{GA}$ relative to the aircraft center of gravity G . The aerodynamic forces are given by

$$(\vec{\mathbf{F}}_A^A)_A = \begin{bmatrix} X_A^A \\ Y_A^A \\ Z_A^A \end{bmatrix}_A = \begin{bmatrix} -D \\ Q \\ -L \end{bmatrix} = \bar{q} \cdot S \cdot \begin{bmatrix} -C_D \\ C_Q \\ -C_L \end{bmatrix}, \quad (3.42)$$

where D is the drag force, Q is the side force, and L is the lift force acting along the axes of the aerodynamic frame A , C_D , C_Q and C_L are the corresponding dimensionless coefficients, $\bar{q} = 0.5 \cdot \rho \cdot V_A^2$ is the dynamic pressure, dependent on the air density ρ and the aerodynamic velocity V_A , and S the reference wing area of the aircraft. It is common to denote the aerodynamic forces in the aerodynamic frame A (see definition in Appendix B), since the directions of the lift and drag forces are defined to be perpendicular and parallel to the aerodynamic velocity vector, respectively. Note that the drag and lift forces are acting in the opposite direction of the corresponding axis of the coordinate system, hence the negative sign.

The aerodynamic moments acting about the aerodynamic reference point A are given by

$$(\vec{\mathbf{M}}_A^A)_B = \begin{bmatrix} L_A^A \\ M_A^A \\ N_A^A \end{bmatrix}_A = \bar{q} \cdot S \cdot \begin{bmatrix} s \cdot C_l \\ \bar{c} \cdot C_m \\ s \cdot C_n \end{bmatrix}, \quad (3.43)$$

where L is the rolling moment, M is the pitching moment, and N is the yawing moment, C_l , C_m and C_n are the corresponding dimensionless coefficients, s is the semi-span, and \bar{c} is the mean aerodynamic chord.

The aerodynamic coefficients are dependent on the specific aircraft and its aerodynamics, and may be modeled as functions of different parameters, for example, angle of attack, Mach number, and control surface deflections, depending on the scope and fidelity of the underlying aerodynamic analysis. For translational dynamics analysis, the drag coefficient C_D is typically described by a simplified model as the sum of the parasitic or zero-lift drag and lift-induced drag,

$$C_D = C_{D_0} + k \cdot C_L^2, \quad (3.44)$$

with C_{D_0} being the zero-lift drag coefficient, k the lift-induced drag coefficient factor (dependent on the aircraft aspect ratio and the span efficiency), and C_L the lift coefficient. The drag is thus a nonlinear function of the lift coefficient and therefore of the rotational dynamics of the aircraft.

The aerodynamic forces and moments act in and about the aerodynamic reference point A , whereas the translational and rotational equations of motion are formulated for the aircraft's center of gravity. The application point of the aerodynamic force may be shifted to the center of gravity,

$$(\vec{\mathbf{F}}_A^A)_A = (\vec{\mathbf{F}}_A^G)_A, \quad (3.45)$$

whereas to obtain the aerodynamic moments about the center of gravity, the relative position of the aerodynamic reference point needs to be considered,

$$(\vec{\mathbf{M}}_A^G)_B = (\vec{\mathbf{M}}_A^A)_B + \mathbf{r}^{GA} \times (\vec{\mathbf{F}}_A^A)_A. \quad (3.46)$$

3.2.3 Propulsion Forces and Moments

The propulsion forces and moments result from the propulsion sources of the aircraft that generates the thrust necessary to maintain the motion through the air, and their relative positions to the aircraft center of gravity and installation angles.

Consider a single propulsion system with air inlet at a relative position $\vec{\mathbf{r}}^{GT_I}$ and outlet at relative position $\vec{\mathbf{r}}^{GT_O}$, with a local propulsion frame P , defined by the installation angles σ and κ . The installed net propulsion force acting on the aircraft center of gravity, is given by the vector sum of the inlet force $\vec{\mathbf{F}}_{P,T_I}^{T_I}$ and the outlet force $\vec{\mathbf{F}}_{P,T_O}^{T_O}$, given by

$$\begin{aligned}
 (\vec{\mathbf{F}}_P^G)_B &= \mathbf{M}_{BA} \cdot (\vec{\mathbf{F}}_{P,T_I}^{T_I})_A + \mathbf{M}_{BP} \cdot (\vec{\mathbf{F}}_{P,T_O}^{T_O})_P \\
 &= \mathbf{M}_{BA} \cdot \begin{bmatrix} -T_I \\ 0 \\ 0 \end{bmatrix}_A + \mathbf{M}_{BP} \cdot \begin{bmatrix} T_O \\ 0 \\ 0 \end{bmatrix}_P,
 \end{aligned} \tag{3.47}$$

where $\mathbf{M}_{BA}(\alpha_A, \beta_A)$ denotes the transformation matrix between the body-fixed and the aerodynamic frame, and $\mathbf{M}_{BP}(\kappa, \sigma)$

$$\mathbf{M}_{BP} = \begin{bmatrix} \cos \kappa \cos \sigma & -\sin \kappa & \cos \kappa \sin \sigma \\ \sin \kappa \cos \sigma & \cos \kappa & \sin \kappa \sin \sigma \\ -\sin \sigma & 0 & \cos \sigma \end{bmatrix} \tag{3.48}$$

denotes the transformation matrix between the body-fixed and the propulsion frame P.

The propulsion moment at the center of gravity consists of gyroscopic moments due to rotating parts of the propulsion system and on moments dependent on the relative position of the intake and output forces,

$$\begin{aligned}
 (\vec{\mathbf{M}}_P^G)_B &= (\vec{\mathbf{M}}_{Gyro}^G)_B + (\vec{\mathbf{r}}^{GT_I})_B \times (\vec{\mathbf{F}}_{P,T_I}^{T_I})_B + (\vec{\mathbf{r}}^{GT_O})_B \times (\vec{\mathbf{F}}_{P,T_O}^{T_O})_B \\
 &= (\vec{\mathbf{M}}_{Gyro}^G)_B + (\vec{\mathbf{r}}^{GT_I})_B \times \mathbf{M}_{BA} \cdot (\vec{\mathbf{F}}_{P,T_I}^{T_I})_A + (\vec{\mathbf{r}}^{GT_O})_B \times \mathbf{M}_{BP} \cdot (\vec{\mathbf{F}}_{P,T_O}^{T_O})_P
 \end{aligned} \tag{3.49}$$

3.2.4 Path Dynamics Couplings

The primary means to achieve a desired flight path curvature for a typical fixed-wing aircraft is by the magnitude and direction of the aircraft lift vector, and the primary means to achieve a desired flight path acceleration is by the magnitude of the thrust from the propulsion system. The lift, mainly influenced by the angle of attack, i.e. the rotational dynamics, couples into the tangential acceleration via lift-induced drag, whereas the propulsion dynamics couples into the path perpendicular dynamics via the geometric relation between the thrust forces and the flight path.

From the flight path controller design perspective, it would be desired to consider the control of the path transverse and path tangential dynamics to be decoupled, i.e. consider the path curvature, controlled by the faster rotational dynamics of the aircraft, and the path acceleration, controlled by the slower propulsion dynamics, to be independent. This allows for the decoupled design of the nominal path control loops using SISO models of the closed inner loop and actuation dynamics.

Thrust Coupling into Path Transverse Dynamics

The outlet thrust force couples into the path transverse dynamics via the kinematic angles of attack, α_K , and sideslip, β_K , as well as the propulsion frame angles σ and κ ,

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix}_K = \mathbf{M}_{KB} \cdot \mathbf{M}_{BP} \cdot \begin{bmatrix} T_O \\ 0 \\ 0 \end{bmatrix}_P. \tag{3.50}$$

Assuming no lateral propulsion system angle κ , i.e. the y_P -axis of the propulsion system aligns with the y_B -axis of the body-fixed system, and using standard trigonometric small angle simplifications, Eq. (3.50) can be written as

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix}_K = T_O \cdot \begin{bmatrix} 1 \\ \alpha_K \mu_K - \beta_K + \sigma(\mu_K + \alpha_K \beta_K) \\ -\alpha_K - \beta_K \mu_K + \sigma(\alpha_K \beta_K \mu_K - 1) \end{bmatrix} \approx T_O \cdot \begin{bmatrix} 1 \\ -\beta_K \\ -\alpha_K - \sigma \end{bmatrix}, \quad (3.51)$$

from which can be concluded that the outlet force mainly affects the lateral path due to the kinematic sideslip, and the vertical path due to the sum of the kinematic angle of attack and propulsion angle σ . As noted in [21], a combined angle of attack and propulsion system angle of e.g., 20 degrees would result in ca. 35 percent of the thrust magnitude affecting the vertical path transverse force component. The thrust force is mainly countering the aircraft drag, hence its magnitude is on the order of that of the drag force, whereas the normal force is dominated by the lift force countering gravity, which is typically one order of magnitude or more compared to the drag force. For a lift-to-drag ratio of 10, the effective normal force disturbance due to thrust changes is thus 3.5 percent or less; for most flight conditions significantly less.

The thrust force further couples into the normal force indirectly via the thrust lever arm \vec{r}^{GT_O} , producing a pitching moment that affects the angle of attack and thus the normal force dynamics. For most aircraft, the propulsion system is installed such that the thrust lever arm is minimized. The normal force is directly controlled by the inner loop controller, at a much higher bandwidth than the propulsion dynamics; hence the remaining pitching moment can be treated as a disturbance into the normal force control. Alternatively, the coupling effect could be compensated by an elevator deflection, calculated from an inversion of the expected pitching moment.

To conclude, the thrust coupling into the path transverse forces can be argued as neglectable for the design of the path transverse and path tangential specific force controllers.

Lift Coupling into Path Tangential Dynamics

The path tangential acceleration is given by

$$\dot{V}_K^G = \frac{1}{m} (X_T^G)_K - g \sin \gamma_K^G = \frac{1}{m} T - \frac{1}{m} D - g \sin \gamma_K^G, \quad (3.52)$$

with the drag $D = \bar{q} S C_D$ according to Eq. (3.42). For small flight path angles, the path tangential acceleration is given by the difference between the thrust acceleration, controlled by the throttle setting, and the drag acceleration, a nonlinear function of dynamic pressure, angle of attack, pitch rate, and elevator deflection, see Eq. (3.50).

When maneuvering the aircraft by changing the magnitude (and direction) of the lift vector, the additional lift-induced drag affects the tangential acceleration. Up to a certain frequency, an automatic thrust controller can reject those drag disturbances, depending on the bandwidth of the propulsion system and desired throttle actuation dynamics, but for higher frequencies, the drag disturbances will not be sufficiently compensated by the propulsion system and the residual acceleration will disturb the velocity magnitude of the aircraft. The natural integration

of the velocity magnitude dynamics will however prevent most of the high-frequency parts of the drag disturbances from manifesting themselves into velocity deviations, which will have to be counteracted by the speed loop error controller. The velocity magnitude in turn couples further into the path dynamics kinematically, see Eq. (3.20), and via the dynamic pressure, see Eq. (3.43), hence the lift magnitude couples back into the lift dynamics. The natural low-pass filtering of the drag disturbances' effect on the velocity magnitude will provide for the higher bandwidth inner loop controller to counteract the effect on the rotational dynamics resulting from velocity disturbances.

With the sources of the external forces and moments analyzed, the Eq. (3.20) and the total forces therein can be written as

$$\begin{bmatrix} \dot{V}_K^G \\ 0 \\ 0 \end{bmatrix}^{EK} + \begin{bmatrix} -\dot{\chi}_K^G \sin \gamma_K^G \\ \dot{\gamma}_K^G \\ \dot{\chi}_K^G \cos \gamma_K^G \end{bmatrix} \times \begin{bmatrix} V_K^G \\ 0 \\ 0 \end{bmatrix}^E = \frac{1}{m} \cdot \begin{bmatrix} X_G^G + X_A^G + X_P^G \\ Y_G^G + Y_A^G + Y_P^G \\ Z_G^G + Z_A^G + Z_P^G \end{bmatrix}_K. \quad (3.53)$$

The total forces may be normalized by the mass m of the aircraft, giving the incremental specific forces for path control,

$$\Delta f_{x,K} = \frac{(X_G^G + X_A^G + X_P^G)}{m} = f_{x,K} - g \sin \gamma_K^G, \quad (3.54)$$

$$\Delta f_{y,K} = \frac{(Y_A^G + Y_P^G)}{m} = f_{y,K}, \quad (3.55)$$

$$\Delta f_{z,K} = \frac{(Z_G^G + Z_A^G + Z_P^G)}{m} = f_{z,K} + g \cos \gamma_K^G, \quad (3.56)$$

where $g \in \mathbb{R}$ represents the acceleration due to gravity, and $f_{x,K}$, $f_{y,K}$, and $f_{z,K}$ represent the specific forces including gravity, illustrated in Figure 3.1.

The incremental specific forces yield an acceleration along or curvature of the vertical or lateral flight path according to

$$\dot{V}_K^G = \Delta f_{x,K}, \quad (3.57)$$

$$\dot{\chi}_K^G = \frac{1}{V_K^G \cos \gamma_K^G} \cdot \Delta f_{y,K} \quad (3.58)$$

$$\dot{\gamma}_K^G = -\frac{1}{V_K^G} \cdot \Delta f_{z,K} \quad (3.59)$$

The incremental specific forces will later be used as the interface to the inner loop controllers, utilized by both the automatic flight path controller and the trajectory controller [85].

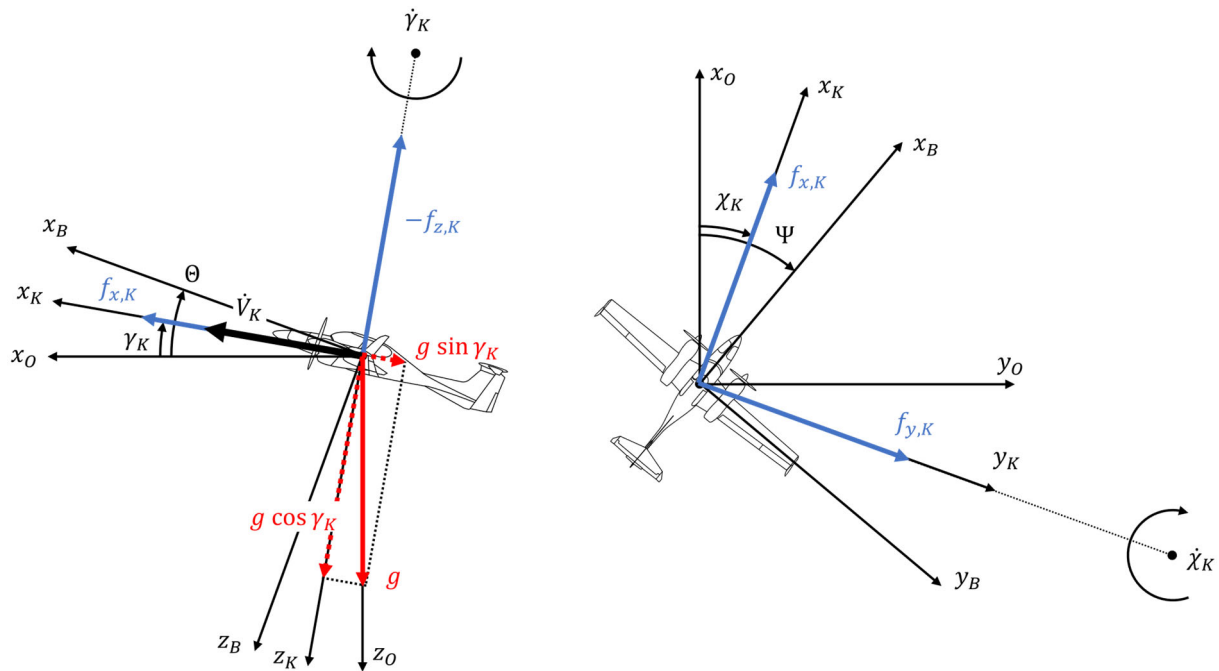


Figure 3.1: Specific forces for path control

3.3 Aircraft Environment

This section describes the modeling of the aircraft environment characteristics relevant to the design and assessment of the flight path controller. The flight path controller only operates when the aircraft is in flight, and the aircraft environment is given entirely by the atmosphere in which the aircraft is moving, its static and dynamic properties, and its impact on the path dynamics. The modeling of effects due to proximity of, or aircraft motion on the ground are not considered here.

The static atmosphere can be modeled as an ideal gas, with properties according to the International Standard Atmosphere (ISA) [105], defining the reference pressure, density, and temperature at sea level as well as their dependence on geopotential altitude.

The dynamic atmosphere, i.e., the motion of the air relative to the Earth's surface, can be modeled as composed of a dynamic wind field, superposed by continuous random turbulence, and discrete gusts. The characteristics of the random turbulence are given by properties defined in the frequency domain and the characteristics of the discrete gusts by properties defined in the time domain.

3.3.1 Wind

Figure 3.2 shows the kinematic velocity \vec{V}_K^G as the vector sum of the aerodynamic velocity vector \vec{V}_A^G , i.e. the motion relative to the air, and the wind velocity vector \vec{V}_W^G , i.e. the motion of the air relative to the Earth,

$$(\vec{V}_K^G)^E = (\vec{V}_A^G)^E + (\vec{V}_W^G)^E. \quad (3.60)$$

Noted for example in the NED frame, the kinematic, aerodynamic and wind velocity components become the respective north, east and vertical velocities,

$$\begin{bmatrix} u_K^G \\ v_K^G \\ w_K^G \end{bmatrix}_0^E = \begin{bmatrix} u_A^G \\ v_A^G \\ w_A^G \end{bmatrix}_0^E + \begin{bmatrix} u_W^G \\ v_W^G \\ w_W^G \end{bmatrix}_0^E. \quad (3.61)$$

The wind field may be modeled as a static, i.e., position- and time-invariant field, or a convective, i.e., position- and time-variant field,

$$(\vec{V}_W^G)^E = (\vec{v}_W^G(\vec{r}^G, t))^E, \quad (3.62)$$

where \vec{r}^G is the aircraft position.

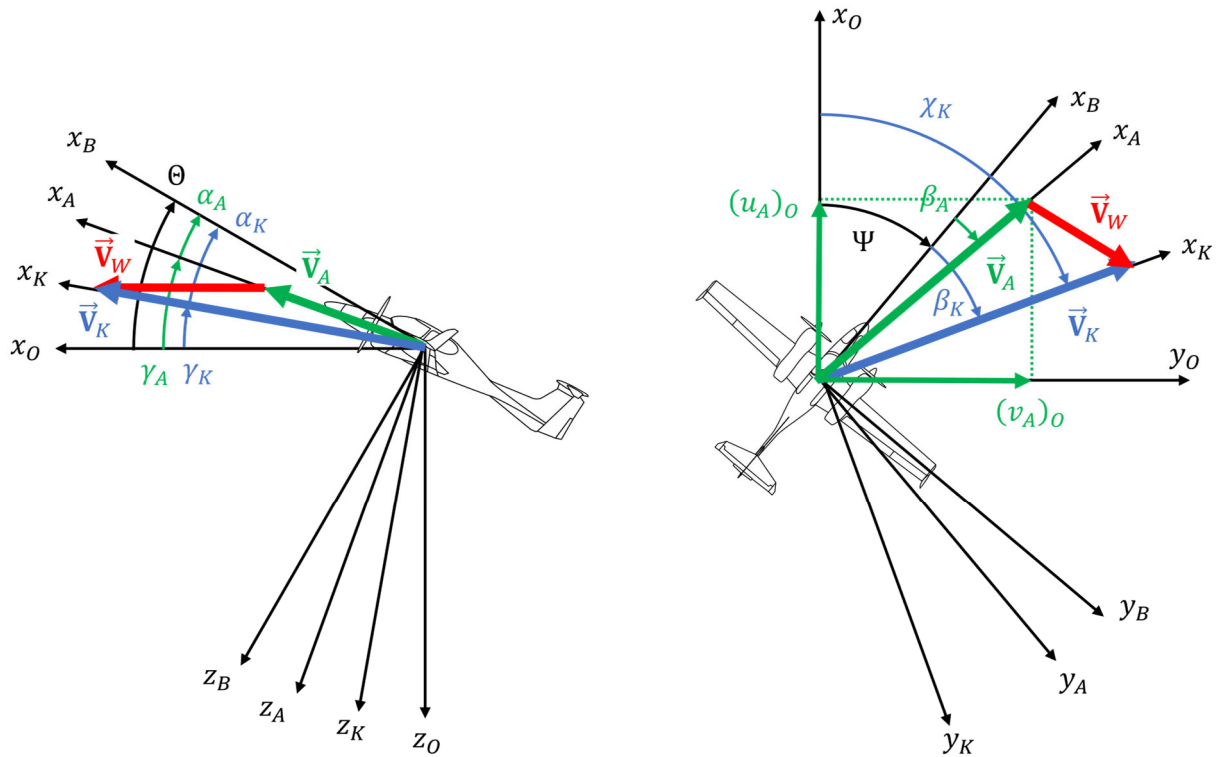


Figure 3.2: Wind velocity component and relation between kinematic and aerodynamic flight path angles.

Aerodynamic Flight Path Vector

Analog to the kinematic flight path vector, both Cartesian and polar notation of the aerodynamic flight path is possible. In Cartesian notation in the NED frame, the aerodynamic velocity components become the north $u_A^G \in \mathbb{R}$, east $v_A^G \in \mathbb{R}$ and vertical $w_A^G \in \mathbb{R}$ velocities,

$$(\vec{\mathbf{V}}_A^G)_O^E = \begin{bmatrix} u_A^G \\ v_A^G \\ w_A^G \end{bmatrix}_O^E. \quad (3.63)$$

In polar notation, using the aerodynamic velocity magnitude $V_A^G \in \mathbb{R}$, the aerodynamic course angle $\chi_A^G \in \mathbb{R}$, and aerodynamic flight path angle $\gamma_A^G \in \mathbb{R}$,

$$(\vec{\mathbf{V}}_A^G)_O^E = \begin{bmatrix} V_A^G \\ \chi_A^G \\ \gamma_A^G \end{bmatrix}_O^E, \quad (3.64)$$

with

$$V_A^G = \|(\vec{\mathbf{V}}_A^G)_O^E\|_2 = \sqrt{((u_A^G)_O^E)^2 + ((v_A^G)_O^E)^2 + ((w_A^G)_O^E)^2}, \quad (3.65)$$

$$\chi_A^G = \tan^{-1} \left(\frac{(v_A^G)_O^E}{(u_A^G)_O^E} \right). \quad (3.66)$$

$$\gamma_A^G = -\tan^{-1} \left(\frac{(w_A^G)_O^E}{\sqrt{((u_A^G)_O^E)^2 + ((v_A^G)_O^E)^2}} \right). \quad (3.67)$$

Aerodynamic Acceleration

The first order time derivative of the aerodynamic velocity vector, with respect to the NED frame, is given by

$$(\dot{\vec{\mathbf{V}}}_A^G)^{EO} = (\dot{\vec{\mathbf{V}}}_K^G)^{EO} - (\dot{\vec{\mathbf{V}}}_W^G)^{EO}. \quad (3.68)$$

The time derivative of the wind velocity vector is then given by

$$(\dot{\vec{\mathbf{V}}}_W^G)^{EO} = \left(\frac{\partial}{\partial t} \right)^O (\vec{\mathbf{V}}_W^G(\vec{\mathbf{r}}^G, t))^E + \left(\frac{\partial}{\partial \vec{\mathbf{r}}^G} \right)^O (\vec{\mathbf{V}}_W^G(\vec{\mathbf{r}}^G, t))^E \cdot \left(\frac{\partial}{\partial t} \right)^O \vec{\mathbf{r}}^G. \quad (3.69)$$

For the types of fixed-wing aircraft that the developed flight path controller is intended for, the velocity with which the aircraft passes through the wind field is typically much greater than any time-dependent changes of the field at any given point. The rate of change of the wind field relative to the aircraft is dominated by the change of the aircraft's position as it passes through the field, rather than by local time-dependent changes of the wind field itself. The wind field can thus for the purpose of the flight path controller development be regarded as time-invariant, and its rate of change only a function of the kinematic velocity, reducing Eq. (3.69) to

$$(\dot{\vec{\mathbf{V}}}_W^G)^{EO} = \left(\frac{\partial}{\partial \vec{\mathbf{r}}^G} \right)^O (\vec{\mathbf{V}}_W^G(\vec{\mathbf{r}}^G))^E \cdot \left(\frac{\partial}{\partial t} \right)^O \vec{\mathbf{r}}^G. \quad (3.70)$$

Different categories of position-dependent wind fields, such as wind shear, divergence free fields or rotating fields, result in characteristic structures of the gradient matrix [3]. For a simple static wind field, without any position-dependent gradient velocities, the time derivative is zero,

$$\left(\dot{\mathbf{V}}_W^G\right)^{EO} = \mathbf{0}, \quad (3.71)$$

from which it follows, according to Eq. (3.68), that the time derivative of the aerodynamic velocity equals that of the kinematic,

$$\left(\dot{\mathbf{V}}_A^G\right)^{EO} = \left(\dot{\mathbf{V}}_K^G\right)^{EO}. \quad (3.72)$$

The modeling of gusts and the reaction of the aircraft is further discussed in Section 3.3.3.

3.3.2 Continuous Turbulence

Several mathematical models of atmospheric turbulence exist, with applications in aircraft and control system design and simulation; most notably the Dryden and von Karman continuous turbulence models [106, 107]. The mathematically somewhat simpler Dryden model is feasible for most practical applications, and is the atmospheric model employed for the design and assessment of the flight path controller that is the subject of this thesis, whereas the more complex von Karman model is tuned to match observed data, and the preferred model when requiring structural analysis [107].

AS94900A [103] specifies requirements on turbulence modeling with exceedance probabilities for different gust intensities, with reference to MIL-HDBK-1797 [108] for mathematical models, which in turn refers to the turbulence models defined in [106]. MIL-HDBK-1797 [108] further defines turbulence severity levels for the analysis of flying qualities in atmospheric disturbances that roughly correspond to the exceedance probabilities defined in [103], see Figure 3.3.

The Dryden model is characterized by rational power spectral densities, parametrized by turbulence intensities and scale lengths, yielding a desired level of turbulence. The spatial frequency of the turbulence, denoted Ω , with unit radians per unit length, is related to the temporal frequency, ω , by the velocity V_0 with which the aircraft travels through the turbulence field,

$$\Omega = \frac{\omega}{V_0}. \quad (3.73)$$

The Dryden form of the spectra for the axial, u_g , lateral, v_g , and normal, w_g , turbulence velocity components are given by

$$\Phi_{u_g}(\Omega) = \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{(L_u\Omega)^2}, \quad (3.74)$$

$$\Phi_{v_g}(\Omega) = \sigma_v^2 \frac{2L_v}{\pi} \frac{1 + 12(L_v\Omega)^2}{[1 + 4(L_v\Omega)^2]^2}, \quad (3.75)$$

$$\Phi_{w_g}(\Omega) = \sigma_w^2 \frac{2L_w}{\pi} \frac{1 + 12(L_w\Omega)^2}{[1 + 4(L_w\Omega)^2]^2}, \quad (3.76)$$

where L_n is the length scale and σ_n is the root-mean-square intensity of the continuous turbulence, for $n = u, v, w$ [108].

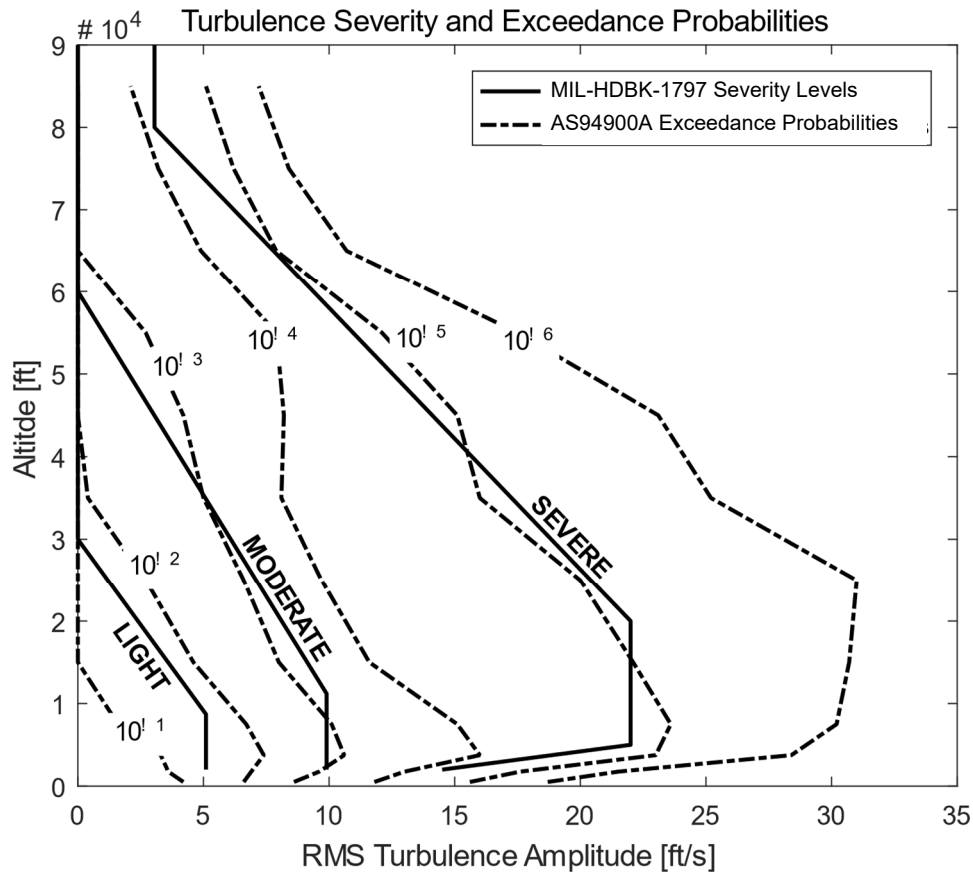


Figure 3.3: Turbulence levels and exceedance probabilities according to MIL-HDBK-1797 [108] and AS94900A [103], adopted from [108].

Medium to High Altitude Clear Air Turbulence

For altitudes above 2000 ft, the turbulence is assumed to be isotropic [108], i.e. the time averaged turbulence velocities have the same value at each location, with the axial, lateral, and normal mean-square intensities given by

$$\sigma_u^2 = \sigma_v^2 = \sigma_w^2, \quad (3.77)$$

and the length scales by

$$L_u = 2L_v = 2L_w, \quad (3.78)$$

with $L_u = 1750$ feet, according to [108].

Low Altitude Clear Air Turbulence

For low altitude turbulence, i.e., for altitudes below 1,000 feet, scale length and root-mean-square intensities are functions of altitude and probability of exceedance [108]. The axial and lateral mean-square intensities are related to the normal one by

$$\sigma_u = \sigma_v = \frac{1}{(0.177 + 0.000823 \cdot h)^{0.4}} \sigma_w, \quad (3.79)$$

where h is altitude above ground, and σ_w is given by

$$\sigma_w = 0.1u_{20}, \quad (3.80)$$

with u_{20} being the mean wind velocity as measured 20 feet above ground, a function of the probability of exceedance. Light, moderate, and severe turbulence correspond to a mean wind velocity at 20 feet of 15, 30, and 45 knots, respectively [108]. The scale lengths are given by

$$2L_w = h, \quad (3.81)$$

for the normal component, and

$$L_u = 2L_v = \frac{h}{(0.177 + 0.000823 \cdot h)^{1.2}} \quad (3.82)$$

for the axial and lateral ones, with h again denoting the altitude above ground.

In the transition region between the medium and low altitude turbulence characteristics, i.e., between 1,000 and 2,000 feet, the turbulence velocities are linearly interpolated over the altitude, between the values of the low altitude model at 1,000 feet, to the value of the medium to high altitude model at 2,000 ft.

3.3.3 Discrete Gusts

According to MIL-F-8785C [106], discrete gusts may be modeled by a "1-cosine" profile,

$$v = \begin{cases} 0, & x < 0 \\ \frac{V_m}{2} \left(1 - \cos\left(\frac{\pi x}{d_m}\right) \right), & 0 \leq x \leq d_m \\ V_m, & x > d_m \end{cases} \quad (3.83)$$

where V_m is the amplitude of the gust, and d_m is one half of the total gust length. The gust may be modeled as a half single-build-up profile according to Eq. (3.83), or as a full build-up-build-down profile according to

$$v = \begin{cases} \frac{V_m}{2} \left(1 - \cos\left(\frac{\pi x}{d_m}\right) \right), & 0 \leq x \leq 2d_m \\ 0, & \text{otherwise.} \end{cases} \quad (3.84)$$

The build-up-build-down gust profile according to Eq. (3.84) is visualized in Figure 3.4. As per [103], the gust length d_m may be chosen to provide maximum system excitation by tuning the gust to the least damped longitudinal and lateral system natural frequencies. The gust amplitude V_m is a function of turbulence severity level and aircraft altitude [106].

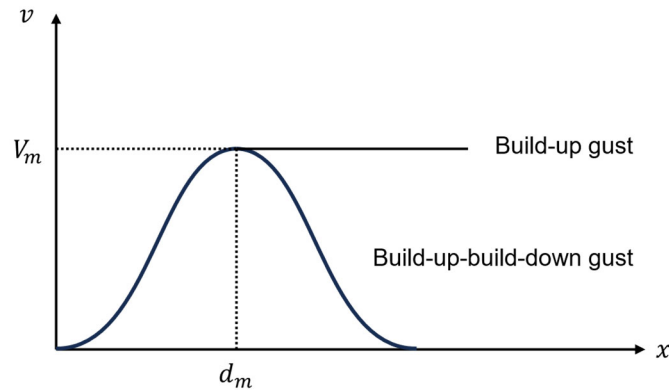


Figure 3.4: "1-cosine" discrete gust.

3.4 Subsystems

The design and analysis of the flight path controller underlie not only the dynamics of the aircraft and its environment, as presented in the previous sections, but also the dynamics of the flight control system itself, i.e., its sensing, actuating, and computing subsystems, as well as mechanical components.

From the perspective of the flight path controller, the closed inner loop dynamics, including the dynamics of the control surface actuators, as well as the closed thrust control loop including the propulsion dynamics, can be regarded as the path control actuation dynamics, i.e., the transfer functions from the commanded specific forces to the actual ones.

3.4.1 Closed Inner Loop and Actuation Systems

The main means to achieve a desired flight path curvature in the vertical or lateral plane is by changing the magnitude and the direction of the lift force vector. The build-up of the lift force, and thus the specific force in the vertical plane $f_{z,K}$, is dominated by the angle of attack (i.e. short period) dynamics, whereas the direction of the lift force, yielding a specific force in the lateral plane $f_{y,K}$, is dominated by the bank angle dynamics.

The simplest approach to model the closed loop and actuation dynamics, suitable for, e.g., control law rapid prototyping, is standard second-order transfer functions, with the closed loop short period and kinematic bank angle dynamics natural frequency and damping as characteristic parameters, respectively,

$$f_{z,K} = \frac{\omega_{0,SP}^2}{s^2 + 2 \cdot \zeta_{SP} \cdot \omega_{0,SP} + \omega_{0,SP}^2} \cdot f_{z,K,C} \quad (3.85)$$

$$f_{y,K} = \frac{\omega_{0,\mu}^2}{s^2 + 2 \cdot \zeta_{\mu} \cdot \omega_{0,\mu} + \omega_{0,\mu}^2} \cdot f_{y,K,C} \quad (3.86)$$

To capture the non-minimum phase nature of the normal specific force response, an additional zero may be added in right half plane,

$$f_{z,K} = \frac{-(T_z \cdot s - 1) \cdot \omega_{0,SP}^2}{s^2 + 2 \cdot \zeta_{SP} \cdot \omega_{0,SP} + \omega_{0,SP}^2} \cdot f_{z,K,C} \quad (3.87)$$

Taking the dynamics of the actuation system into account, it too may be modeled as second-order transfer functions, with corresponding relative damping and natural frequency.

For early controller validation using model-in-the-loop simulation, nonlinear characteristics such as position, velocity, and angular acceleration limits in the actuation system may be included in an implementation of the second-order dynamics model. For controller verification, a high-fidelity model with the full nonlinear equations of motion of the actuation system, including effects like friction, hysteresis, and backlash, according to [88] is used.

3.4.2 Closed Thrust Loop and Propulsion Systems

The propulsion system is, together with active drag control, the main means of actuation for the tangential acceleration along the flight path, subject to the path dynamic couplings discussed in Section 3.2.4, as well as managing the total energy state of the aircraft. From the perspective of the flight path controller, the dynamics of the closed thrust control loop including the propulsion dynamics constitutes the actuation dynamics along the flight path.

The magnitude and dynamics of the thrust force are greatly dependent on the type of propulsion, e.g. jet engine, turbofan, turboprop, or electric propulsion. For the development and validation of the path control concept, different desired or assumed closed loop dynamics of the relevant order, with the dominating time constants and limitations, may be employed. For any specific AFCS application that includes the speed by thrust function, a realistic, higher-fidelity model of the propulsion system, as well as a detailed design and analysis of an aircraft-specific thrust control loop, is necessary. The implemented thrust loop controller for the path control concept development is presented in Section 5.2.11.

A simple but realistic model for the engine dynamics, and the relation between the thrust lever command and the thrust force, is a first-order transfer function according to

$$T = T_{max} \cdot \delta_T = T_{max} \cdot \frac{1}{T_{\delta_T} \cdot s + 1} \cdot \delta_{T,C}, \quad (3.88)$$

with $\delta_T \in [0,1]$ being the normalized thrust lever and its commanded value, respectively, and T_{max} being the maximum available thrust, typically a function of aerodynamic velocity and density altitude,

$$T_{max} = f(V_A, h). \quad (3.89)$$

To model engine spool-up and -down characteristics, the throttle rate may be limited according to

$$\dot{\delta}_{T,min} \leq \dot{\delta}_T \leq \dot{\delta}_{T,max}. \quad (3.90)$$

3.4.3 Sensor Systems

The automatic flight path controller utilizes sensors that measure the rigid body states of the aircraft, as well as other subsystem and aircraft configuration states. Table 3.1 gives an overview of the main sensor sources and measurement inputs to the path control loops.

Table 3.1: Sensor sources and measurement inputs to the path control.

Sensor	Measurement	Description
Air Data System (ADS)	V_{IAS}	Indicated airspeed, airspeed as measured from the dynamic pressure
	V_{CAS}	Calibrated airspeed, airspeed corrected for static source errors
	h	Pressure altitude relative standard atmospheric pressure 1013.25 hPa
	h_{baro}	Baro corrected altitude for given altimeter setting (QNH)
	p_S	Static pressure
	\bar{q}	Dynamic pressure
Global Navigation Satellite System-aided Inertial Navigation System (GNSS/INS)	$\begin{bmatrix} u_K \\ v_K \\ w_K \end{bmatrix}_O^E$	Flight path
Attitude Heading Reference System (AHRS)	$\begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix}$	Attitude as Euler angles
Inertial Measurement Unit (IMU)	$\begin{bmatrix} p \\ q \\ r \end{bmatrix}_B$	Body axis rates
	$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_B$	Body axis specific forces

Other sensor signals, such as gear state indications and flap position are utilized for gain and parameter scheduling and adjusting command and envelope limits.

Additional sensor signals, such as radar altitude, actuator positions, control surface positions, and engine loads, are not directly used by the path loop controller but indicated to the pilot on the mode control and monitoring interfaces.

Sensor measurements are affected by the aircraft-specific sensor installation position and angle, delays due to internal signal processing and data fusion, and transmission delays. Measurements are also affected by sensor noise characteristics, measurement bias, and scaling errors. IEEE provides standards with sensor model equations for relevant sensor technologies. The indicated rate $\tilde{\omega}_{ib}$ from a laser gyro may for example be modeled as

$$\tilde{\omega}_{ib} = \frac{I + E + D}{1 + 10^{-6}\epsilon_k}, \quad (3.91)$$

where I denotes the ideal sensor input, E denotes the *environmentally sensitive terms*, collecting bias-like errors from environmental influences such as temperature, D denotes the *drift terms* due to for example sensor noise, and ϵ_k denotes the *composite scale factor errors* [109]. The data protocol for a given sensor gives the available measurement update rates and resolutions.

Depending on sensor installation position, denoted S , the lever arm relative to the aircraft center of gravity \vec{r}^{GS} affects the measured specific forces $(\vec{f}^S)_B$ in the body-axis frame according to

$$(\vec{f}^S)_B = \frac{\sum \vec{F}_T^G}{m} - (\vec{g}^G)_B + (\dot{\vec{\omega}}^{IB})_B \times (\vec{r}^{GS})_B + (\vec{\omega}^{IB})_B \times [(\vec{\omega}^{IB})_B \times (\vec{r}^{GS})_B], \quad (3.92)$$

with $(\vec{g}^G)_B$ being the gravitational acceleration in the body-fixed frame.

3.5 Energy Constraints and Climb and Acceleration Tradeoffs

The total energy E_{tot} of an object, neglecting any rotational energy, is given by the sum of its kinetic and potential energy,

$$E_{tot} = E_{kin} + E_{pot} = \frac{1}{2}mV_K^2 + mgh, \quad (3.93)$$

determined by its height h and speed V_K relative to some arbitrary reference (for an aircraft, e.g., the Earth's surface). For any given total energy, altitude (potential energy) and speed (kinetic energy) can be traded according to Eq. (3.93). The achievable altitude if all kinetic energy is converted to potential energy is given by the weight-specific total energy, called the *energy height*,

$$h_E = \frac{E_{tot}}{mg} = h + \frac{V_K^2}{2g}. \quad (3.94)$$

The energy state of the aircraft is changed through the application of power, typically through thrust (which tends to increase energy state) and drag (which tends to decrease energy) control. The rate of change of the energy height, Eq. (3.94), is called the *energy flow rate* or *specific excess power* (SEP), given by

$$\dot{h}_E = \frac{V_K \dot{V}_K}{g} + \dot{h} = V_K \left(\frac{\dot{V}_K}{g} + \sin \gamma_K \right). \quad (3.95)$$

Eq. (3.95) shows that increased energy (thrust greater than drag) may be used to increase the altitude or the speed, i.e. converted to a kinematic climb angle γ_K or kinematic acceleration \dot{V}_K . Analogous, reduced energy (drag exceeds thrust) must result in either deceleration or a negative flight path angle. Aircraft maneuvering can thus be viewed in terms of change of total energy and an exchange between potential and kinetic energy. The achievable combination of acceleration and flight path angle is determined by the achievable specific excess power

from the aircraft propulsion system, or active drag control. As both the flight path angle γ_K and acceleration \dot{V}_K can be measured or estimated, the specific, speed-normalized energy flow rate is known. Thus, for a given or achievable energy flow rate, a desired trade-off between flight path angle and acceleration can be calculated.

The kinetic energy and specific excess power according to Eq. (3.93)-(3.95) are given by the kinematic velocity V_K and kinematic acceleration \dot{V}_K of the aircraft. However, the aerodynamic forces and moments acting upon the aircraft are depending on its motion relative to the surrounding air, and thus on the aerodynamic velocity V_A and acceleration \dot{V}_A . Hence, the total energy of the aircraft, its distribution, and rate of change are rather to be analyzed based on the aerodynamic velocity, using the so-called aero-kinetic energy [3],

$$E_{A,tot} = E_{A,kin} + E_{pot} = \frac{1}{2} m V_A^2 + mgh. \quad (3.96)$$

Assuming a simple static wind field, without any position-dependent gradient velocities, according to Eq. (3.71), the aerodynamic acceleration \dot{V}_A is equivalent to the \dot{V}_K , Eq. (3.72), and the following energy constraints are derived using the kinematic acceleration \dot{V}_K .

The energy rate equivalent acceleration \dot{V}_E represents the case where all specific excess power is converted to acceleration, i.e., the achievable horizontal acceleration at zero flight path angle,

$$\dot{V}_E = \dot{V}_K + g \sin \gamma_K. \quad (3.97)$$

The energy rate equivalent climb angle (or simply the energy angle) γ_E represents the case where all specific excess power is converted to a flight path angle, i.e., the achievable flight path angle at zero horizontal acceleration,

$$\gamma_E = \sin^{-1} \left(\sin \gamma_K + \frac{\dot{V}_K}{g} \right). \quad (3.98)$$

The SEP can now be described in terms of energy rate equivalent acceleration or flight path angle, according to

$$\dot{h}_E = V_K \frac{\dot{V}_E}{g} = V_K \sin \gamma_E. \quad (3.99)$$

The energy angle is often displayed on primary flight displays as a cue for acceleration along the flight path. For small angles, it is simply the sum of the flight path angle and the normalized acceleration.

From Eq. (3.97) and (3.98) we may now calculate trade-offs between desired and achievable flight path angles and accelerations, for a given energy flow rate. The achievable flight path angle $\gamma_{K,\dot{V}_{K,des}}$ for a given desired acceleration $\dot{V}_{K,des}$ is given by

$$\gamma_{K,\dot{V}_{K,des}} = \sin^{-1} \left(\frac{\dot{V}_E - \dot{V}_{K,des}}{g} \right) = \sin^{-1} \left(\frac{\dot{V}_K - \dot{V}_{K,des}}{g} + \sin \gamma_K \right). \quad (3.100)$$

The achievable acceleration $\dot{V}_{K,\gamma_{des}}$ for a desired flight path angle $\gamma_{K,des}$ is similarly given by

$$\dot{V}_{K,\gamma_{des}} = g(\sin \gamma_E - \sin \gamma_K) = \dot{V}_K + g(\sin \gamma_{K,des} - \sin \gamma_K). \quad (3.101)$$

Eq. (3.100) and (3.101) consider only the current energy flow. Considering the achievable energy flow, given by the potential additional thrust from the aircraft propulsion system, or for example active drag control, the potential acceleration or flight path angle can be extended by an estimation of the additional available energy flow rate, $\Delta \dot{h}_{E,max/min}$, given by

$$\Delta \dot{h}_{E,max/min} = V_K \frac{\Delta f_{x,K,max/min}}{g}, \quad (3.102)$$

where $\Delta f_{x,K,max/min}$ is the maximum or minimum available specific force along the x_K -axis,

$$\Delta f_{x,K,max/min} = f_{x,K,max/min} - f_{x,K}. \quad (3.103)$$

The potential acceleration or flight path angle according to Eq. (3.100) and (3.101) may now be corrected by an estimation of the potential energy flow rate (for example automatically provided by an automatic thrust or drag control), with the achievable flight path angle and acceleration given by

$$\gamma_{K,\dot{V}_{K,des}} = \sin^{-1} \left(\frac{\dot{V}_K - \dot{V}_{K,des}}{g} + \sin \gamma_K + \frac{\Delta f_{x,K,max/min}}{g} \right), \quad (3.104)$$

$$\dot{V}_{K,\gamma_{des}} = \dot{V}_K + g(\sin \gamma_K - \sin \gamma_{K,des}) + \Delta f_{x,K,max/min}. \quad (3.105)$$

3.6 Force Constraints and Path Curvature Tradeoffs

A desired curvature of the vertical or lateral flight path, i.e. $\dot{\gamma}_{K,des}$ or $\dot{\chi}_{K,des}$, is achieved by changing the magnitude of the specific force perpendicular to the flight path in the vertical or lateral plane according to

$$\dot{\chi}_{K,des} = \frac{1}{V_K \cos \gamma_K} \cdot f_{y,K} \quad (3.106)$$

$$\dot{\gamma}_{K,des} = \frac{1}{V_K} (-f_{z,K} - g \cos \gamma_K) \quad (3.107)$$

For most conventional aircraft configurations, the primary force acting perpendicular to the flight path is the (positive) lift force, since the main wings are the principal parts of the aircraft designed with efficient aerodynamic force generation in mind. (Other means of producing flight path perpendicular forces include thrust vectoring and direct side force control.) However, just as in the vertical plane, where the limited available energy rate precludes an arbitrary combination of acceleration and flight path angle from being achieved, the limited available transverse force precludes an arbitrary curvature of the vertical and lateral flight path. Thus, for high-bandwidth concurrent maneuvering in the vertical and lateral plane, the permissible transverse force has to be taken into account and, in case the transverse force is saturated, a

prioritization or trade-off between desired curvature in the vertical and lateral plane becomes necessary.

The specific force along the z -axes in the kinematic K and the intermediate kinematic \bar{K} frames is noted with the opposite sign as the more intuitive load factors $n_{z,K}$ and $n_{z,\bar{K}}$, whereas the load factors and specific forces along the kinematic and intermediate kinematic y -axes are of the same sign,

$$n_{z,K} = -\frac{f_{z,K}}{g} \quad (3.108)$$

$$n_{y,K} = \frac{f_{y,K}}{g} \quad (3.109)$$

$$n_{z,\bar{K}} = \sqrt{(n_{y,K})^2 + (n_{z,K})^2} = -\sqrt{\left(\frac{f_{y,K}}{g}\right)^2 + \left(\frac{f_{z,K}}{g}\right)^2}, \quad (3.110)$$

$$\mu_K = \tan^{-1}\left(\frac{n_{y,K}}{n_{z,K}}\right). \quad (3.111)$$

Here, μ_K is the kinematic bank angle, i.e. the rotation angle of the intermediate kinematic frame \bar{K} in relation to the kinematic frame K , see Appendix B. Thus, for the subsequent force constraint analysis, load factors are used instead of the corresponding specific forces.

The flight path transverse load factors in the kinematic frame for desired curvatures of the vertical and lateral flight path, $\dot{\gamma}_{K,des}$ and $\dot{\chi}_{K,des}$, are given by

$$n_{z,K} = \frac{V_K}{g} \dot{\gamma}_{K,des} + \cos \gamma_K, \quad (3.112)$$

$$n_{y,K} = \frac{V_K}{g} \cos \gamma_K \dot{\chi}_{K,des}. \quad (3.113)$$

The total transverse load factor $n_{z,\bar{K}}$ for achieving the desired vertical and lateral path curvatures, $\dot{\gamma}_{K,des}$ and $\dot{\chi}_{K,des}$ respectively, must be equal to or less than the maximum transverse load factor $n_{z,\bar{K},max}$,

$$n_{z,\bar{K}} = \sqrt{\left(\frac{V_K}{g} \cos \gamma_K \dot{\chi}_{K,des}\right)^2 + \left(\frac{V_K}{g} \dot{\gamma}_{K,des} + \cos \gamma_K\right)^2} \leq n_{z,\bar{K},max}. \quad (3.114)$$

From Eq. (3.114), constraints on $\dot{\gamma}_{K,des}$ and $\dot{\chi}_{K,des}$ may be derived, in order to achieve the desired curvature in a prioritized plane. For example, if maneuvering in the vertical plane is prioritized, the constraints on the lateral path curvature are given by

$$\dot{\chi}_{K,max} = \frac{g}{V_K \cos \gamma_K} \sqrt{n_{z,\bar{K},max}^2 - \left(\cos \gamma_K + \frac{V_K}{g} \dot{\gamma}_{K,des}\right)^2}, \quad (3.115)$$

$$\dot{\chi}_{K,min} = -\frac{g}{V_K \cos \gamma_K} \sqrt{n_{z,\bar{K},max}^2 - \left(\cos \gamma_K + \frac{V_K}{g} \dot{\gamma}_{K,des} \right)^2} \quad (3.116)$$

In order for the expression under the square root in Eq. (3.115) and (3.116) to be positive, the following constraints on the vertical path curvature apply,

$$\dot{\gamma}_{K,max} = \frac{g}{V_K} (n_{z,\bar{K},max} - \cos \gamma_K), \quad (3.117)$$

$$\dot{\gamma}_{K,min} = \frac{g}{V_K} (-n_{z,\bar{K},max} - \cos \gamma_K). \quad (3.118)$$

Eq. (3.118) represents the case when the aircraft is inverted and pulling with maximum load factor to achieve a negative vertical flight path curvature.

Analogously, maneuvering in the lateral plane may be prioritized, either by allowing the full transverse load factor to produce a lateral path curvature (i.e. with bank angle equal to 90 degrees), or by allowing the residual load factor when maintaining the current flight path angle for maneuvering in the lateral plane, thereby inhibiting further curvature of the vertical flight path.

The constraints on the vertical path curvature are given by

$$\dot{\gamma}_{K,max} = \frac{g}{V_K} \sqrt{n_{z,\bar{K},max}^2 - \left(\frac{V_K}{g} \cos \gamma_K \dot{\chi}_{K,des} \right)^2} - \frac{V_K}{g} \cos \gamma_K, \quad (3.119)$$

$$\dot{\gamma}_{K,min} = -\frac{g}{V_K} \sqrt{n_{z,\bar{K},max}^2 - \left(\frac{V_K}{g} \cos \gamma_K \dot{\chi}_{K,des} \right)^2} - \frac{V_K}{g} \cos \gamma_K, \quad (3.120)$$

and the constraints on the lateral curvature, for full transverse load factor,

$$\dot{\chi}_{K,max} = \frac{g}{V_K \cos \gamma_K} n_{z,\bar{K},max}, \quad (3.121)$$

$$\dot{\chi}_{K,min} = -\frac{g}{V_K \cos \gamma_K} n_{z,\bar{K},max}. \quad (3.122)$$

If required to maintain the current climb angle, the residual maneuvering load factor in the lateral plane is given by

$$\dot{\chi}_{K,max} = \frac{g}{V_K} \sqrt{\left(\frac{n_{z,\bar{K},max}}{\cos \gamma_K} \right)^2 - 1}, \quad (3.123)$$

$$\dot{\chi}_{K,min} = -\frac{g}{V_K} \sqrt{\left(\frac{n_{z,\bar{K},max}}{\cos \gamma_K} \right)^2 - 1}. \quad (3.124)$$

4 Control Theory Preliminaries

This chapter introduces the mathematical and control theoretical concepts necessary for the presentation of the control strategy in Chapter 5. The objective is a brief introduction to required concepts and notations so that the focus of the subsequent chapters may be on the application-specific aspects.

The basic control principle for the path control loops, i.e., airspeed, flight path angle, and track angle, is a reference model-based dynamic inversion of the translational equations of motion. The general nonlinear dynamic inversion is described in Section 4.1. With pseudo-control command variables determined, reference models of sufficient degree to produce consistent and smooth trajectories for the reference states and their pseudo-control may be designed for desired input/output dynamics. A reference model-based dynamic inversion structure is presented in Section 4.2. In Section 4.3, the dynamics of errors due to modeling uncertainties and neglected dynamics is investigated. The closed inner loop and actuation dynamics is purposely excluded from the inversion of the flight path dynamics. Introducing pseudo-control hedging into the controller structure, presented in Section 4.4, allows for the consideration of the closed inner loop and actuation dynamics and limitations, by feeding back control reaction deficits to the reference model and adjusting the reference dynamics and control error. This modification of the feedback control structure, making the reference model not only a feedforward element, requires additional stability analyses taking all feedback loops into account.

4.1 Nonlinear Dynamic Inversion

The objective of the nonlinear dynamic inversion, also known as *input/output linearization* or *feedback linearization*, is to find a nonlinear state transformation

$$\mathbf{z} = \Phi(\mathbf{x}) \quad (4.1)$$

of a nonlinear plant so that the transformed system has linear input-output dynamics. The transformed plant may then be analyzed using traditional linear system and control theory and a linear controller designed and assessed using typical control approaches and classical stability criteria available. Another advantage is that the nonlinear inversion further decouples the nonlinear Multiple-Input/Multiple-Output (MIMO) system into several SISO systems.

Transformation of the Dynamic System

Consider an input-affine MIMO system on the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{h}(\mathbf{x})\end{aligned}\quad (4.2)$$

where $\mathbf{x} = [x_1, \dots, x_n]^T \in \Omega_{\mathbf{x}} \subseteq \mathbb{R}^n$ is the state vector, $\mathbf{u} = [u_1, \dots, u_m]^T \in \Omega_{\mathbf{u}} \subseteq \mathbb{R}^m$ is the input vector, $\mathbf{y} = [y_1, \dots, y_m]^T \in \Omega_{\mathbf{y}} \subseteq \mathbb{R}^m$ is the system output vector, and $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T: \Omega_{\mathbf{x}} \rightarrow \mathbb{R}^n$ and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_m(\mathbf{x})]^T: \Omega_{\mathbf{x}} \rightarrow \mathbb{R}^m$ are nonlinear sufficiently smooth mappings. The basic approach of the input-output linearization is to repeatedly differentiate each output y_i until one of the inputs in \mathbf{u} appears, and then design a decoupling control law for \mathbf{u} to cancel the nonlinearities.

Differentiating output y_i of equation (4.2) yields

$$\begin{aligned}\dot{y}_i &= \frac{\partial y_i}{\partial t} = \left(\frac{\partial h_i}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \left(\frac{\partial h_i}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x}) + \left(\frac{\partial h_i}{\partial \mathbf{x}}\right)^T \mathbf{G}(\mathbf{x})\mathbf{u} \\ &= L_{\mathbf{f}}h_i(\mathbf{x}) + L_{\mathbf{G}}h_i(\mathbf{x})\mathbf{u}\end{aligned}\quad (4.3)$$

$$\begin{aligned}\ddot{y}_i &= \frac{\partial \dot{y}_i}{\partial t} = \left(\frac{\partial L_{\mathbf{f}}h_i}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \left(\frac{\partial L_{\mathbf{f}}h_i}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x}) + \left(\frac{\partial L_{\mathbf{f}}h_i}{\partial \mathbf{x}}\right)^T \mathbf{G}(\mathbf{x})\mathbf{u} \\ &= L_{\mathbf{f}}^2h_i(\mathbf{x}) + L_{\mathbf{G}}L_{\mathbf{f}}h_i(\mathbf{x})\mathbf{u}\end{aligned}\quad (4.4)$$

⋮

$$\begin{aligned}y_i^{(r_i)} &= \frac{\partial y_i^{(r_i-1)}}{\partial t} = \left(\frac{\partial L_{\mathbf{f}}^{(r_i-1)}h_i}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \left(\frac{\partial L_{\mathbf{f}}^{(r_i-1)}h_i}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x}) + \left(\frac{\partial L_{\mathbf{f}}^{(r_i-1)}h_i}{\partial \mathbf{x}}\right)^T \mathbf{G}(\mathbf{x})\mathbf{u} \\ &= L_{\mathbf{f}}^{(r_i)}h_i(\mathbf{x}) + L_{\mathbf{G}}L_{\mathbf{f}}^{(r_i-1)}h_i(\mathbf{x})\mathbf{u}\end{aligned}\quad (4.5)$$

until for some integer r_i , $L_{\mathbf{G}}L_{\mathbf{f}}^{(r_i-1)}h_i(\mathbf{x}) \neq \mathbf{0}$ for some $\mathbf{x} = \mathbf{x}_0$ in $\Omega_{\mathbf{x}}$. Here

$$L_{\mathbf{f}}h_i(\mathbf{x}) \triangleq \left(\frac{\partial h_i}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x})\quad (4.6)$$

is the *Lie derivative*, the directional derivative of the scalar function $h_i(\mathbf{x})$ with respect to $\mathbf{f}(\mathbf{x})$, for $i = 1, \dots, m$. The number r_i of required differentiations of the output y_i is referred to as the *relative degree* of the output y_i . For the MIMO system (4.2), the total or vectorial relative degree r is defined as

$$r = r_1 + \dots + r_m = \sum_{i=1}^m r_i \leq n.\quad (4.7)$$

Identifying the Eq. (4.5) for each input y_i , the input/output dynamics of the system can now be written on the form

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u} \quad (4.8)$$

with

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{f}}^{(r_1)} h_1(\mathbf{x}) \\ \vdots \\ L_{\mathbf{f}}^{(r_m)} h_m(\mathbf{x}) \end{bmatrix} \quad (4.9)$$

and

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{g}_1} L_{\mathbf{f}}^{(r_1-1)} h_1(\mathbf{x}) & \cdots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{(r_1-1)} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ L_{\mathbf{g}_1} L_{\mathbf{f}}^{(r_m-1)} h_m(\mathbf{x}) & \cdots & L_{\mathbf{g}_m} L_{\mathbf{f}}^{(r_m-1)} h_m(\mathbf{x}) \end{bmatrix}. \quad (4.10)$$

The matrix $\mathbf{A}(\mathbf{x})$ is called the *decoupling matrix* of the system, since it represents the relation between the inputs u_1, \dots, u_m and the first directly controllable derivative of each output y_1, \dots, y_m , i.e. $y_1^{(r_1)}, \dots, y_m^{(r_m)}$, and will be used further on to design the linearizing feedback law. The decoupling matrix $\mathbf{A}(\mathbf{x})$ must be nonsingular, i.e., invertible in the region \mathbf{x} , in order for a linearizing feedback to exist [11].

The first r equations of the transformation $\mathbf{z} = \Phi(\mathbf{x})$ are given by the outputs y_i and their derivatives $\dot{y}_i, \dots, y_i^{(r_i-1)}$, for $1 \leq i \leq m$, according to

$$\begin{aligned} z_1 &= \xi_1^1 = \Phi_1(\mathbf{x}) = y_1 = h_1(\mathbf{x}) \\ z_2 &= \xi_2^1 = \Phi_2(\mathbf{x}) = \frac{\partial \Phi_1(\mathbf{x})}{\partial t} = \dot{y}_1 = L_{\mathbf{f}} h_1(\mathbf{x}) \\ &\vdots \\ z_{r_1} &= \xi_{r_1}^1 = \Phi_{r_1}(\mathbf{x}) = \frac{\partial \Phi_{r_1-1}(\mathbf{x})}{\partial t} = y_1^{(r_1-1)} = L_{\mathbf{f}}^{(r_1-1)} h_1(\mathbf{x}) \\ z_{r_1+1} &= \xi_1^2 = \Phi_{r_1+1}(\mathbf{x}) = y_2 = h_2(\mathbf{x}) \\ &\vdots \\ z_{r_1+r_2} &= \xi_{r_2}^2 = \Phi_{r_1+r_2}(\mathbf{x}) = \frac{\partial \Phi_{r_1+r_2-1}(\mathbf{x})}{\partial t} = y_2^{(r_2-1)} = L_{\mathbf{f}}^{(r_2-1)} h_2(\mathbf{x}) \\ &\vdots \\ z_r &= \xi_{r_m}^m = \Phi_r(\mathbf{x}) = \frac{\partial \Phi_{r_1+\dots+r_m-1}(\mathbf{x})}{\partial t} = y_m^{(r_m-1)} = L_{\mathbf{f}}^{(r_m-1)} h_m(\mathbf{x}). \end{aligned} \quad (4.11)$$

The transformation $\mathbf{z} = \Phi(\mathbf{x})$ must (at least locally) be diffeomorphic, i.e., the Jacobi matrix $\partial \Phi / \partial \mathbf{x}$ must be non-singular, in order for the inverse transformation $\mathbf{x} = \Phi^{-1}(\mathbf{z})$ to exist. This is the case if all row vectors of the Jacobi matrix $\partial \Phi / \partial \mathbf{x}$ are linearly independent. As for the first r rows this is true, as they are constituted by the differentials $\partial h_i(\mathbf{x}), L_{\mathbf{f}} h_i(\mathbf{x}), \dots, L_{\mathbf{f}}^{(r_i-1)} h_i(\mathbf{x})$ for $1 \leq i \leq m$ (their linear independence is proven in e.g. [11]). From this follows that the relative degree of a system of order n is always smaller than or equal to n , because there cannot be more than n linearly independent vectors in an n -dimensional space.

If the total relative degree is equal to the degree of the system, i.e. $r = n$, the transformation is given by Eq. (4.11) and the full dynamics of the system is observable and controllable. If, however, $r < n$, the transformation system exhibits some internal states, unobservable in the linear input-output dynamics, called the *internal dynamics* of the system. For the internal dynamics, an additional $n - r$ coordinates must be found,

$$\begin{aligned} z_{r+1} &= \eta_1 = \Phi_{r+1}(\mathbf{x}) \\ z_n &= \eta_{n-r} = \Phi_n(\mathbf{x}) \end{aligned} \quad (4.12)$$

with linear independent differentials $\partial\Phi_i/\partial\mathbf{x}$ for $r + 1 \leq i \leq n$, linear independent also with respect those of the already defined r coordinates. The boundedness of the internal dynamics is essential for the input-output linearization to be successful, as unbounded internal states in any physical system would either lead to some form of destruction of the system at hand, or, if internal states are saturated, inhibit the input-output linearization. Since the internal dynamics is described by a set of nonlinear differential equations, the analysis of its stability properties is non-trivial, and usually includes complex methods, for example using Lyapunov functions [11]. A detailed investigation of the internal dynamics for such systems is superfluous for the purpose of the flight path controller, as the path dynamics will be shown to exhibit no such internal dynamics and is thus omitted here.

Linearizing Feedback

Consider a nonlinear state feedback law according to

$$\mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x})\mathbf{v} \quad (4.13)$$

with a new *pseudo-control vector* $\mathbf{v} = [v_1 \dots v_m]^T$, and inserting it into Eq. (4.8),

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x}) \cdot \left[\boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}) \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \right]. \quad (4.14)$$

Selecting $\boldsymbol{\alpha}(\mathbf{x})$ and $\boldsymbol{\beta}(\mathbf{x})$ according to

$$\begin{aligned} \boldsymbol{\alpha}(\mathbf{x}) &= -\mathbf{A}^{-1}(\mathbf{x})\mathbf{b}(\mathbf{x}) \\ \boldsymbol{\beta}(\mathbf{x}) &= \mathbf{A}^{-1}(\mathbf{x}) \end{aligned} \quad (4.15)$$

and inserting into Eq. (4.13) yields the control law

$$\mathbf{u} = \mathbf{A}^{-1}[\mathbf{v} - \mathbf{b}(\mathbf{x})], \quad (4.16)$$

which when inserted into the input-output dynamics, Eq. (4.8) yields

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x}) \cdot \mathbf{A}^{-1} \left[-\mathbf{b}(\mathbf{x}) + \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \right], \quad (4.17)$$

i.e., m scalar equations on the form $y_i^{(r_i)} = v_i$. This means that the previously coupled, nonlinear dynamics of the initial system is substituted by decoupled chains of integrators

between each pseudo-control input v_i and its corresponding output y_i . Since each input now only affects a single output, the linearizing feedback, Eq. (4.16), is called the *decoupling control law*.

The transformed system on normal form can now be written as

$$\dot{\xi}^i = \begin{bmatrix} \dot{\xi}_2^i \\ \vdots \\ \dot{\xi}_{r_i}^i \\ v_i \end{bmatrix}, \quad y_i = \xi_1^i \quad (4.18)$$

for each output y_i and its corresponding linearizing coordinates $\xi_j^i = L_f^{(r_j-1)} h_i(\mathbf{x})$, $1 \leq j \leq r_i$, $1 \leq i \leq m$; on state space form

$$\begin{aligned} \dot{\xi}^i &= \mathbf{A}_i \xi^i + \mathbf{b}_i v_i \\ y_i &= \mathbf{c}_i^T \xi^i, \end{aligned} \quad (4.19)$$

with

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{r_i \times r_i} \quad \mathbf{b}_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{r_i \times 1} \quad \mathbf{c}_i = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}_{r_i \times 1}. \quad (4.20)$$

The Transformed System

The original MIMO system, Eq. (4.2), with m inputs and m outputs, and coupled and nonlinear input-output dynamics, has been transformed, using a coordinate transformation and a linearizing state feedback (the "inversion"), into a system with decoupled and linear dynamics between a new pseudo-control input vector and the system outputs, where each output y_i results directly from the its corresponding pseudo-control v_i , following r_i integrations.

The relation between the original system inputs \mathbf{u} , and the derivatives of the outputs $y_i^{(r_i)}$ is defined by a set of nonlinear, algebraic equations,

$$\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u}, \quad (4.21)$$

composed by the vector fields $\mathbf{f}, \mathbf{g}_1, \dots, \mathbf{g}_m$ determining the inherent and input dynamics of the system, and their directional derivatives, dependent on the current state and input vectors \mathbf{x} and \mathbf{u} , respectively. Through the feedback linearization, the set of equations defining the relation between the inputs and derivatives of the outputs, i.e., the dynamics of the original system, is simply inverted,

$$\mathbf{u} = \hat{\mathbf{F}}(\mathbf{x}, \mathbf{v}) = \hat{\mathbf{A}}^{-1}(\mathbf{x})[\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})], \quad (4.22)$$

with the pseudo-control vector as the desired output derivatives.

For the outputs of the system to follow some desired reference trajectories, given by $\mathbf{y}_R(t)$, the corresponding r_i -th derivative of each output reference trajectory $y_{R,i}(t)$, i.e. $y_{R,i}^{(r_i)}(t)$ must be

determined, and fed to the system as the pseudo-control input v_i . This is commonly realized using *reference models* for the desired dynamics. A graphical representation of the input-output linearization with reference models is given in Figure 4.1.

For the path controller implementation, the variables associated with the reference models are denoted with index RM , such as the pseudo-control vector $\mathbf{v}_{RM} = [v_{RM,1} \dots v_{RM,m}]^T$; this notation will be used in the following.

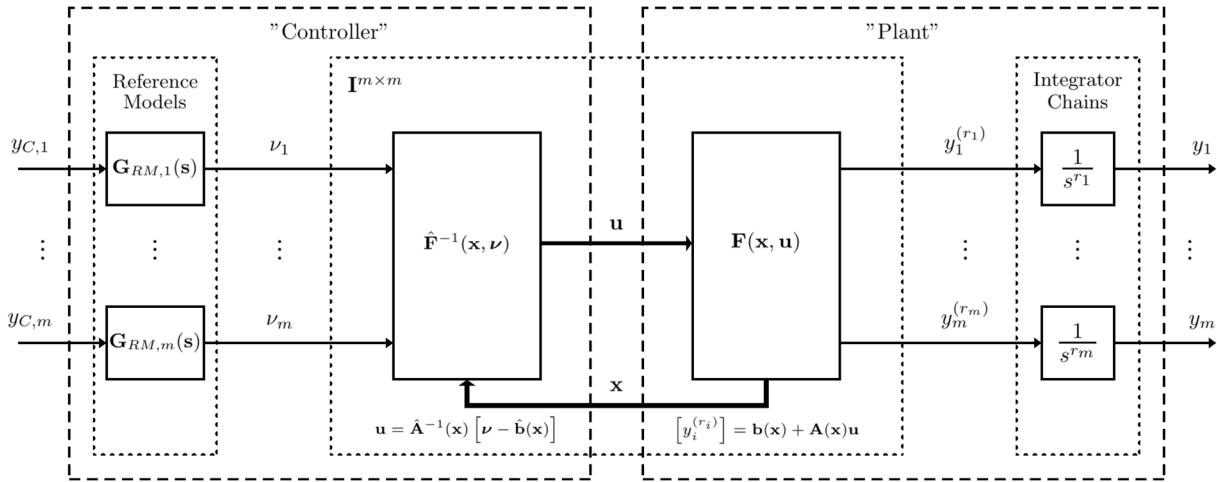


Figure 4.1: Reference model based dynamic inversion.

4.2 Reference Models for Desired Closed Loop Dynamics

The reference trajectory $y_{RM,i}(t)$, of each output and its derivatives up to the relative degree r_i , and hence pseudo control $v_{RM,i}$, are generated using a reference model of the corresponding or higher order. A reference model of a higher order than the relative degree may be utilized to produce smooth pseudo-control trajectories.

With the selection of the reference model parameters, the desired closed-loop dynamics can thereby be defined. The desired reference dynamics, however, cannot be chosen arbitrarily fast. The achievable reference dynamics underlies the bandwidth and deflection limitations of subsequent inner loops, determined by for example the inherent dynamics of the system and achievable actuation dynamics. Since a reference model is commonly used to shape a command $y_{C,i}$ that the system cannot instantaneously track, for example stepwise or discontinuous flight path commands, it is further known as a *command filter*. Often, a linear reference model is chosen, since its dynamics may be easily defined and interpreted, for example, in terms of time constants or natural frequencies and damping.

The general form for a linear reference model of order n is given by

$$y_{RM}^{(n)} + a_{n-1}y_{RM}^{(n-1)} + \dots + a_1\dot{y}_{RM} + a_0y_{RM} = a_0y_C. \quad (4.23)$$

In the case that the reference model is of the same order as the relative degree of the current output y , i.e., $n = r$, the pseudo control v_{RM} equals the highest derivative,

$$\begin{aligned}
 v_{RM} &= y_{RM}^{(n)} \\
 &= -a_{r-1}y_{RM}^{(n-1)} - \dots - a_1\dot{y}_{RM} - a_0y_{RM} + a_0y_C.
 \end{aligned}
 \tag{4.24}$$

With a more general reference model, with order higher than the relative degree, i.e., $n > r$, the extra orders of dynamics can be used to additionally shape the form of the pseudo-control v_{RM} . The pseudo-control and its states are then given by

$$\begin{aligned}
 v_{RM}^{(n-r)} &= y_{RM}^{(n)} \\
 v_{RM}^{(n-r-1)} &= y_{RM}^{(n-1)} \\
 &\vdots \\
 \dot{v}_{RM} &= y_{RM}^{(r+1)} \\
 v_{RM} &= y_{RM}^{(r)}.
 \end{aligned}
 \tag{4.25}$$

Figure 4.2 shows a linear reference model of order n , that generates a reference output $y_{RM}(t)$, as well as its derivatives, $y_{RM}^{(i)}(t)$, up to the relative degree r of the plant, and $0 \leq i \leq r \leq j \leq n$.

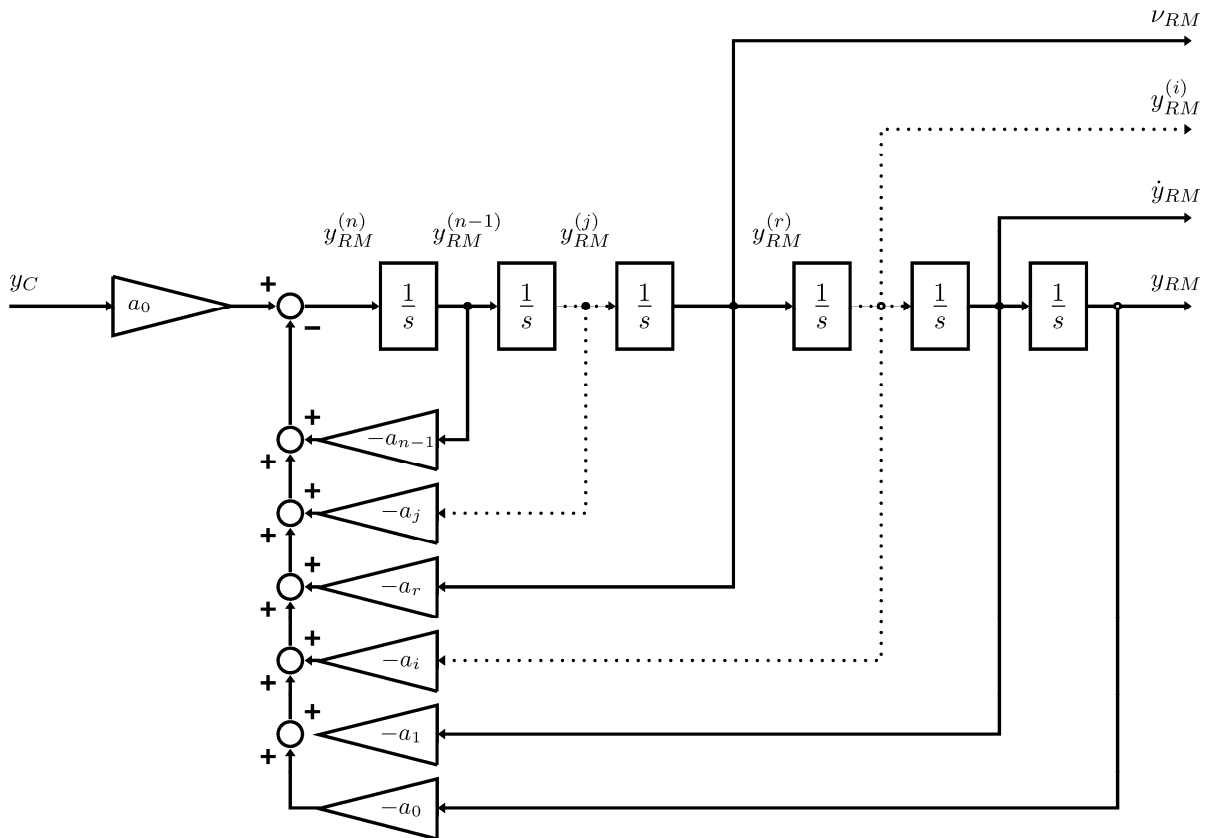


Figure 4.2: Linear reference model of order n for pseudo-controls up to order r .

In the frequency domain, the corresponding transfer function from the command y_C to the reference output y_{RM} is given by

$$G_{y_{RM}y_c}(s) = \frac{a_0}{s^r + a_{r-1}s^{r-1} + \dots + a_1s + a_0}. \quad (4.26)$$

4.3 Error Dynamics and Stabilizing Controller

If the plant dynamics used for the inversion perfectly represents the true plant dynamics, and if the initial conditions of the reference states and all their derivatives perfectly match those of the true plant, i.e., for $t = t_0$ and $1 \leq i \leq m$,

$$\begin{bmatrix} y_i(t_0) \\ \dot{y}_i(t_0) \\ \vdots \\ y_i^{(r_i-1)}(t_0) \end{bmatrix} = \begin{bmatrix} y_{RM,i}(t_0) \\ \dot{y}_{RM,i}(t_0) \\ \vdots \\ y_{RM,i}^{(r_i-1)}(t_0) \end{bmatrix}. \quad (4.27)$$

the output of the system would perfectly match the desired reference states, i.e.

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} y_{RM,1}(t) \\ y_{RM,2}(t) \\ \vdots \\ y_{RM,m}(t) \end{bmatrix}, \quad \forall t \geq t_0. \quad (4.28)$$

For practical applications, this is generally not the case. Since the true dynamics of a system, $\mathbf{F}(\mathbf{x}, \mathbf{u}) = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u}$, can be arbitrarily complex, a simplified (yet possibly still complex) model of the system is always used for the inversion,

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{v}) = \hat{\mathbf{A}}^{-1}(\mathbf{x})[\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] \quad (4.29)$$

whose dynamics differ from that of the true system for a number of reasons:

- **Model and parameter uncertainties:** The inverted equations of motions differ from those of the real system, since they may include assumptions and simplifications. Numerical values of parameters in the inversion model (e.g., aircraft mass) may differ from the actual values of the real system. From the perspective of the path controller, the translational dynamics is well known, and any inversion error is due to uncertainties in the measurement of the aircraft states.
- **Neglected dynamics:** The inversion may purposely neglect certain dynamics, for example, actuation dynamics. Such a neglect may be justified if the bandwidth of the neglected dynamics is sufficiently time scale separated from the inverted dynamics.
- **Sensor dynamics, errors, delays, and filters:** The measured system states used for the inversion are subject to the dynamics of the sensors used, including biases, delays, noise, and update rates. To reduce the impact of noise, known structural couplings, and poor update rates, the measured signals are commonly filtered before entering the inversion, for example using low-pass, band-stop, notch, or complementary filtering techniques.
- **External disturbances:** Every physical system is subject to non-measurable external disturbances, acting as additional inputs to the system dynamics that are not included in the inversion.

Error Representation

With the true plant dynamics given by Eq. (4.2), the model dynamics may be analogously given by

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \hat{\mathbf{f}}(\mathbf{x}) + \hat{\mathbf{G}}(\mathbf{x})\mathbf{u}, \\ \hat{\mathbf{y}} &= \hat{\mathbf{h}}(\mathbf{x}).\end{aligned}\quad (4.30)$$

The input-output dynamics of the model and the true plant are given by,

$$\begin{bmatrix} \hat{y}_1^{(r_1)}(\mathbf{x}, \mathbf{u}) \\ \vdots \\ \hat{y}_m^{(r_m)}(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \hat{\mathbf{b}}(\mathbf{x}) + \hat{\mathbf{A}}(\mathbf{x}) \cdot \mathbf{u}, \quad (4.31)$$

and

$$\begin{bmatrix} y_1^{(r_1)}(\mathbf{x}, \mathbf{u}) \\ \vdots \\ y_m^{(r_m)}(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x}) \cdot \mathbf{u}, \quad (4.32)$$

respectively. With $\hat{\mathbf{b}}$ and \mathbf{b} given by Eq. (4.9), and $\hat{\mathbf{A}}$ and \mathbf{A} by Eq. (4.10), the deviation $\Delta(\mathbf{x}, \mathbf{u})$ between plant and model is given by

$$\Delta(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \Delta_1(\mathbf{x}, \mathbf{u}) \\ \vdots \\ \Delta_m(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \begin{bmatrix} y_1^{(r_1)}(\mathbf{x}, \mathbf{u}) \\ \vdots \\ y_m^{(r_m)}(\mathbf{x}, \mathbf{u}) \end{bmatrix} - \begin{bmatrix} \hat{y}_1^{(r_1)}(\mathbf{x}, \mathbf{u}) \\ \vdots \\ \hat{y}_m^{(r_m)}(\mathbf{x}, \mathbf{u}) \end{bmatrix}. \quad (4.33)$$

With the linearizing feedback law, Eq. (4.16), based on the model dynamics, i.e.

$$\mathbf{u} = \hat{\mathbf{A}}^{-1}[\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})], \quad (4.34)$$

the deviation, Eq. (4.33), can be written as a function of the reference model pseudo control vector \mathbf{v} ,

$$\begin{aligned}\Delta(\mathbf{x}, \mathbf{v}) &= [\mathbf{b}(\mathbf{x}) - \hat{\mathbf{b}}(\mathbf{x})] + [\mathbf{A}(\mathbf{x}) - \hat{\mathbf{A}}(\mathbf{x})] \cdot \hat{\mathbf{A}}^{-1}(\mathbf{x})[\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] \\ &= [\mathbf{b}(\mathbf{x}) - \hat{\mathbf{b}}(\mathbf{x})] + \mathbf{A}(\mathbf{x}) \cdot \hat{\mathbf{A}}^{-1}(\mathbf{x}) \cdot [\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] - [\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] \\ &= \mathbf{b}(\mathbf{x}) + \mathbf{A}(\mathbf{x}) \cdot \hat{\mathbf{A}}^{-1}(\mathbf{x}) \cdot [\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] - \mathbf{v}.\end{aligned}\quad (4.35)$$

With the first two terms on the right-hand side being the real input-output dynamics, Eq. (4.32), with the linearizing feedback law, Eq. (4.34), Eq. (4.35) reduces to

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} v_{RM,1} \\ \vdots \\ v_{RM,m} \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_m \end{bmatrix}. \quad (4.36)$$

Hence, with an exact inversion, the output y_i would follow the desired reference $y_{RM,i}$, but due to model uncertainties, neglected dynamics and disturbances, the r_i -th derivative of the output y_i differs from the pseudo control, $y_i^{(r_i)} = v_{RM,i} + \Delta_i$, propagating through the r_i integrations, and leading to an output tracking error.

The chain of integrators of the inversion structure means that the input-output-linearized system has poles in the origin of the complex plane, and any error dynamics will therefore not be inherently stable. An additional error controller is necessary to move those poles into the left half complex plane, in order to stabilize the error dynamics and minimize the tracking error.

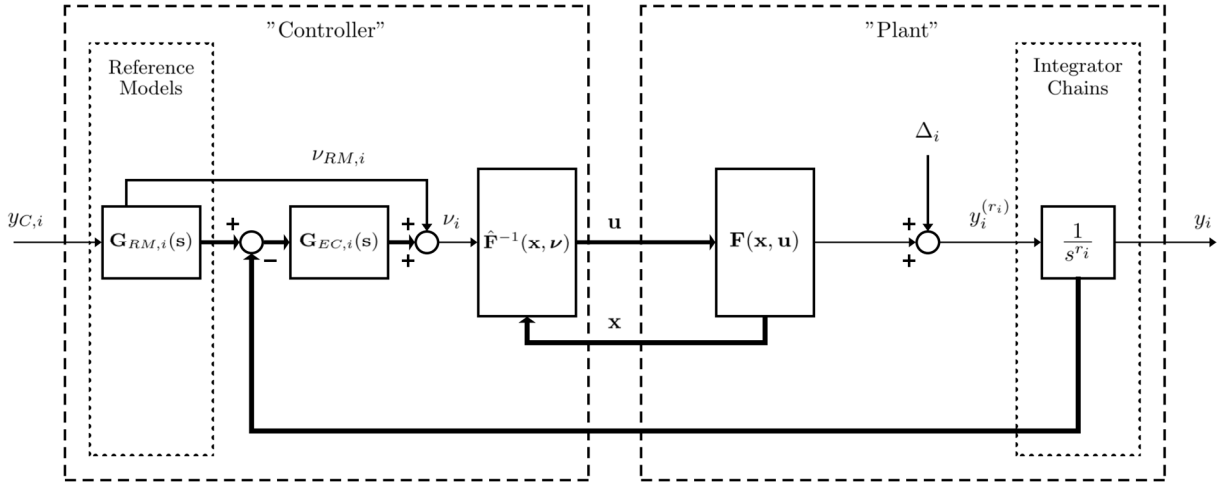


Figure 4.3: Reference model based dynamic inversion with error dynamics stabilization.

Error Stabilization

The control error vector is defined as the deviations between each output and its corresponding reference value, according to

$$\mathbf{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} y_{RM,1} - y_1 \\ \vdots \\ y_{RM,m} - y_m \end{bmatrix}, \quad (4.37)$$

and the higher order error derivatives analogously,

$$e_i^{(j)} = y_{RM,i}^{(j)} - y_i^{(j)}. \quad (4.38)$$

The input-output linearization transforms the original nonlinear system into a set of decoupled and linear SISO systems, wherefore the error dynamics in the following is analyzed for the SISO case, omitting the index i .

An error controller is introduced, amending the pseudo-control from the reference model, v_{RM} , by a pseudo-control v_{EC} , consisting of proportional feedback of the control error up to and including its derivative of order $r - 1$, giving the total pseudo-control

$$\begin{aligned}
v &= v_{RM} + v_{EC} \\
&= v_{RM} + k_{P,r-1} \cdot (y_{RM}^{(r-1)} - y^{(r-1)}) + \dots + k_{P,1} \cdot (\dot{y}_{RM} - \dot{y}) \\
&\quad + k_{P,0} \cdot (y_{RM} - y) \\
&= v_{RM} + \sum_{i=1}^r k_{P,i-1} \cdot (y_{RM}^{(i-1)} - y^{(i-1)}) \\
&= v_{RM} + \sum_{i=1}^r k_{P,i-1} \cdot e^{(i-1)},
\end{aligned} \tag{4.39}$$

and the control law

$$\begin{aligned}
u &= \frac{1}{\hat{a}(\xi)} \cdot [-\hat{b}(\xi) + v] \\
&= \frac{1}{\hat{a}(\xi)} \cdot [-\hat{b}(\xi) + v_{RM} + v_{EC}] \\
&= \frac{1}{\hat{a}(\xi)} \cdot \left[-\hat{b}(\xi) + v_{RM} + \sum_{i=1}^r k_{P,i-1} \cdot e^{(i-1)} \right].
\end{aligned} \tag{4.40}$$

Inserting the Eq. (4.39) into the input-output dynamics, Eq. (4.36), gives

$$\begin{aligned}
y^{(r)} &= v + \Delta \\
&= v_{RM} + v_{EC} + \Delta \\
&= v_{RM} + \sum_{i=1}^r k_{P,i-1} \cdot e^{(i-1)} + \Delta,
\end{aligned} \tag{4.41}$$

which together with

$$e^{(r)} = y_{RM}^{(r)} - y^{(r)} = v_{RM} - y^{(r)} \tag{4.42}$$

yields the differential equation for the control error,

$$e^{(r)} + \sum_{i=1}^r k_{P,i-1} \cdot e^{(i-1)} = -\Delta. \tag{4.43}$$

The linear controller with the coefficients $k_{P,i}$ for the stabilization of the error dynamics can be designed using any method from standard linear control theory, for example extending the error controller by an integrator for ensuring static accuracy.

The error dynamics is independent of the reference dynamics, which is given by the coefficients a_i of the reference model, see Eq. (4.23).

4.4 Consideration of Actuation Dynamics through Pseudo Control Hedging

Up till this point, the actuation dynamics has remained unconsidered in the inversion and the linearizing state feedback control law. Any deviations resulting from unconsidered actuation dynamics are assumed to be small and part of the modeling error Δ , with the real plant input \mathbf{u} following the commanded input without any significant dynamics or delay. Neglecting the actuation dynamics in this way during the inversion may be justified if the reference dynamics and dynamics of subsequent inner loops are sufficiently timescale separated. This is not the case for the path controller to be designed. As the dynamics and limitations of the closed inner loop and actuation dynamics, as well as closed thrust loop and propulsion dynamics, are not necessarily time scale separated from the desired closed loop path dynamics, the plant will not perfectly follow the desired dynamics given by the reference models. A full inversion structure including an inversion of the actuation dynamics may be complex, since the actuators themselves may have an arbitrary nonlinear dynamic behavior, with deflection and rate limitations. The dynamics of the closed inner loops have to be considered by other means.

The Pseudo-Control Hedging (PCH) approach [17, 18], allows for a different way to take the actuation dynamics into account. It was originally introduced as a method for dealing with actuation nonlinearities in more complex inversion-based control structures with adaptive elements but has been shown suitable for propagating limitations in the dynamics and reaction deficits of cascaded inversion control structures [19]. The reaction dynamics of the plant is accounted for by feeding back the difference between command and actual plant response, a so-called *hedging signal*, and subtracting it before the reference state integration in order to slow down the reference model output and allow it to track the achievable plant trajectory. The hedging signals are feedback signals; thus, the reference model cannot be considered as an open-loop, feedforward element when considering the stability of the overall system.

Actuator Dynamics

The transfer function matrix between a virtual control input \mathbf{u}_C from the inversion, and the actual control \mathbf{u} entering the plant, is given by

$$\frac{\mathbf{u}}{\mathbf{u}_C} = \mathbf{G}_A(s), \quad (4.44)$$

with $\mathbf{G}_A(s) = \text{diag}\{G_{A,1}(s), \dots, G_{A,m}(s)\}$. The relation between the commanded and actual inputs are in the following summarized as $\mathbf{u} = \mathbf{G}_A \mathbf{u}_C$.

With the linearizing control law, Eq. (4.34), the true plant input becomes

$$\mathbf{u} = \mathbf{G}_A \mathbf{u}_C = \mathbf{G}_A \left[\hat{\mathbf{A}}^{-1}(\mathbf{x}) [\mathbf{v} - \hat{\mathbf{b}}(\mathbf{x})] \right]. \quad (4.45)$$

The deviation between the reference and plant can be split up between the deviation due to modeling error and the deviation due to actuation dynamics,

$$\begin{aligned}
y_i^{(r_i)}(\mathbf{x}, \mathbf{u}) - \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}_C) &= y_i^{(r_i)}(\mathbf{x}, \mathbf{u}) - \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}) + \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}) - \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}_C) \\
&= \Delta_i(\mathbf{x}, \mathbf{u}) + \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}) - \hat{y}_i^{(r_i)}(\mathbf{x}, \mathbf{u}_C).
\end{aligned} \tag{4.46}$$

The deviation due to uncertainties in the inversion of the true plant dynamics is not directly measurable, whereas the deviation due to the actuation dynamics may be estimated using the plant model, and the measured or estimated plant input.

Expected Reaction Deficit

If it is possible to measure the real output of the actuation dynamics, i.e., the real value of \mathbf{u} is known, the real plant response can be estimated using the inversion model,

$$\hat{\mathbf{v}} = \hat{\mathbf{F}}(\mathbf{x}, \mathbf{u}) = \hat{\mathbf{F}}(\mathbf{x}, \mathbf{G}_A \mathbf{u}_C), \tag{4.47}$$

with \mathbf{G}_A being the true actuation dynamics. If the output of the actuation dynamics is not measurable, i.e., the real value of \mathbf{u} is unknown, the plant response can be estimated using the plant model and an estimation $\hat{\mathbf{G}}_A$ of the actuation dynamics,

$$\hat{\mathbf{v}} = \hat{\mathbf{F}}(\mathbf{x}, \mathbf{u}) = \hat{\mathbf{F}}(\mathbf{x}, \hat{\mathbf{G}}_A \mathbf{u}_C). \tag{4.48}$$

With the estimated plant response $\hat{\mathbf{v}}$, the expected reaction deficit of the actuation loop can be determined, i.e., the hedge signal,

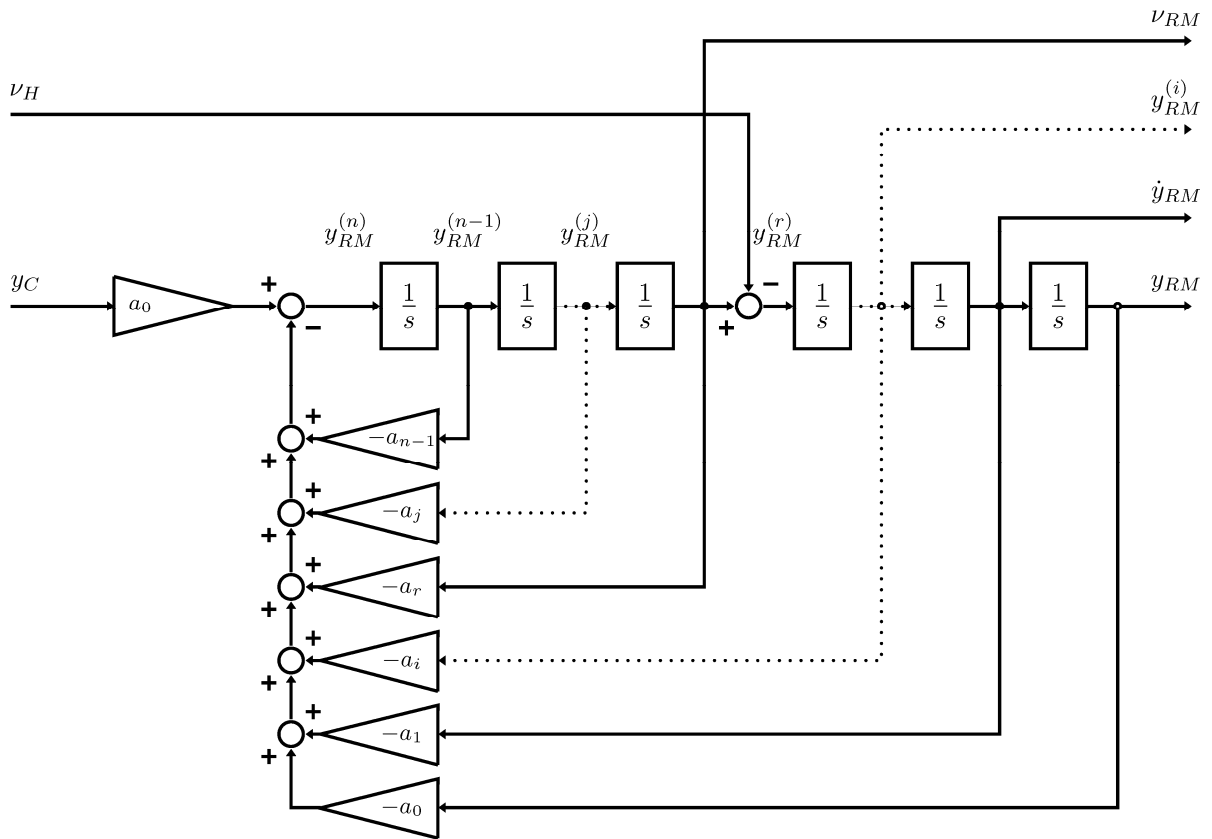
$$\begin{aligned}
\mathbf{v}_H &= \mathbf{v} - \hat{\mathbf{v}} \\
&= \mathbf{v} - \hat{\mathbf{F}}(\mathbf{x}, \mathbf{u}) \\
&= \mathbf{v} - \hat{\mathbf{F}}(\mathbf{x}, \mathbf{G}_A \mathbf{u}_C) \\
&= \mathbf{v} - \hat{\mathbf{F}}(\mathbf{x}, \mathbf{G}_A [\hat{\mathbf{F}}^{-1}(\mathbf{x}, \mathbf{v})]).
\end{aligned} \tag{4.49}$$

Modified Reference Model Dynamics

To keep the reference model output consistent with the true reaction of the plant, this hedge signal is fed back to the reference model to slow down the reference dynamics. The hedging signal is subtracted from the highest derivate of the reference dynamics, after the pseudo control but before the innermost integrator, in order to slow the reference dynamics down by the expected reaction deficit,

$$\begin{aligned}
y_{i, RM}^{(r_i)} &= v_{i, RM} - v_{i, H} \\
&= -a_{r-1} \cdot y_{i, RM}^{(r_i-1)} - \dots - a_1 \cdot \dot{y}_{i, RM} - a_0 \cdot y_{i, RM} - v_{i, H},
\end{aligned} \tag{4.50}$$

with the reference model pseudo control $v_{i, RM}$ calculated as previously in Eq. (4.24).


 Figure 4.4: Linear reference model of order n with Pseudo-Control Hedging.

Modified Error Dynamics

Analogously to the analysis of the error dynamics without PCH, error dynamics including PCH is analyzed for the SISO case, omitting the reference model index i . The error dynamics including the hedging signal is given by extending the Eq. (4.41),

$$\begin{aligned}
 y^{(r)} &= v + \Delta - v_H \\
 &= v_{RM} + v_{EC} + \Delta - v_H \\
 &= v_{RM} + v_{EC} + \Delta - v_H \\
 &= v_{RM} + \sum_{i=1}^r k_{P,i-1} \cdot e^{(i-1)} + \Delta - v_H.
 \end{aligned} \tag{4.51}$$

According to Eq. (4.50),

$$y_{RM}^{(r)} = v_{RM} - v_H \tag{4.52}$$

which together with Eq. (4.42) gives

$$e^{(r)} = y_{RM}^{(r)} - y^{(r)} = v_{RM} - y^{(r)} - v_H. \tag{4.53}$$

The differential equation for control error including PCH is then given by

$$e^{(r)} + \sum_{i=1}^r k_{p,i-1} \cdot e^{(i-1)} = -\Delta \quad (4.54)$$

which is identical to the error dynamics without actuation dynamics and PCH, Eq. (4.43). Hence, PCH "hides" the actuator dynamics from the error dynamics. Another advantage of PCH is that in the case the error controller is amended by an additional integral part in order to achieve steady state accuracy, so-called integrator wind-up, i.e. excessive integrator loading when compensating control errors due to saturated actuation, is avoided, while the hedge signal prevents the reference model state to diverge.

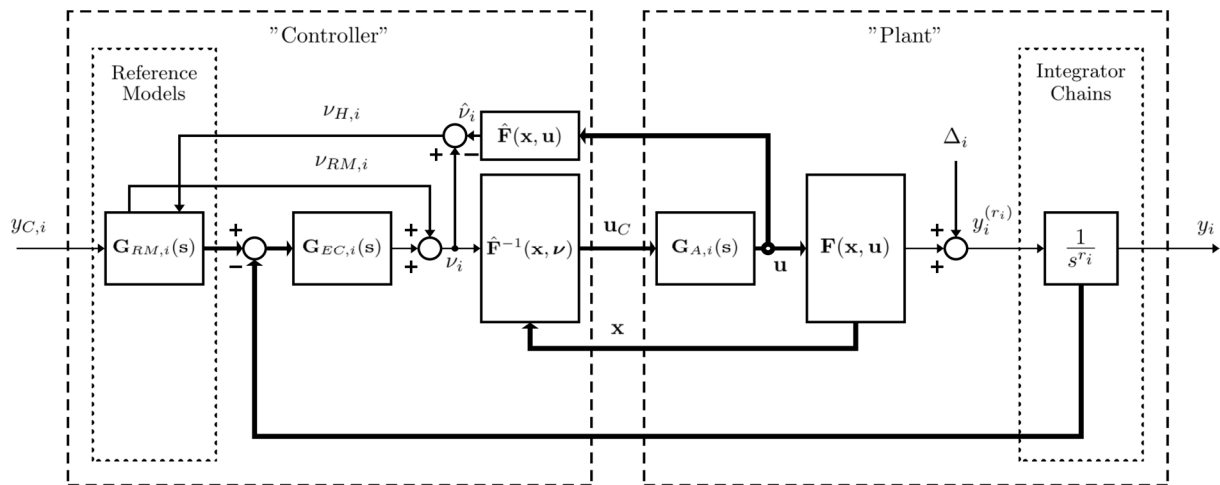


Figure 4.5: Reference model based dynamic inversion with error dynamics stabilization and Pseudo-Control Hedging.

Implications

The introduction of the PCH changes the feedback structure of the controller. The hedging signal is a feedback signal, which means that the reference model cannot be considered as an open-loop, feedforward element when considering the stability of the overall system. The hedging signal is not only active for hard nonlinearities such as limitations and actuation saturations but is constantly active adjusting for inner loop dynamics (inner loop dynamics limited by the inherent aircraft moment dynamics and the actuation dynamics).

Stability analysis must consider the error between the reference model state and the plant response, and the error between commanded value and the reference model state, with the required stability margins (loop gain and phase margins) of the closed-loop system including the reference models.

5 System Definition

This chapter describes the system definition activities of the AFCS according to the System Definition Track of the System Development Process defined in Section 2.3.2:

- System Functions and Architecture
- Controller Design
- Gain and Parameter Design
- Human-Machine Interface Design (design aspects coupled with Controller Design)

5.1 System Functions & Architecture

The AFCS concept is designed and implemented as part of a generic, modular flight control system, see the example life cycle model in Figure 2.2, Section 2.1.3, adaptable to a multitude of aircraft application platforms with separate life cycles. The scope and performance of the system functions are in each case driven by the desired operational concept and mission needs, as defined in the concept phase (see Figure 2.2).

For the development of the AFCS concept, no application-specific higher-level requirements drive the AFCS design in a classical top-down sense. The AFCS functionalities shall enable flexible application to a broad range of application platforms, with flexible integration of state-of-the-art automatic control modes to enable a range of operational scenarios and maximize the number of possible use cases of the system on different application platforms. The functionalities are to be portable, and a modular architecture shall allow the easy configuration and verification of the desired functionalities, with retesting concentrated to application platform-specific functionalities and performance requirements.

An example use case is provided, that of an Optionally-Piloted Vehicle (OPV) AFCS, described in Section 5.1.1, representing possible application scenarios with associated typical functionalities, as a baseline for the definition of the system functions.

The AFCS system functions, described in Section 5.1.2 are grouped according to the functional elements in the principal system architecture in Figure 2.11, Section 2.3.2, each with their respective application-generic and application-specific functions.

The application-generic functions are those where the desired functional behavior is essentially independent of the application scenario, only parameterized to meet application-specific performance requirements. The application-specific functions are those where the functional

design is customized to meet the specific characteristics of the application scenario or system integration environment, for example automatic trim functions, actuation system interfacing, or override and other safety functions.

The set of system functions and functional descriptions are the main input to the system-level FHA, described in Chapter 6, together with the operational concept (operational scenarios, flight phases), and applicable certification regulations and guidance material. The FHA generates the system-level FHA together with any safety-related system functions and constraints.

The main output from the function development is a set of system-level functional and performance requirements, formalizing the functional description and specifying the desired and adequate system performance.

The system architecture is driven by the required system-level FDAL and the system environment into which the AFCS is integrated. The system architecture is dependent on the level of hardware and software redundancy and independence, associated IDALs, integration with legacy sensor and actuation systems, display and mode control interfaces, aircraft trim systems, and physical installation constraints.

System requirements relating to the physical system architecture and items, such as physical and installation requirements, reliability, and availability requirements, as well as maintainability requirements are outside the scope of this thesis.

5.1.1 Example Use Case: Optionally Piloted Vehicle AFCS

An OPV is an aircraft that is equipped and can be operated as a conventional piloted aircraft and as a UAV. An OPV is thus able to fly with or without a human pilot at the controls (on board the aircraft). Examples of OPVs include the Cessna 337-O2 Skymaster OPV Pelican of the Center for Interdisciplinary Remotely-Piloted Aircraft Studies (CIRPAS) [110] and the research platform of the TUM-FSD, the DA42 OE-FSD, described in [90], see Figure 1.4, Section 1.4.1.

OPVs are used as a safe, reliable, and low-cost platform alternative to UAVs in research or experimentation, due to the ability to take off and land at conventional airfields, and transfer to a designated flight test area, with a safety pilot monitoring of experimental software. A typical OPV mission, illustrated in Figure 5.1, thus includes phases where it operates as a conventional aircraft, with standard AFCS functionalities, and mission phases either automatically controlled or remotely piloted, where direct human control is desired as an immediate backup option. If equipped and certified to operate as a UAV, the OPV is further unimpeded by the physiological limitations of an onboard pilot and may operate under more adverse conditions or with greater endurance.

From an AFCS perspective, an AFCS enabling OPV operations needs to combine the functions of a general aviation AFCS with functions of a UAV AFCS, enabling a higher level of automatic control and more interfaces for external guidance or mission management systems, ground control systems, and system automation logic.

Where general aviation autopilots have limited control authority and bandwidth, an OPV AFCS would need full-envelope, high-bandwidth control authority to enable emulation of UAV operations or perform mission-specific aerial work flight patterns.

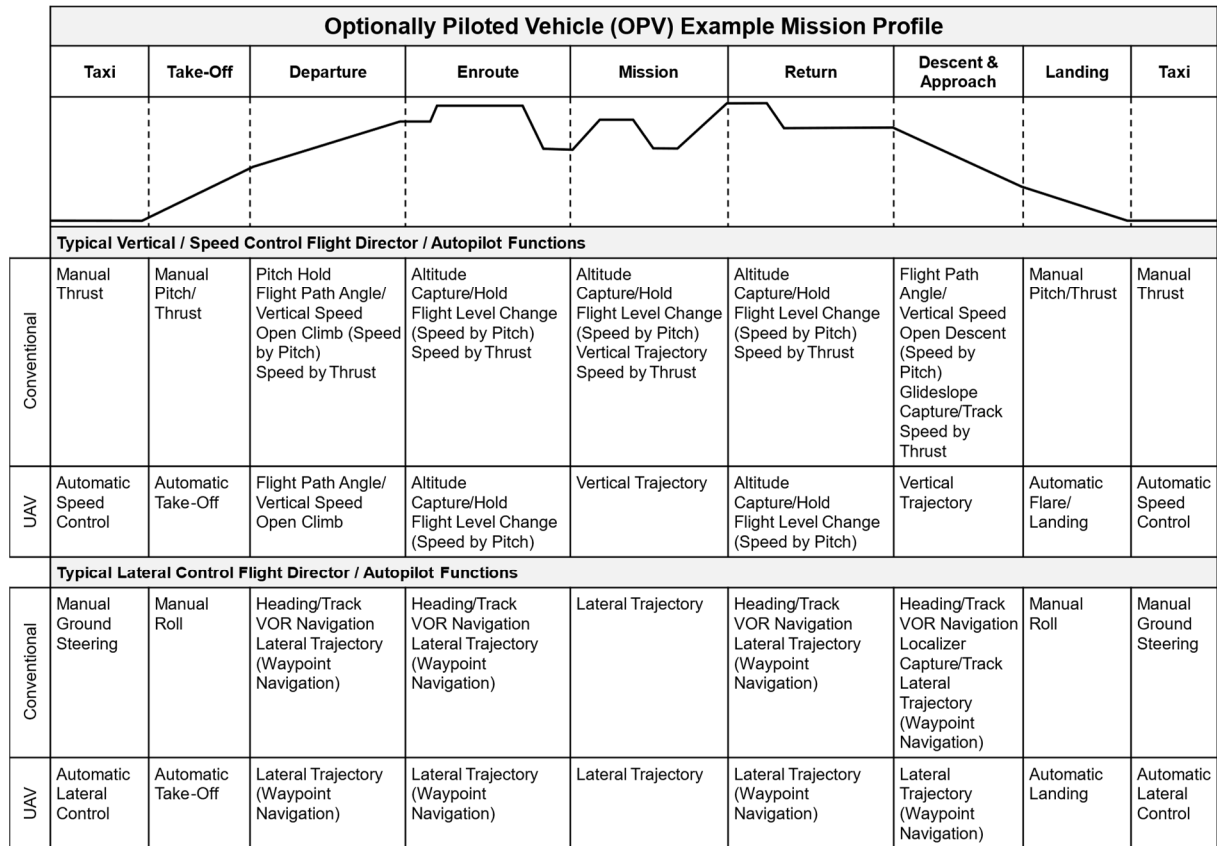


Figure 5.1: Example mission profile for an OPV, with typical vertical/speed control and lateral control modes for conventional and UAV operation.

5.1.2 AFCS System Functions

The AFCS system functions are developed according to the functional elements in the principal system architecture in Figure 2.11, Section 2.3.1:

- Automatic Flight Control Functions
- HMI Functions
- Autopilot Functions
- Autothrust Functions
- Safety Functions

The AFCS system functions are further grouped into

- Application-Generic Functions:
 - Functional behavior independent of the application scenario
 - Parameterized to meet application-specific performance requirements
- Application-Specific Functions:

- Functional behavior is customized from the specific characteristics of the application scenario or system integration environment

An overview of identified application-generic and application-specific functions is given in Table 5.1.

Table 5.1: Application-generic and application-specific system functions overview.

System Functions	Application-Generic	Application-Specific
Automatic Flight Control Functions	<ul style="list-style-type: none"> • Guidance modes <ul style="list-style-type: none"> ○ Selected vertical/lateral/speed ○ Managed vertical/lateral/speed <p style="text-align: right;">See Table 5.2</p>	
	<ul style="list-style-type: none"> • Vertical/lateral path control modes <ul style="list-style-type: none"> ○ Vertical/lateral path control ○ Climb/descent (open/constrained) • Vertical/lateral trajectory control modes <ul style="list-style-type: none"> ○ Vertical/lateral trajectory navigation ○ Altitude capture hold ○ Radio navigation/approach • Speed control modes <ul style="list-style-type: none"> ○ Speed by thrust ○ Speed by pitch <p style="text-align: right;">See Table 5.3</p>	<ul style="list-style-type: none"> • Attitude control modes <ul style="list-style-type: none"> ○ Attitude hold ○ Attitude protection (attitude integrity) • Direct path curvature modes <ul style="list-style-type: none"> ○ Direct vertical/lateral path curvature ○ Direct vertical/lateral path curvature stick mode
	<ul style="list-style-type: none"> • Envelope protection modes (energy integrity) • Control objective prioritization modes (maneuvering integrity) <ul style="list-style-type: none"> ○ Path/speed priority ○ Vertical/lateral plane priority <p style="text-align: right;">See Table 5.4</p>	<ul style="list-style-type: none"> • Envelope protection modes (aerodynamic integrity, structural integrity)
Human-Machine-Interface Functions	<ul style="list-style-type: none"> • Mode and target selection/annunciation • Actuation resource selection/annunciation • Flight director <p style="text-align: right;">See Table 5.5</p>	<ul style="list-style-type: none"> • Control Wheel Steering
Autopilot Functions	<ul style="list-style-type: none"> • Automatic pitch/roll/yaw control • Autopilot engagement/disengagement <p style="text-align: right;">See Table 5.6</p>	<ul style="list-style-type: none"> • Stability augmentation modes • Turn coordination • Automatic trim • Individual pitch/roll/yaw control engagement/disengagement
Autothrust Functions	<ul style="list-style-type: none"> • Automatic thrust control • Autothrust engagement/disengagement <p style="text-align: right;">See Table 5.6</p>	
Safety Functions	<ul style="list-style-type: none"> • Input monitoring functions • Subsystem monitoring functions • Autopilot/Autothrust automatic disengagement <p style="text-align: right;">See Table 5.7</p>	<ul style="list-style-type: none"> • Autopilot/Autothrust override • Autopilot/Autothrust safety interlock

Application-Generic Functions

The thesis at hand focuses on the design and implementation of the application-generic functions, which are further described in Table 5.2-Table 5.7.

Selected vs. Managed Operation (VNAV/LNAV/SPD FMS ON/OFF)

The AFCS has two guidance modes, *selected* and *managed operation*. In selected mode, the AFCS operates as a conventional manned aircraft, with the pilot directly commanding the desired modes and target via the onboard HMI. In managed mode, the AFCS takes desired modes and targets as external guidance from a Flight Management System (FMS) or a Mission Management System (MMS).

Selected and managed operation is possible for each control plane separately, i.e., managed lateral guidance from an FMS/MMS is possible with selected vertical guidance from the HMI. The guidance mode designations and combinations are described in Table 5.2, with reference to applicable paragraphs for functional requirements in DO-325 [35] and AS94900A [103].

Table 5.2: Guidance modes and combinations for vertical/lateral plane and speed.

Guidance Modes	Vertical Plane	Lateral Plane	Speed	Description	Reference Guidance Material
Selected Operation (SEL)	VNAV FMS OFF			Vertical modes and targets from HMI	DO-325 1.5.8 AS94900A 3.7.7.1.9
		LNAV FMS OFF		Lateral modes and targets from HMI	DO-325 1.5.8 AS94900A 3.7.7.1.9
			SPD FMS OFF	Speed modes and targets from HMI	DO-325 1.5.8 AS94900A 3.7.7.1.9
Managed Operation (MAN)	VNAV FMS ON			Vertical modes and targets from MMS/FMS	DO-325 1.5.4 AS94900A 3.3.4.3
		LNAV FMS ON		Lateral modes and targets from MMS/FMS	DO-325 1.5.4 AS94900A 3.3.4.3
			SPD FMS ON	Speed modes and targets from MMS/FMS	AS94900A 3.3.4.3

Coupled Path, Speed and Trajectory Control Modes

The AFCS provides automatic control of the vertical and lateral flight path plane, as well as control of the speed along the flight path.

Control modes for the vertical plane and speed are always coupled, i.e., they are not possible to combine arbitrarily. The control modes for the lateral plane are decoupled from the vertical and speed control functions, except for in approach mode, when vertical, lateral, and speed modes are all coupled.

The control modes for the vertical plane, lateral plane, and speed are likewise coupled to the autopilot and autothrust functions that provide the primary automatic pitch/roll/yaw actuation and automatic thrust control, realizing the desired flight path and acceleration.

When the vertical plane is explicitly controlled by the vertical control mode, for example in flight path angle or altitude hold mode, the autopilot provides the corresponding automatic pitch/roll/yaw control, and the autothrust provides the corresponding desired acceleration.

When the vertical plane is indirectly controlled by the vertical mode, for example in an open climb or descent mode with speed by pitch, the autopilot provides the corresponding pitch to maintain desired acceleration, while the autothrust provides a desired climb or descent thrust setting. The actual climb or descent angle is thus not actively controlled but results from the current energy balance (although subject to limitations such as no negative flight path in climb mode and vice versa).

The principal automatic flight control mode and autopilot/autothrust couplings are illustrated in Figure 5.2.

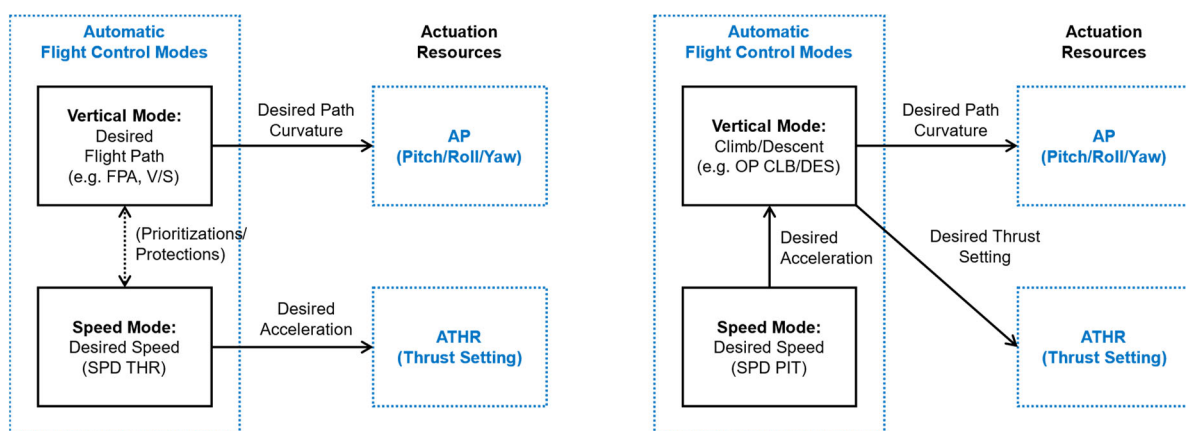


Figure 5.2: Principal coupling between automatic flight control modes and autopilot/autothrust functions.

The control mode designations and combinations are described in Table 5.3, with reference to applicable paragraphs for desired control mode behavior and adequate performance requirements in DO-325 [35] and AS94900A [103].

Control Objective Prioritization and Envelope Protection Modes

The control objective prioritization and envelope protection modes provide deterministic system behavior in the case of saturated energy rate or specific force for maneuvering.

In the case of saturated energy rate, which precludes concurrent vertical flight path speed target tracking, the path/speed priority modes prioritize between vertical path or speed tracking, with or without the autothrust engaged.

In the case of saturated maneuvering force, the vertical/lateral plane priority modes prioritize between maneuvering in the vertical or lateral plane.

In the case of the aircraft approaching the minimum or maximum airspeed, the energy protection mode limits the vertical flight path to prevent exceeding minimum speed, while available specific excess power is distributed in favor of reducing/increasing acceleration.

Maneuvering in the lateral plane is also limited to allow desired maneuvering in the vertical plane (vertical plane maneuvering priority).

The control objective prioritization and envelope protection modes are described in Table 5.4, with reference to applicable paragraphs for desired control mode behavior and adequate performance requirements in DO-325 [35] and AS94900A [103].

HMI Functions

The HMI is the main interface for the pilot or operator to activate/arm/deactivate the automatic flight control modes, to preselect and confirm control mode targets, and to select desired actuation resources (autopilot and autothrust). The HMI provides annunciation regarding active/armed modes and targets, mode transitions, and active actuation resources.

The HMI functions are described in Table 5.5, with reference to applicable paragraphs for desired control mode behavior and adequate performance requirements in DO-325 [35] and AS94900A [103].

Autopilot and Autothrust Functions

The autopilot provides automatic control of the pitch, roll, and yaw axis, according to a desired flight path curvature, and closes the control loop for the automatic flight control modes (without autopilot engaged, the pilot closes the loop when flying manually with flight director guidance). The autothrust provides automatic control of the thrust or power levers to provide a desired acceleration along the flight path or a fixed climb or descent thrust.

The autopilot and autothrust functions are described in Table 5.6, with reference to applicable paragraphs for desired control mode behavior and adequate performance requirements in DO-325 [35] and AS94900A [103].

Safety Functions

The safety functions provide monitoring of guidance and sensor inputs and monitoring of subsystem health status. In the case of erroneous inputs or unhealthy system components, the autopilot and autothrust are automatically disengaged. The disengage mechanism may be application-specific, either a hardware disconnect in the case of a conventional AFCS with a mechanical control system, or a software disconnect in the case of a UAV with an FBW flight control system.

Depending on the application, specific safety functions may include autopilot/autothrust override functions or safety interlocks to prevent unintended operation.

The application-generic safety-related functions are described in Table 5.7, with reference to applicable paragraphs for desired behavior and adequate performance requirements in DO-325 [35] and AS94900A [103].

Table 5.3: Coupled path and trajectory control modes for vertical/lateral plane and speed.

Control Mode	Vertical Plane	Lateral Plane	Speed	Description	Reference Guidance Material
Path Control	Flight Path Angle (FPA)		Speed by Thrust (SPD THR)	AP acquisition and tracking of flight path angle target ATHR acquisition and tracking of airspeed target	DO-325 1.5.1 DO-325 1.5.10 AS94900A 3.3.4.2.6
	Vertical Speed (V/S)		Speed by Thrust (SPD THR)	AP acquisition and tracking of vertical speed target ATHR acquisition and tracking of airspeed target	DO-325 1.5.1 DO-325 1.5.10 AS94900A 3.3.4.2.6
		Track (TRK)		AP acquisition and tracking of track angle target	DO-325 1.5.2 AS94900A 3.3.4.2.3 AS94900A 3.3.4.2.4
		Heading (HDG)		AP acquisition and tracking of heading target	DO-325 1.5.2 AS94900A 3.3.4.2.3 AS94900A 3.3.4.2.4
			Speed by Pitch (SPD PIT)	AP acquisition and tracking of airspeed target ATHR inactive	DO-325 1.5.1 DO-325 1.5.10 AS94900A 3.3.4.2.6
	Open Climb / Descent (OP CLB / DES)		Speed by Pitch (SPD PIT)	Flight Level Change (FLCH) coupled mode AP acquisition and tracking of airspeed target for climb or descent ATHR climb/descent thrust	DO-325 1.5.1 DO-325 1.5.10 AS94900A 3.3.4.2.6
Trajectory Control	Vertical Trajectory Navigation (VNAV)		Speed by Thrust (SPD THR)	AP tracking of vertical trajectory commands from FMS/MMS ATHR acquisition and tracking of airspeed target	DO-325 1.5.4 DO-325 1.5.10 AS94900A 3.3.4.2.6 AS94900A 3.3.4.3
		Lateral Trajectory Navigation (LNAV)		AP tracking of lateral trajectory commands from FMS/MMS	DO-325 1.5.4 AS94900A 3.3.4.3
	Altitude Capture (ALT*) Altitude Hold (ALT)		Speed by Thrust (SPD THR)	AP capture and tracking of altitude target ATHR acquisition and tracking of airspeed target	DO-325 1.5.1 DO-325 1.5.10 DO-325 2.2.10 AS94900A 3.3.4.2.1 AS94900A 3.3.4.2.6
		VOR/Localizer Capture (VOR/LOC*) VOR/Localizer Track (VOR/LOC)		AP capture and tracking of VOR course or localizer signal for lateral navigation	DO-325 1.5.2 AS94900A 3.3.4.3.6 AS94900A 3.3.4.3.7
	Glideslope Capture (G/S*) Glideslope Track (G/S)	Localizer Capture (LOC*) Localizer Track (LOC)		Speed by Thrust (SPD THR)	Approach (APPR) coupled mode AP capture and tracking of glideslope signal AP capture and tracking of localizer signal ATHR acquisition and tracking of airspeed target

Table 5.4: Coupled control objective prioritization modes for vertical/lateral plane and speed.

Priority Modes	Vertical Plane	Lateral Plane	Speed	Description	Reference Guidance Material
Path/Speed Priority	Path Priority (Absolute) (PPRIO ABS)		Path Priority (Absolute) (PPRIO ABS)	Speed tracking is limited to allow desired vertical path to be tracked Available specific excess power distributed in favor of path tracking	AS94900A 3.2.7.3.8
	Path Priority (Relative) (PPRIO REL)			Speed tracking is limited to allow desired vertical path to be tracked	AS94900A 3.2.7.3.8
	Speed Priority (Absolute) (VPRIO ABS)		Speed Priority (Absolute) (VPRIO ABS)	Vertical flight path is limited to allow desired speed to be tracked Available specific excess power distributed in favor of speed tracking	AS94900A 3.2.7.3.8
	Speed Priority (Relative) (VPRIO REL)			Vertical flight path is limited to allow desired acceleration to be tracked	AS94900A 3.2.7.3.8
Vertical/Lateral Plane Priority	Vertical Plane Priority (VERT)	Vertical Plane Priority (VERT)		Maneuvering in lateral plane limited to allow desired maneuvering in vertical plane	AS94900A 3.2.7.3.8
	Lateral Plane Priority (LAT)	Lateral Plane Priority (LAT)		Maneuvering in vertical plane limited to allow desired maneuvering in lateral plane	AS94900A 3.2.7.3.8
Energy Protection	Low Speed Protection (LOW SPD PROT)	Vertical Plane Priority (VERT)	Low Speed Protection (LOW SPD PROT)	Vertical flight path is limited to prevent exceeding minimum speed Available specific excess power distributed in favor of acceleration Maneuvering in lateral plane limited to allow desired maneuvering in vertical plane	DO-325 1.5.5 DO-325 2.2.9
	High Speed Protection (HIGH SPD PROT)	Vertical Plane Priority (VERT)	High Speed Protection (HIGH SPD PROT)	Vertical flight path is limited to prevent exceeding maximum speed Available specific excess power distributed in favor of acceleration Maneuvering in lateral plane limited to allow desired maneuvering in vertical plane	DO-325 1.5.5 DO-325 2.2.9

Table 5.5: HMI functions and guidance material references.

HMI Function	Subfunction	Description	Reference Guidance Material
Mode/Target Selection	Active Mode Selection	Selection of active mode in selected operation	DO-325 1.5.8 DO-325 2.2.7 AS94900A 3.7.7.1.9
	Target Preselection/ Confirmation	Preselection and confirmation of target in selected operation	DO-325 1.5.8 DO-325 2.2.7 AS94900A 3.7.7.1.9
	Target Synchronization	Preselected target is synchronized to the current aircraft state	DO-325 1.5.8 DO-325 2.2.7 AS94900A 3.7.7.1.9
	Mode/Target Source Selection	Selection of mode and target source (selected/managed operation)	DO-325 1.5.8 DO-325 2.2.7 AS94900A 3.7.7.1.9
Mode/Target Annunciation	Active/Armed Mode Annunciation	Annunciation of active and armed control modes	DO-325 2.2.7 AS94900A 3.6.9.9 AS94900A 3.3.1.3
	Mode Transition Annunciation	Annunciation of manual and automatic mode transitions	DO-325 2.2.7 AS94900A 3.6.9.9 AS94900A 3.3.1.3
	Envelope Protection Annunciation	Annunciation of active envelope protections	DO-325 2.2.8 AS94900A 3.6.9.9 AS94900A 3.3.1.3
	Active/Preselected Target Annunciation	Annunciation of preselected and active target	DO-325 2.2.7 AS94900A 3.6.9.9 AS94900A 3.3.1.3
	Mode/Target Source Annunciation	Annunciation of current mode and target source (selected/managed operation)	DO-325 2.2.7 AS94900A 3.6.9.9 AS94900A 3.3.1.3
Flight Director (FD)	FD Engagement	Manual engagement of flight director guidance cues	DO-325 2.2.5.1
	FD Disengagement	Manual disengagement of flight director guidance cues	DO-325 2.2.5.2
	Display Vertical Flight Guidance	Vertical guidance cues consistent with the current control mode	DO-325 2.2.5
	Display Lateral Flight Guidance	Lateral guidance cues consistent with the current control mode	DO-325 2.2.5
	Display Thrust Guidance	Thrust/throttle guidance cues consistent with the current control mode	DO-325 2.2.5
Actuation Resource Selection	Autopilot Selection	Dedicated control for autopilot engagement/ disengagement selection	DO-325 2.2.1.1.1 DO-325 2.2.1.1.2
	Autothrust Selection	Dedicated control for autothrust engagement/ disengagement selection	DO-325 2.2.6.1.1 DO-325 2.2.6.1.2
Actuation Resource Annunciation	Autopilot Engagement State Annunciation	Annunciation of autopilot engagement state	DO-325 2.2.1.1.1 DO-325 2.2.1.1.2
	Autothrust Engagement State Annunciation	Annunciation of autopilot engagement state	DO-325 2.2.6.1.1 DO-325 2.2.6.1.2

Table 5.6: Autopilot and autothrust functions and guidance material references.

Function	Subfunction	Description	Reference Guidance Material
Autopilot (AP)	Provide automatic pitch/roll/yaw control	AP provides automatic pitch/roll/yaw control consistent with the current AFCF mode of operation	DO-325 1.5.1 DO-325 1.5.2 DO-325 2.2.1
	AP Engagement	Engagement of autopilot control of actuation resources	DO-325 2.2.1.1.1 AS94900A 3.3.3.5
	AP Disengagement	Disengagement of autopilot control of actuation resources	DO-325 2.2.1.1.2 AS94900A 3.3.1.2
Autothrust (ATHR)	Provide automatic thrust control	AP provides automatic thrust control consistent with the current AFCF mode of operation	DO-325 1.5.10 DO-325 2.2.6
	ATHR Engagement	Manual engagement of autothrust control of actuation resources	DO-325 2.2.6.1.1
	ATHR Disengagement	Manual disengagement of autothrust control of actuation resources	DO-325 2.2.6.1.2 AS94900A 3.3.1.2

Table 5.7: Safety functions and guidance material references.

Function	Subfunction	Description	Reference Guidance Material
Input Monitoring	Monitor External Guidance	Provide monitoring of input range, noise, update rates etc.	AS94900A 3.3.4.3.1 AS94900A 3.3.4.3.2 AS94900A 3.3.4.3.4 AS94900A 3.3.4.3.5
	Monitor Sensor Inputs	Provide monitoring of input range, noise, update rates etc.	AS94900A 3.2.6.2 AS94900A 3.2.7.3.7 AS94900 3.6.10
Subsystem Monitoring	Monitor Subsystem Health	Provide monitoring of subsystem and component health status	AS94900A 3.2.6.2 AS94900A 3.2.7.3.7
AP/ATHR Disengagement	AP Automatic Disengagement	Provide automatic AP disengagement	AS94900A 3.3.4.1.4
	ATHR Automatic Disengagement	Provide automatic ATHR disengagement	AS94900A 3.3.4.1.4

Application-Specific Functions

The application-specific functions are designed and implemented based on the context-specific needs and constraints for a given aircraft application. The thesis at hand focuses on the development of the application-generic functions described in the previous section.

Application-Specific Automatic Flight Control Functions

Application-specific automatic control modes may include for example attitude control modes (pitch/roll hold, attitude protection), and additional envelope protection modes (such as angle of attack or sideslip protections). Application-specific automatic control modes may also include a set of backup control laws for reduced sets of sensor inputs (for example no GNSS/INS flight path information), with reduced functionalities and operational envelope, maximizing survivability of the aircraft, especially desired for UAV AFCS applications.

Application-Specific HMI Functions

Application-specific HMI functions may include for example Control Wheel Steering (CWS) as an additional means for the pilot to select path or attitude control targets.

Application-Specific Autopilot Functions

Application-specific autopilot functions may include stability augmentation modes (for example separate yaw damper mode) and automatic trim functions based on the aircraft-specific trim design.

Application-Specific Safety Functions

Application-specific safety functions depend on the AFCS integration and interactions with existent flight control systems elements (mechanical controls with dedicated AFCS servos, or integration with FBW FCS). Safety functions include for example autopilot override functions to enable manual control in case of erroneous operation, or safety interlock functions to inhibit AFCS operation under certain circumstances.

5.1.3 System Requirements

Structuring and Categorization

The set of system requirements is developed from the system functions and applicable guidance material. The system requirements related to the functional elements AFCE, HMI functions, AP/ATHR functions, and safety functions are grouped into

- General System Requirements: applicable to all functions of that functional element
- Function-Specific System Requirements: applicable to a single or subset of functions

The system requirements are categorized depending on their origin, in line with the ARP4754B requirements capture process (see Section 2.2.1) as:

- Functional Requirements (originating system functions and architecture development)
- Safety Requirements (originating from the safety assessment process)
- Certification Requirements (originating from airworthiness regulations)
- Derived Requirements (originating from the design process)

The thesis at hand focuses on the design and implementation of the application-generic functions, described in Table 5.2-Table 5.7, which are captured as general and function-specific functional requirements. The functional requirements are further identified by their type, in line with the ARP4754B requirements capture process (see Section 2.2.1), as

- Functional Behavior Requirements
- Operational Requirements
- Performance Requirements
- Interface Requirements

The principal development of the system functions to sets of general and function-specific functional requirements is illustrated in Figure 5.3.

Table 5.8 gives an overview of general and function-specific functional requirements developed from the application-generic functions defined in Section 5.1.2

Functional requirements further include physical and installation requirements, and maintainability requirements associated with the aircraft-specific physical hardware and integration environment. These are highly application-specific and are not elaborated here.

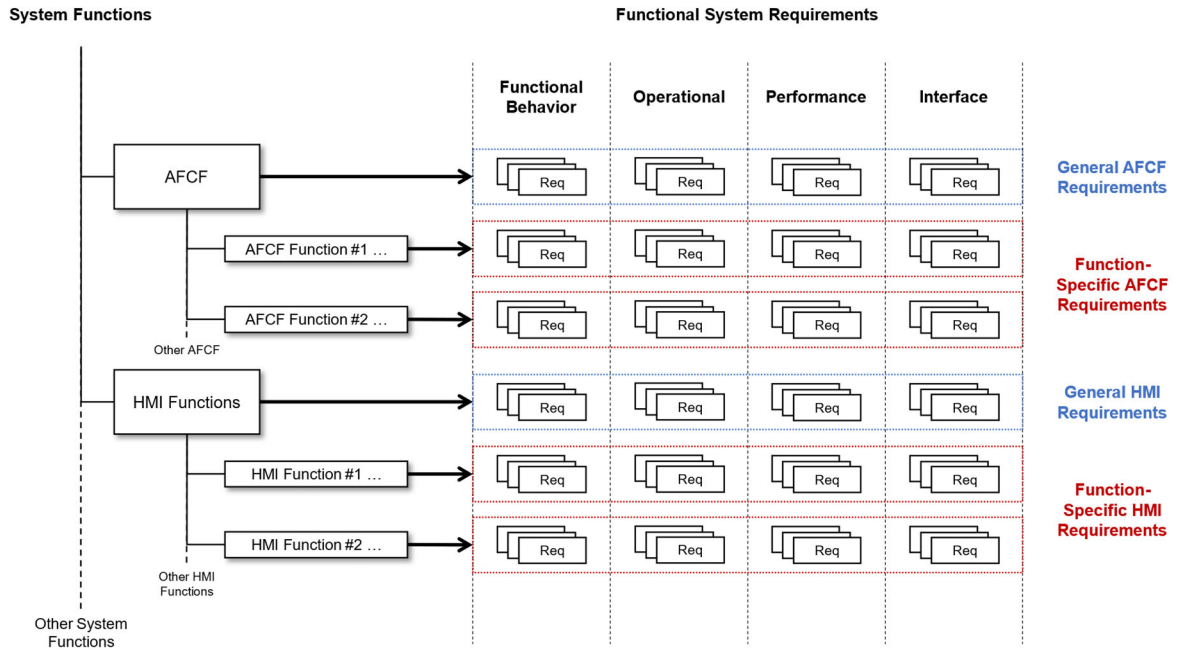


Figure 5.3: Principal development of system functions to sets of general and function-specific functional requirements.

Table 5.8: Grouping of general and function-specific functional requirements.

Functional Requirements	General System Requirements	Function-Specific System Requirements
Functional Behavior	<ul style="list-style-type: none"> Control and guidance modes Envelope protection and control objective prioritization modes 	<ul style="list-style-type: none"> Intended control mode behavior Intended protection/prioritization mode behavior
Operational Characteristics	<ul style="list-style-type: none"> Manual/automatic mode transitions and transition criteria Mode and target selection Mode and target annunciation and monitoring 	
Performance	<ul style="list-style-type: none"> Operational envelope Aircraft configurations Command limits General performance, e.g. stability and robustness requirements 	<ul style="list-style-type: none"> Function-specific envelope Function-specific aircraft configurations Function-specific/flight phase command limits Function-specific/flight phase desired/adequate performance and robustness
Interfaces	<ul style="list-style-type: none"> Command inputs for selected and managed operation Sensor inputs (range, resolution, update rates) Mode control and monitoring interfaces 	

A subset of the functional requirements counts as high-level requirements according to [49], from which a direct implementation of the design model may be made. The design model itself, and attributes specified therein, such as data types and signal ranges, count as software architecture and software low-level requirements. An overview of the relation between general and function-specific requirements, design model, and verification artifacts is found in Figure 5.4.

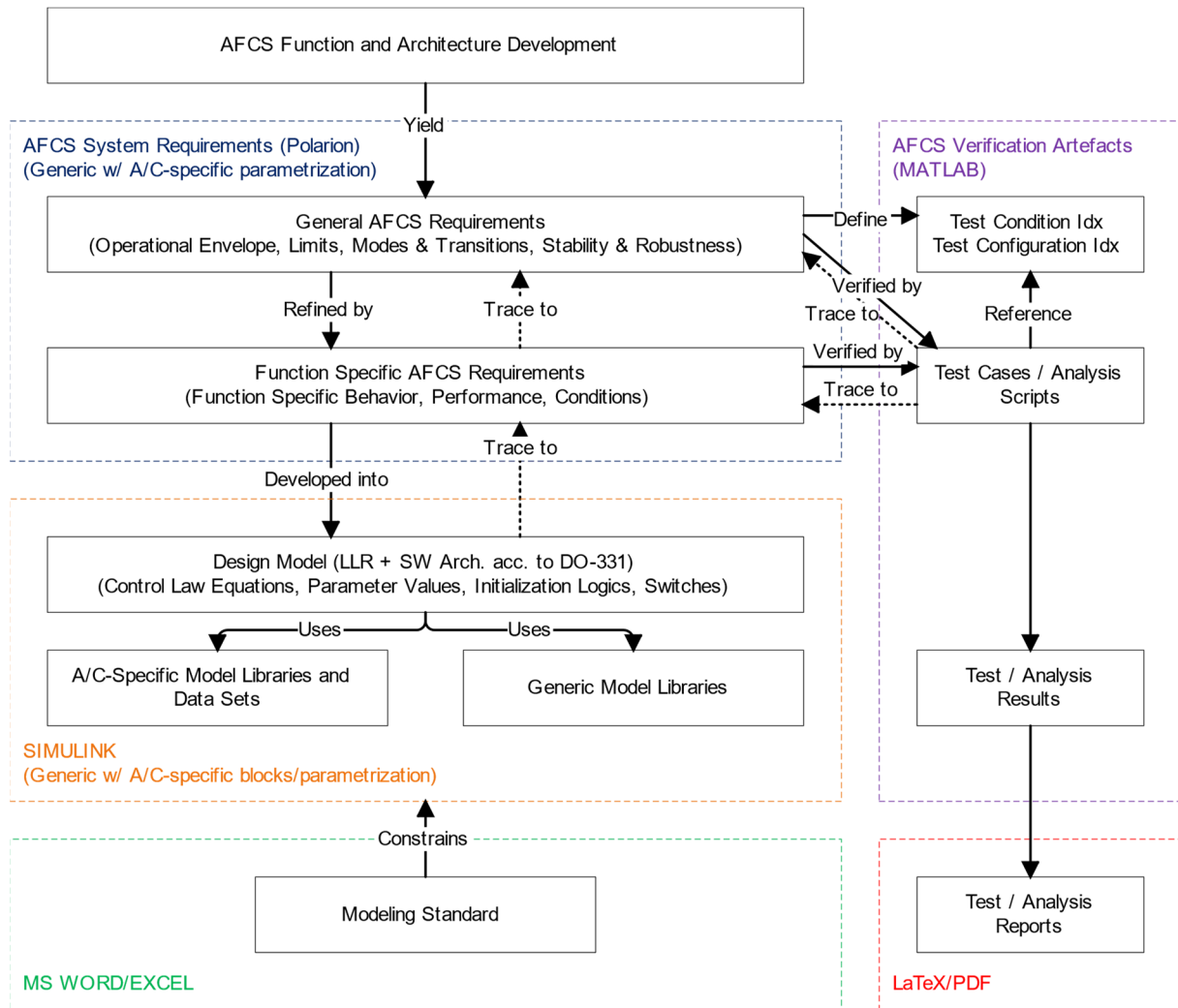


Figure 5.4: Relation between general and function-specific requirements, design model, and verification artifacts.

General AFCF Requirements

The general AFCF system requirements are applicable to all automatic flight control modes and functional elements. Table 5.9 summarizes the application-generic general functional requirements structure with reference to applicable guidance material for desired functional behavior, operational characteristics, and performance.

Table 5.10 and Table 5.11 provide two examples of general performance requirements: general stability margins and mode transition transients, with desired and adequate performance. The example requirements template contains fields for additional requirement information and metadata, such as:

- **Rationale:** i.e., the motivation behind the requirement and description of any assumptions and underlying information that may aid in the validation of the requirement, impact analysis in the case of a requirement update, as well as re-use of the requirement formulation or parameters in new application scenarios
- **Scope of Validity:** the applicability of the requirement in terms of functions, operational envelope, aircraft configurations, and flight phases
- **Means of Compliance:** defines how the requirement shall be verified, under what conditions, and by what means (see means of verification and validation in Section 2.3.5)
- **Requirement Source:** reference to guidance material or other source from which the requirement text or performance parameters are sourced
- **Upstream Trace:** traceability to requirements from which the requirement at hand is developed or provides additional specification of
- **Guidance material:** reference to any guidance material used in the development of the requirement at hand

Figure 5.5 and Figure 5.6 provide an example of the permissible vertical/speed and lateral control mode transitions, with the transitions from and to the automatic flight control modes in selected and managed operation described. The transitions include pilot inputs from the MCP pushbuttons (PB) and toggle buttons, which are further described in Section 5.4.

Table 5.9: List of application-generic general functional requirements.

Function	Requirement Type	Requirement Name	Reference Guidance Material
All Guidance and Control Modes	Functional Behavior	Vertical Plane Control Modes	DO-325 1.5.1 AS94900A 3.3.4.2
		Lateral Plane Control Modes	DO-325 1.5.2 AS94900A 3.3.4.2
		Speed Control Modes	DO-325 1.5.1 DO-325 1.5.10 AS94900A 3.3.4.2
		Managed Operation	DO-325 1.5.4 AS94900A 3.3.4.3
		Selected Operation	DO-325 1.5.8 AS94900A 3.6.9.1 AS94900A 3.7.7.1.9.1
	Operational Characteristics	Vertical Plane/Speed Control Mode Transitions	AS94900A 3.3.1.3.1
		Lateral Plane Control Mode Transitions	AS94900A 3.3.1.3.1
		Mode Selection	DO-325 1.5.8 AS94900A 3.7.7.1.9.1
		Mode Annunciation	AS94900A 3.3.1.3 AS94900A 3.6.9.9
		Target Preselection	DO-325 1.5.8 AS94900A 3.7.7.1.9.1
		Active/Preselected Target Annunciation	AS94900A 3.3.1.3 AS94900A 3.6.9.9
		Actuation Resource Selection	DO-325 2.2.1.1.1 DO-325 2.2.1.1.2 DO-325 2.2.1.6.1 DO-325 2.2.1.6.2
		Actuation Resource Annunciation	DO-325 2.2.1.1.1 DO-325 2.2.1.1.2 DO-325 2.2.1.6.1 DO-325 2.2.1.6.2 AS94900A 3.3.1.3 AS94900A 3.6.9.9
	Performance	Operational Envelope	AS94900A 2.3
		Aircraft Configurations	
		Operation in Atmospheric Disturbances	AS94900A 3.3.1.5
		General Control Loop Stability Margins	AS94900A 3.3.1.4.1
		Input Command Signal Limiting	AS94900A 3.3.4.3.2
		Output Command Signal Limiting	AS94900A 3.3.4.3.2
		Engage-Disengage Transients	AS94900A 3.3.4.3.1.2
		Mode Transition Transients	AS94900A 3.3.4.3.3
		Coordination in Steady Banked Turns	AS94900A 3.3.4.1.3.1
		Lateral Acceleration Limits	AS94900A 3.3.4.1.3.2
	Interface	Managed Operation Interface	AS94900A 3.3.4.3.1
		Selected Operation Interface	AS94900A 3.7.7.1.9.1
		GNSS/INS Interface	AS94900A 3.6.10
		AHRS Interface	AS94900A 3.6.10
IMU Interface		AS94900A 3.6.10	
ADS Interface		AS94900A 3.6.10	

Table 5.10: Example general performance requirement – general stability margins.

<REQ ID>	General Stability Margins	
Category	FUNCTIONAL (PERFORMANCE)	
Description	Automatic flight control modes shall satisfy the following gain and phase margin limits.	
Rationale:	Automatic flight control modes robustness against gain or phase variations due to plant, controller and subsystem parameter and transfer delay uncertainties	
Performance Parameter	Desired Performance	Adequate Performance
Gain margin (GM)	GM = ± 8.0 dB	GM = ± 6.0 dB
Phase margin (PM)	PM = ± 50 degrees	PM = ± 45 degrees
Scope of Validity	Vertical/Lateral/Speed Control Modes Operational envelope and aircraft configurations as defined in TBD	
Means of Compliance	Analysis: <ul style="list-style-type: none"> Linear system gain and phase margin analysis with full inner loop, actuation, and sensor dynamics All feedback paths held at their nominal values except for the path under investigation The loop breaks for analysis made at the actuator commands Test condition index according to TBD Test configuration index according to TBD 	
Requirement Source	Adequate performance from AS94900A 3.3.1.4.1 Aerodynamic-Closed Loop	
Upstream Trace	Vertical Plane Control Modes Lateral Plane Control Modes Speed Control Modes	
Guidance Material	AS94900A	

Table 5.11: Example general performance requirement – mode transition transients.

<REQ ID>	Mode Transition Transients	
Category	FUNCTIONAL (PERFORMANCE)	
Description	Transitions between automatic flight control modes shall not cause transients in normal acceleration or roll attitude greater than specified limits.	
Rationale:	Transition between automatic flight control modes shall be smooth	
Performance Parameter	Desired Performance	Adequate Performance
Body axis normal acceleration at the center of gravity	0 g	± 0.05 g
Bank angle	0 degrees	± 1 degrees
Scope of Validity	Operational envelope and aircraft configurations as defined in TBD	
Means of Compliance	Test: <ul style="list-style-type: none"> Model-in-the-Loop verification of mode transition Test condition index according to TBD Test configuration index according to TBD 	
Requirement Source	Adequate performance from AS94900A 3.3.4.3.3 Switching	
Upstream Trace	Vertical Plane/Speed Control Mode Transitions Lateral Plane Control Mode Transitions	
Guidance Material	AS94900A	

AFCS Lateral Mode Transitions		Lateral Selected				Lateral Managed						
		From	To	HDG	TRK	VOR/LOC*	VOR/LOC	HDG	TRK	VOR/LOC*	VOR/LOC	LNAV
Lateral Selected	HDG		HDG/TRK Toggle = TRK					Lateral FMS ON [HDG desired mode]	Lateral FMS ON [TRK desired mode]			Lateral FMS ON [LNAV desired mode]
	TRK			HDG/TRK Toggle = HDG		Auto TSN		Lateral FMS ON [HDG desired mode]	Lateral FMS ON [TRK desired mode]			Lateral FMS ON [LNAV desired mode]
	VOR/LOC*		HDG/TRK PB [HDG/TRK Toggle = HDG]				Auto TSN	Lateral FMS ON [HDG desired mode]	Lateral FMS ON [TRK desired mode]			Lateral FMS ON [LNAV desired mode]
	VOR/LOC		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]				Lateral FMS ON [HDG desired mode]	Lateral FMS ON [TRK desired mode]			Lateral FMS ON [LNAV desired mode]
Lateral Managed	HDG		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]					TRK desired mode	Auto TSN [APPR FMS ON]		LNAV desired mode
	TRK		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]				HDG desired mode		Auto TSN [APPR FMS ON]		LNAV desired mode
	VOR/LOC*		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]				APPR FMS OFF [HDG desired mode]	APPR FMS OFF [TRK desired mode]		Auto TSN	LNAV desired mode
	VOR/LOC		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]				APPR FMS OFF [HDG desired mode]	APPR FMS OFF [TRK desired mode]			LNAV desired mode
	LNAV		HDG/TRK PB [HDG/TRK Toggle = HDG]	HDG/TRK PB [HDG/TRK Toggle = TRK]				HDG desired mode	TRK desired mode			

Figure 5.6: Lateral control modes transitions, selected and managed.

Function-Specific AFCF Requirements

The characteristics of the individual control modes are captured in the function-specific requirements. The function-specific requirements are structured similarly to the generic requirements, with the definition of desired functional behavior, operational characteristics, and desired/adequate performance.

Desired/adequate performance for target acquisition is defined from desired step response characteristics, such as target overshoot and settling time, see Figure 5.7.

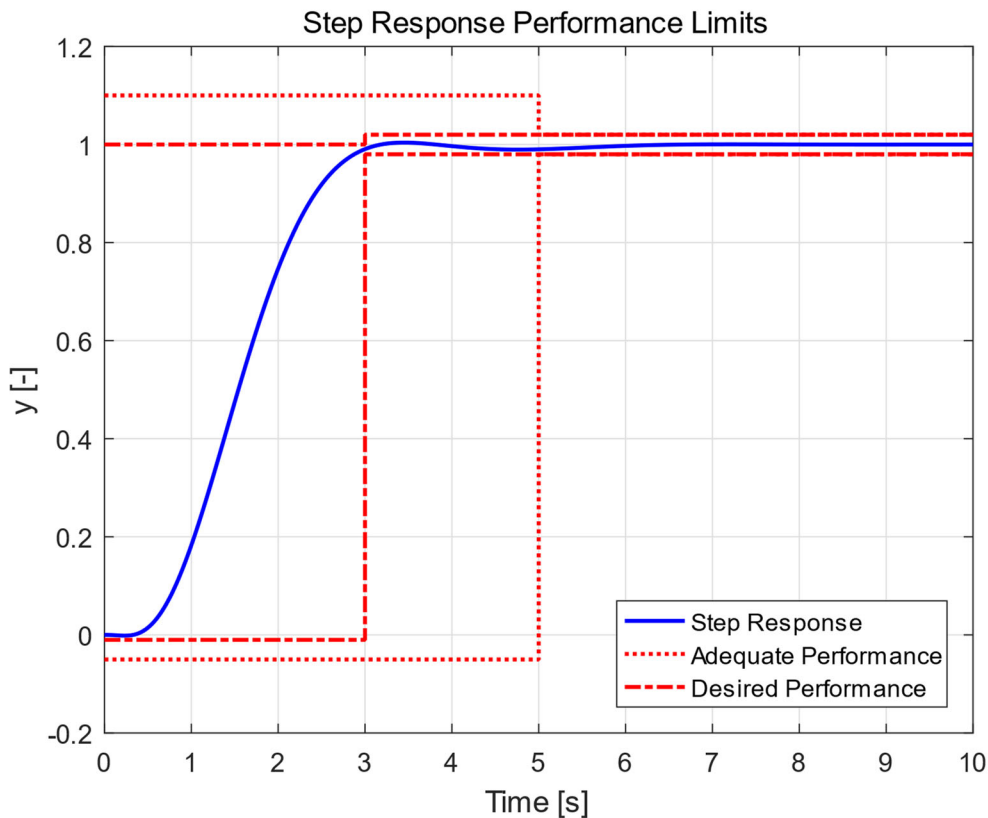


Figure 5.7: Desired and adequate step response performance example.

Desired/adequate performance for disturbance rejection and tracking accuracy in smooth air (static accuracy) as well as in turbulent air are defined with turbulence profiles defined according to Section 3.3.2.

Table 5.12 provides a partial list of application-generic function-specific functional requirements for the FPA, TRK and SPD THR modes, with reference to applicable guidance material for functional behavior, operational characteristics, and adequate performance levels.

Table 5.12: Partial list of application-generic function-specific functional requirements.

Control Mode	Requirement Type	Requirement Name	Reference Guidance Material	
Flight Path Angle (FPA)	Functional Behavior	FPA Mode Desired Behavior	DO-325 1.5.1	
	Operational Characteristics	FPA Mode Activation	AS94900A 3.3.1.3.1	
		FPA Mode Manual Deactivation	AS94900A 3.3.1.3.1	
		FPA Mode Automatic Deactivation	AS94900A 3.3.1.3.1	
		FPA Mode Target Selection	DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
		FPA Mode Target Adjustment	DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
	Performance	FPA Mode Operational Range		
		FPA Mode Overshoot	AS94900A 3.3.4.2.2	
		FPA Mode Settling Time	AS94900A 3.3.4.2.2	
		FPA Mode Disturbance Rejection	AS94900A 3.3.4.2.2	
		FPA Mode Tracking Accuracy in Smooth Air	AS94900A 3.3.4.2.2	
		FPA Mode Tracking Accuracy in Turbulent Air	AS94900A 3.3.4.2.2	
	Track (TRK)	Functional Behavior	TRK Mode Desired Behavior	DO-325 1.5.2
		Operational Characteristics	TRK Mode Acquire Direction	AS94900A 3.3.4.2.4
TRK Mode Activation			AS94900A 3.3.1.3.1	
TRK Mode Manual Deactivation			AS94900A 3.3.1.3.1	
TRK Mode Automatic Deactivation			AS94900A 3.3.1.3.1	
TRK Mode Target Selection			DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
Performance		TRK Mode Target Adjustment	DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
		TRK Mode Overshoot	AS94900A 3.3.4.2.4	
		TRK Mode Turn Rate Limitation	AS94900A 3.3.4.2.4	
		TRK Mode Disturbance Rejection	AS94900A 3.3.4.2.4	
		TRK Mode Tracking Accuracy in Smooth Air	AS94900A 3.3.4.2.3	
		TRK Mode Tracking Accuracy in Turbulent Air	AS94900A 3.3.4.2.3	
Speed by Thrust (SPD THR)		Functional Behavior	SPD THR Mode Desired Behavior	DO-325 1.5.10
		Operational Characteristics	SPD THR Mode Activation	AS94900A 3.3.1.3.1
	SPD THR Mode Manual Deactivation		AS94900A 3.3.1.3.1	
	SPD THR Mode Automatic Deactivation		AS94900A 3.3.1.3.1	
	SPD THR Mode Target Selection		DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
	SPD THR Mode Target Adjustment		DO-325 1.5.8 AS94900A 3.7.7.1.9.1	
	Performance	SPD THR Mode Operational Range		
		SPD THR Mode Overshoot	AS94900A 3.3.4.2.6	
		SPD THR Mode Settling Time	AS94900A 3.3.4.2.6	
		SPD THR Mode Disturbance Rejection	AS94900A 3.3.4.2.6	
		SPD THR Mode Tracking Accuracy in Smooth Air	AS94900A 3.3.4.2.6	
		SPD THR Mode Tracking Accuracy in Turbulent Air	AS94900A 3.3.4.2.6	

Derived Inner Loop Performance Requirements

The function-specific performance requirements consider the full control path or speed loop performance, including the closed inner loop dynamics, actuation dynamics, and sensor dynamics such as delays and filtering. In order for the path control loops to be able to satisfy the stated performance requirements, performance requirements for the inner loop have to be derived that allow for desired and adequate path loop performance.

The required inner loop performance is dependent on the path loop control structure, and thus on the controller design process. Since these inner loop performance requirements are not uniquely related to a high-level system requirement but result from the design process itself, they are considered *derived requirements* according to the ARP4754B requirements process [37], see Section 2.2.1.

Derived performance requirements for the inner loops comprise frequency and damping envelopes for desired and adequate path control performance, given reference inner loop transfer functions for the incremental specific forces, (see Eq. (3.85)-(3.86)), given by

$$\Delta f_{z,K} = \frac{-(T_{z,SP} \cdot s - 1) \cdot \omega_{0,SP}^2}{s^2 + 2 \cdot \zeta_{SP} \cdot \omega_{0,SP} + \omega_{0,SP}^2} \cdot \Delta f_{z,K,C}, \quad (5.1)$$

$$\Delta f_{y,K} = \frac{\omega_{0,\mu}^2}{s^2 + 2 \cdot \zeta_{\mu} \cdot \omega_{0,\mu} + \omega_{0,\mu}^2} \cdot \Delta f_{y,K,C}. \quad (5.2)$$

The incremental specific dynamics in the vertical plane includes a right half plane zero to model the non-minimum phase behavior of the closed inner loop dynamics.

The performance requirements are generated using a MATLAB model structure visualized in Figure 5.9, consisting of:

- Linear models of the AFCS control elements, such as reference models, error controllers, inversions, and input signal filters
- The inner loop transfer functions
- A simplified plant dynamics model with coupled speed and path dynamics

The required inner loop frequency and damping are generated for each analyzed point in the operational envelope (grid density may vary) analyzing the path loop step response characteristics against the stated requirements for a set of frequency and damping combinations (the density of the frequency/damping grid may be varied). This results in permissible inner loop performance envelopes for achievable path loop desired and adequate performance.

An example result is illustrated in Figure 5.8, for a speed of $V_A = 50$ m/s, flight path angle loop second-order reference model with time constants $T_1 = 1.5$ s and $T_2 = 0.1$ s, no sensor measurement delays or signal filtering, and inner loop right half plane zero of $T_z = 0.1$ s. Inner loop frequency is varied between 1.5 and 4 rad/s and damping between 0.4 and 0.8.

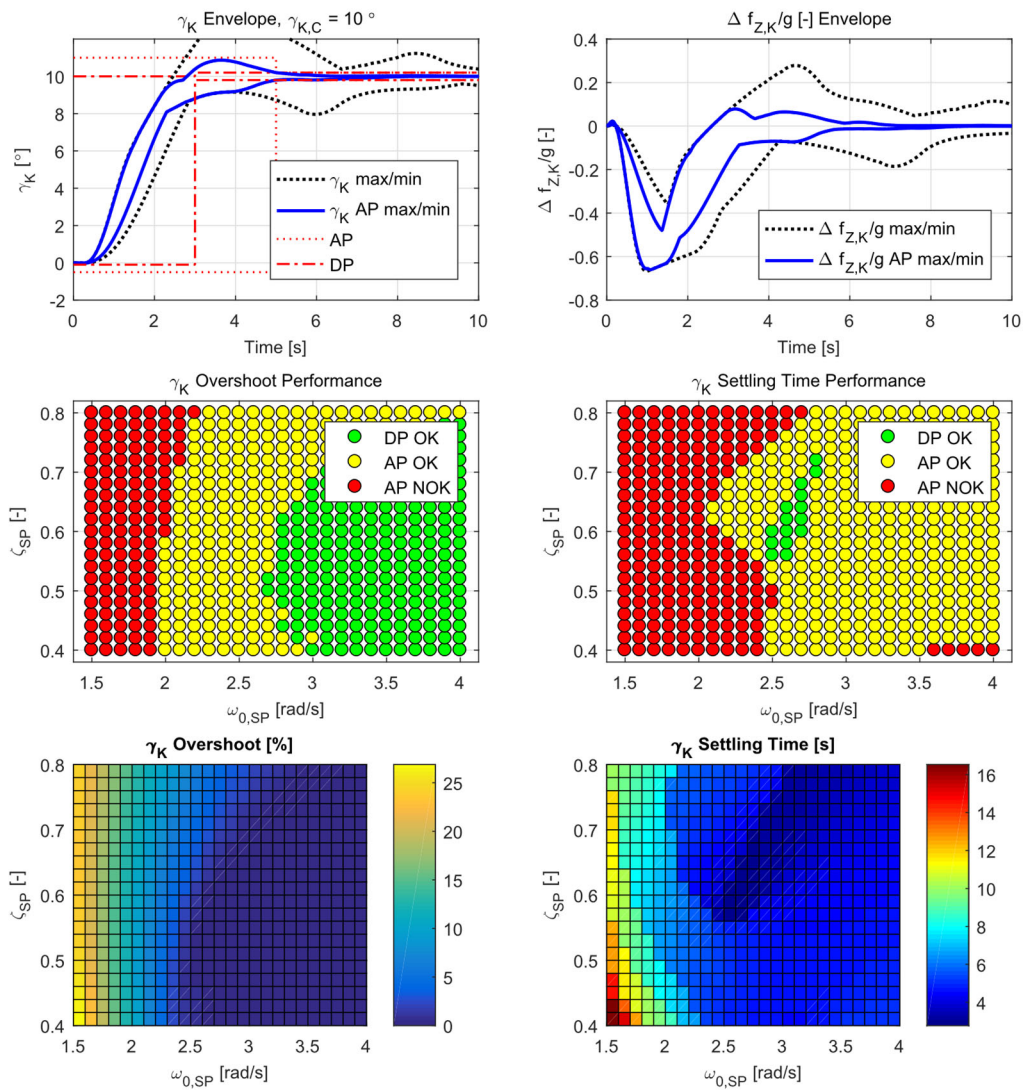


Figure 5.8: Derivation of inner loop performance envelopes for desired and adequate path loop control performance.

The upper left figure shows the envelope curves for the flight path angle step response for all evaluated inner loop frequency and damping combinations (black dotted lines), the envelope curves for the subset of the flight path angle responses that satisfy the adequate performance criteria (blue solid lines), as well as the adequate and desired performance limits. The upper right shows the corresponding incremental specific force envelope curves: for all inner loop combinations (black dotted lines) and for the subset satisfying flight path angle adequate performance (blue solid lines).

The second row shows the evaluated inner loop frequency and damping combinations resulting in a desired, adequate, or not adequate flight path angle overshoot (left figure) and settling time (right figure). The third row shows the corresponding flight path angle overshoot (left figure) and settling time (right figure).

5.2 Controller Design

This section presents the architecture and design of the automatic flight control functional algorithms, i.e., the control laws and logic designed to implement the application-generic AFCF to meet the functional requirements discussed in Section 5.1.3. The design is based on the flight dynamics and control theory and constraints presented in Chapters 3 and 4. Parts of the work related to the flight path controller design and the control objective prioritization strategy have previously been published in [71, 72, 73, 74].

5.2.1 Functional Architecture Preliminary Considerations

The overall architectural principle for the modular FGCS illustrated in Figure 1.1 (see Section 1.1), is a modular design with encapsulated functionalities and configuration-controlled interfaces that allow for the necessary concurrent development of the AFCS, inner loop, trajectory controller, and higher level system automation functionalities. The objective is to maintain identical elements between different platforms in order to avoid significant adjustments, rework, and retesting. The degree of commonality between aircraft applications is higher for the higher-level functionalities such as the system automation, trajectory controller, and AFCS, whereas the functional design of the inner loop and thrust control modules, selected as well as automatic trim functionalities, have a more aircraft-specific functional design.

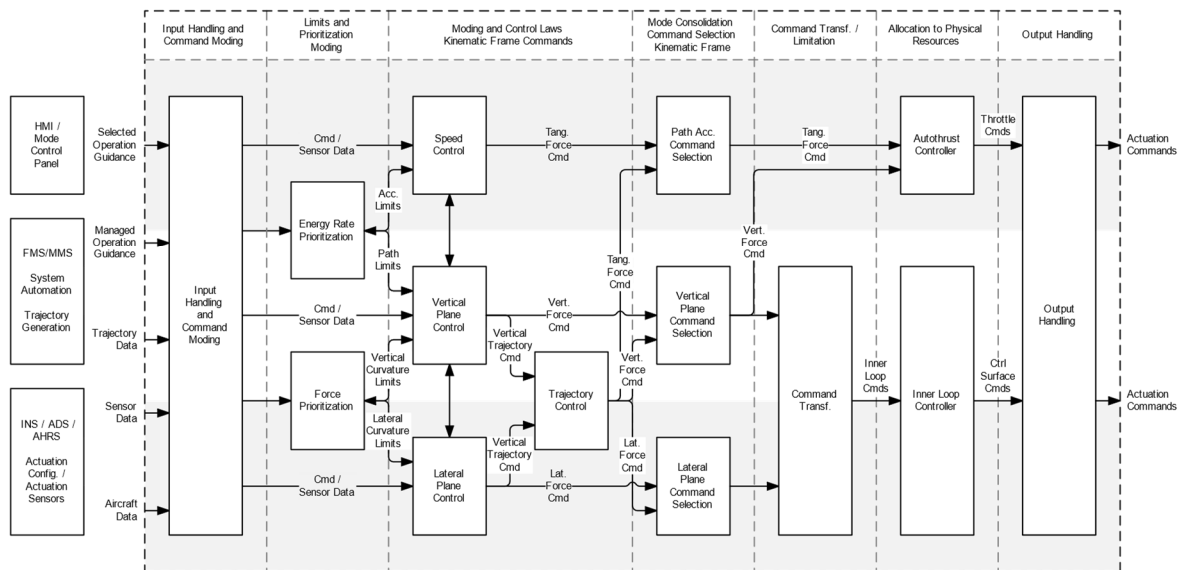


Figure 5.10: Principal AFCS functional architecture elements.

The modularity principle extends to the AFCS architecture, illustrated in Figure 5.10. The objectives here are:

- Contain functionalities into modules of limited complexity for simplified unit testing and configuration management

- Allow reuse of common functional elements and structures by enabling model references and library blocks
- Separate generic and aircraft-specific functionalities

The AFCS has a distributed limit/protection and control law mode control logic as well as encapsulated control law elements for the speed, vertical, and lateral planes that enables configuration and adaptation of functionalities for new application platforms without extensive retesting of the basic functionalities. Controller gains and parameters such as limits and filter constants will always be designed according to the application aircraft and the available performance of the inner loop.

The AFCS module is sectioned into functional blocks for input and output handling, the control modes for the different control planes, the cross-plane functional couplings and limitation calculations, as well as transformation and limitation of commands to the inner loop and autothrust controllers, see Figure 5.10. This section gives a brief overview of the functional structure, and thus the functional environment of the control laws detailed in subsequent sections.

Interfaces External to the AFCS

The AFCS receives pushbutton actions and preselected/confirmed targets from a mode control panel interface for selected operation, and from an external guidance interface for managed operation, see description of the guidance modes in Section 5.1.2. In selected operation, the AFCS operates similarly to traditional autopilots, with the pushbutton actions transformed into desired modes according to specified transition criteria. The AFCS receives the individual pushbuttons and targets and performs the main part of the mode control internally. The mode control panel is further described in Section 5.4.3.

The external guidance interface allows a higher-level system automation to request modes and targets, as well as desired trajectory and deviation information. The external guidance interface allows for dynamic limits and prioritizations depending on, e.g., flight phase.

Further inputs are inertial, air data, and radio navigation sensor data, as well as information about aircraft configuration (gear, flaps settings) and actuation deflections.

Distributed Mode Control Logic

A centralized mode transition logic in the form of a single state machine with a large number of states and transition conditions quickly becomes complex and requires extensive verification. For any application-specific modifications or addition/removal of control modes depending on operational scenario would require retesting of the entire logic. The AFCS module instead utilizes a distributed mode control logic architecture, principally illustrated in Figure 5.11. The distributed approach involves three “stages” of mode control, reducing complexity at each stage and the possibility to add or remove function-specific mode control without requiring a retesting of the entire mode control logic.

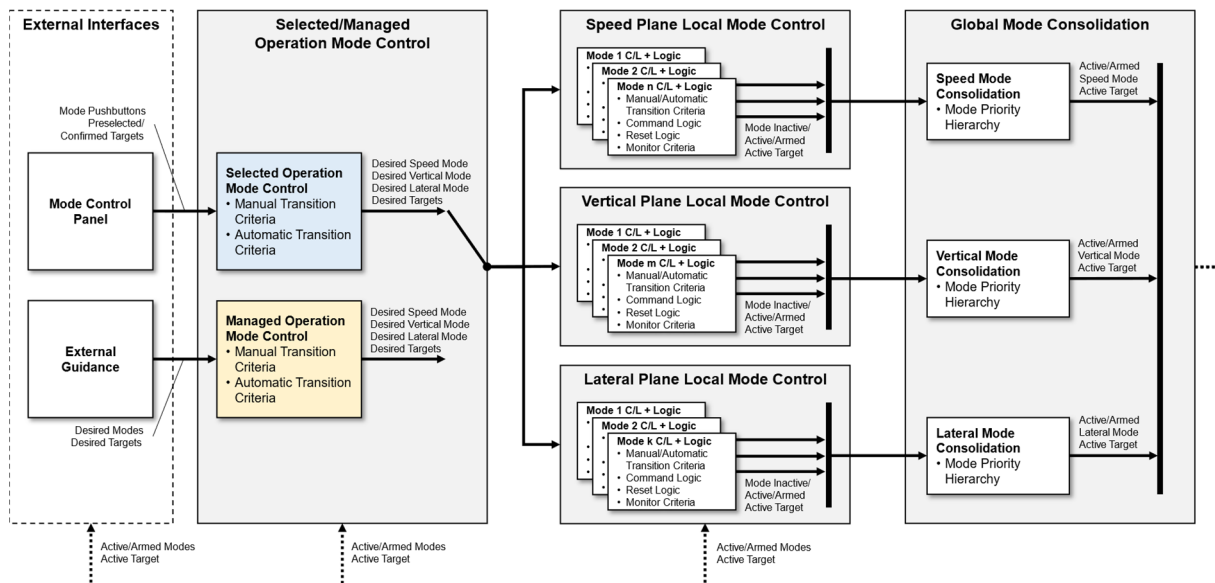


Figure 5.11: Principal overview of the AFCS distributed mode control logic.

The three stages are described in the following.

- Selected/managed operation mode control
 - Selected operation mode control based on MCP pushbutton and selector knob inputs
 - Managed operation mode control based on desired modes and targets from an external guidance system
 - Outputs a consolidated command bus with *desired* modes and targets
 - Downstream mode control and control laws are independent of selected or managed operation
- Speed/vertical/lateral plane mode control
 - Local function-specific mode control for each control mode
 - Determines the state of each function based on manual and automatic transition criteria
 - Outputs each function state as *inactive*, *active*, or *armed*
 - Function-specific control law integrator reset and initialization logic
 - Function-specific mode-internal monitoring criteria may deactivate function and transition it to an inactive state
 - Allows for addition or removal of individual functional blocks, or adjustment of function-specific logic or monitor criteria without extensive retesting
- Global mode consolidation and output command selection
 - Consolidation of the speed, vertical, and lateral mode states
 - Outputs the *active* and *armed* speed, vertical, and lateral mode
 - Mode priority hierarchy where the higher priority active mode and command is forwarded to the command transformation and inner loop

- Feedback of active/armed modes to MCP and external guidance for annunciation/mode awareness
- Fail-safe behavior where deactivation of all modes results in default outputs being incremental specific forces in the kinematic frame equal to zero, i.e., no path curvature command

Input Handling and Command Mode Control

The input handling contains the selected/managed operation mode control described above, with the output being a consolidated internal command bus with desired modes and targets.

The input handling also contains sensor filtering and mapping of sensor signals onto an AFCS internal sensor bus utilized by the individual control modes, with required sensor data and sensor signal status information.

Limits and Protection Mode Control

This block contains the energy-rate and force prioritization as well as energy protection mode control and calculation of limits, as inputs to the flight path and trajectory control modes downstream. The energy-rate prioritization and energy protection module handles the cross-coupling of the speed and vertical path control, described in Section 5.2.8, and the force prioritization module handles the cross-coupling of the vertical and lateral flight path control, described in Section 5.2.9.

Control of the Flight Path – Kinematic Frame Commands

The control laws for the path and trajectory following are based on the kinematic frame, independent of the aircraft attitude. The speed controlled is the (indicated) aerodynamic speed along the flight path. Each speed, vertical, and lateral control mode block encapsulate the control laws and local control law mode control, such as limitations and integrator initialization and reset logics.

The path control architecture is described in Section 5.2.2, with details for the vertical, lateral, and speed control loops described in Sections 5.2.3-5.2.5, respectively. The embedded trajectory controller and how it is utilized to implement AFCS trajectory modes such as the altitude hold mode are described in Section 5.2.6. The radio navigation and approach modes are briefly described in Section 5.2.7.

Command Transformation and Allocation to Physical Resources – Body Axis Commands

Curvature of the flight path is ultimately achieved by primarily changing the magnitude and the direction of the lift vector, through the faster, attitude-dependent rotational dynamics of the aircraft via control surface deflections. The desired path curvature commands in the kinematic frame are translated into bank angle and specific force commands in the body-axis frame as inputs to the inner loop controller. The inner loop controller transforms the body-axis bank angle and specific force commands into the required control surface deflections. The inner loop command transformation is described in Section 5.2.10.

The desired path acceleration is achieved the changing the thrust setting of the propulsion system. The autothrust control loop is described in Section 5.2.11.

Output Handling

The controller calculates the desired control surface deflections, maps those, and outputs the corresponding commands to the actuation system, i.e., desired aileron, elevator, and rudder, as well as throttle actuator positions.

Various information on the internal status of the controller could be of interest for real-time monitoring or post-test analysis, e.g., saturation status of limiters, and activation/deactivation of protection and prioritization modes.

5.2.2 Path Control Architecture

The basic control principle for the control of the vertical and lateral flight path modes, i.e. vertical speed/flight path angle and heading/track angle, as well as the airspeed along the flight path, is a reference model-based dynamic inversion of the point mass equations of motion, Eq. (3.57)-(3.59), with pseudo-control hedging to account for the inner loop and propulsion dynamics not considered in the inversion [71, 73].

The basic flight path control structure with reference models and error controllers is pictured in Figure 5.12.

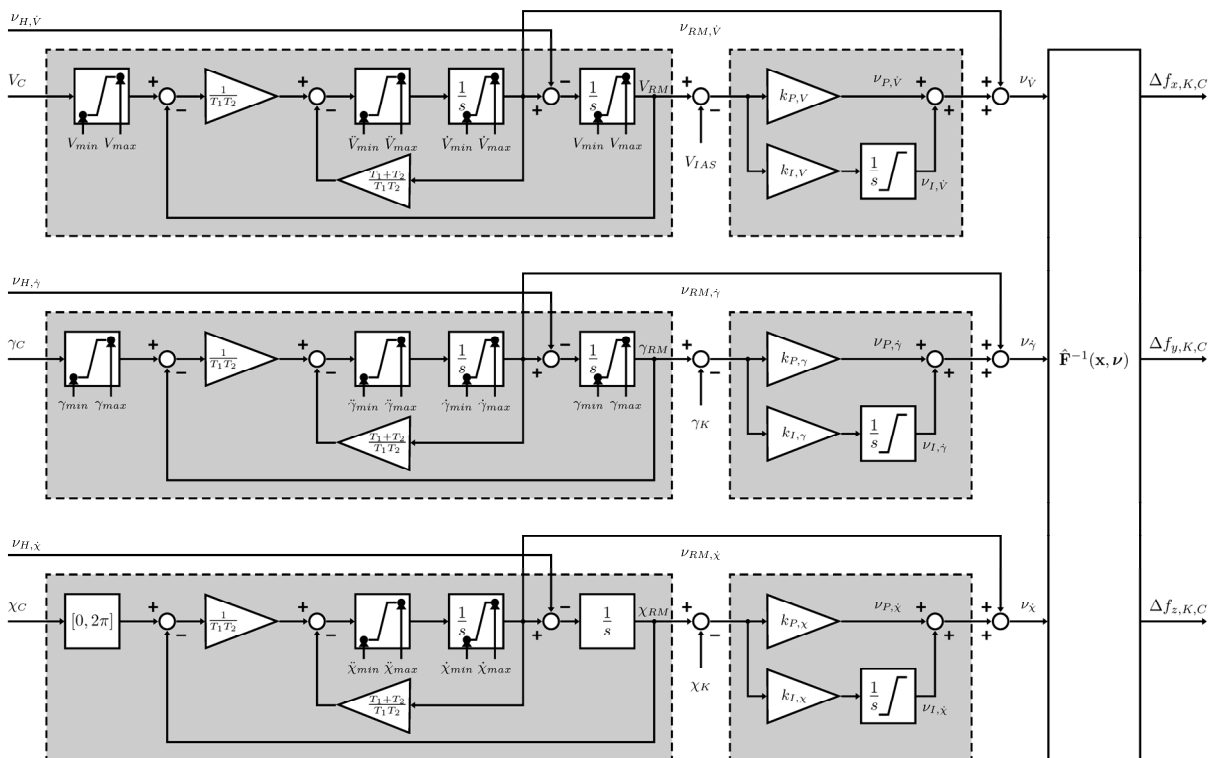


Figure 5.12: Basic flight path control structure with reference models and error controllers.

Inversion of the Path Dynamics

The path dynamics and thus the system subject to the inversion is given by the Eq.(3.57)-(3.59), with the state vector \mathbf{x} being the kinematic velocity magnitude V_K , the course angle χ_K and the flight path angle γ_K , according to Eq. (3.16),

$$\mathbf{x} = (\bar{\mathbf{V}}_K^G)_O^E = \begin{bmatrix} V_K \\ \chi_K \\ \gamma_K \end{bmatrix}_O^E \quad (5.3)$$

and the incremental specific forces $\Delta f_{x,K}$, $\Delta f_{y,K}$, and $\Delta f_{z,K}$ as the desired plant inputs, giving the system

$$\underbrace{\begin{bmatrix} \dot{V}_K \\ \dot{\chi}_K \\ \dot{\gamma}_K \end{bmatrix}}_{\dot{\mathbf{x}}}^{EO} = \underbrace{\mathbf{0}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{V_K \cos \gamma_K} & 0 \\ 0 & 0 & -\frac{1}{V_K} \end{bmatrix}}_{\mathbf{G}(\mathbf{x})=[\mathbf{g}_1(\mathbf{x}) \quad \mathbf{g}_2(\mathbf{x}) \quad \mathbf{g}_3(\mathbf{x})]} \underbrace{\begin{bmatrix} \Delta f_{x,K} \\ \Delta f_{y,K} \\ \Delta f_{z,K} \end{bmatrix}}_{\mathbf{u}}, \quad (5.4)$$

$$\underbrace{\begin{bmatrix} V_K \\ \chi_K \\ \gamma_K \end{bmatrix}}_{\mathbf{y}}^E = \underbrace{\begin{bmatrix} V_K \\ \chi_K \\ \gamma_K \end{bmatrix}}_{\mathbf{h}(\mathbf{x})}.$$

In Eq. (5.4) it can be directly seen that it takes one differentiation of each output for an input to appear, i.e., the path dynamics is of relative degree one. Thus, an inversion architecture corresponding to a relative degree of one is desired.

The pseudo-control vector represents the desired acceleration along the flight path and the desired curvature of the vertical or lateral flight path according to

$$\mathbf{v} = \begin{bmatrix} v_{\dot{V}_K} \\ v_{\dot{\chi}_K} \\ v_{\dot{\gamma}_K} \end{bmatrix}. \quad (5.5)$$

With $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, the decoupling matrix $\hat{\mathbf{A}}(\mathbf{x})$ according to Eq. (4.10) is given by

$$\hat{\mathbf{A}}(\mathbf{x}) = \begin{bmatrix} \left(\frac{\partial h_1}{\partial \mathbf{x}}\right)^T \mathbf{g}_1(\mathbf{x}) & \left(\frac{\partial h_1}{\partial \mathbf{x}}\right)^T \mathbf{g}_2(\mathbf{x}) & \left(\frac{\partial h_1}{\partial \mathbf{x}}\right)^T \mathbf{g}_3(\mathbf{x}) \\ \left(\frac{\partial h_2}{\partial \mathbf{x}}\right)^T \mathbf{g}_1(\mathbf{x}) & \left(\frac{\partial h_2}{\partial \mathbf{x}}\right)^T \mathbf{g}_2(\mathbf{x}) & \left(\frac{\partial h_2}{\partial \mathbf{x}}\right)^T \mathbf{g}_3(\mathbf{x}) \\ \left(\frac{\partial h_3}{\partial \mathbf{x}}\right)^T \mathbf{g}_1(\mathbf{x}) & \left(\frac{\partial h_3}{\partial \mathbf{x}}\right)^T \mathbf{g}_2(\mathbf{x}) & \left(\frac{\partial h_3}{\partial \mathbf{x}}\right)^T \mathbf{g}_3(\mathbf{x}) \end{bmatrix} \quad (5.6)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{V_K \cos \gamma_K} & 0 \\ 0 & 0 & -\frac{1}{V_K} \end{bmatrix}.$$

and $\hat{\mathbf{b}}(\mathbf{x}) = \mathbf{0}$, the control law for the path loops becomes

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \Delta f_{x,K,C} \\ \Delta f_{y,K,C} \\ \Delta f_{z,K,C} \end{bmatrix}}_{\mathbf{u}_C} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & V_K \cos \gamma_K & 0 \\ 0 & 0 & -V_K \end{bmatrix}}_{\hat{\mathbf{A}}^{-1}} \underbrace{\begin{bmatrix} v_{\dot{V}_K} \\ v_{\dot{\chi}_K} \\ v_{\dot{\gamma}_K} \end{bmatrix}}_{\mathbf{v}} - \underbrace{\mathbf{0}}_{\hat{\mathbf{b}}(\mathbf{x})} \\
 &= \begin{bmatrix} v_{\dot{V}_K} \\ V_K \cos \gamma_K v_{\dot{\chi}_K} \\ -V_K v_{\dot{\gamma}_K} \end{bmatrix}.
 \end{aligned} \tag{5.7}$$

Reference Models for the Path Dynamics

Commands for the path dynamics (V_C , $\chi_{K,C}$, $\gamma_{K,C}$) are passed through reference models to create continuous and transient-free trajectories for the pseudo controls and thus for the commanded incremental specific forces $\Delta f_{x,K,C}$, $\Delta f_{y,K,C}$, and $\Delta f_{z,K,C}$. As the path dynamics is of relative degree one with respect to the incremental specific forces, a first-order reference model would suffice to generate the required pseudo control, according to

$$\dot{y}_{RM} = \frac{1}{T_{RM}} (y_C - y_{RM}) - v_H = -\frac{1}{T_{RM}} y_{RM} + \frac{1}{T_{RM}} y_C - v_H, \tag{5.8}$$

$$v_{RM} = \frac{1}{T_{RM}} (y_C - y_{RM}) = -\frac{1}{T_{RM}} y_{RM} + \frac{1}{T_{RM}} y_C, \tag{5.9}$$

or on state space matrix/vector form

$$\underbrace{[\dot{y}_{RM}]}_{\mathbf{A}_{RM}} = \underbrace{\begin{bmatrix} -\frac{1}{T_{RM}} \end{bmatrix}}_{\mathbf{A}_{RM}} \underbrace{[y_{RM}]}_{\mathbf{y}_{RM}} + \underbrace{\begin{bmatrix} \frac{1}{T_{RM}} & -1 \end{bmatrix}}_{\mathbf{B}_{RM}} \underbrace{\begin{bmatrix} y_C \\ v_H \end{bmatrix}}_{\mathbf{u}_{RM}}, \tag{5.10}$$

$$\underbrace{\begin{bmatrix} y_{RM} \\ v_{RM} \end{bmatrix}}_{\mathbf{C}_{RM}} = \underbrace{\begin{bmatrix} 1 \\ -\frac{1}{T_{RM}} \end{bmatrix}}_{\mathbf{C}_{RM}} \underbrace{[y_{RM}]}_{\mathbf{y}_{RM}} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{1}{T_{RM}} & 0 \end{bmatrix}}_{\mathbf{D}_{RM}} \underbrace{\begin{bmatrix} y_C \\ v_H \end{bmatrix}}_{\mathbf{u}_{RM}}. \tag{5.11}$$

where T_{RM} is the time constant of the reference dynamics, y_C is the flight path command, y_{RM} the reference model state, v_{RM} the pseudo control to the inversion and v_H the hedging signal added to adjust the reference model dynamic according to the plant response.

However, in order to also smoothen the reference state derivatives, aperiodic second-order reference models are applied, according to

$$\ddot{y}_{RM} = \frac{1}{T_{RM,1} T_{RM,2}} (y_C - y_{RM}) - \frac{T_{RM,1} + T_{RM,2}}{T_{RM,1} T_{RM,2}} v_{RM}, \tag{5.12}$$

$$\dot{y}_{RM} = v_{RM} - v_H \tag{5.13}$$

or on matrix/vector form

$$\begin{bmatrix} \ddot{y}_{RM} \\ \dot{y}_{RM} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{T_{RM,1} + T_{RM,2}}{T_{RM,1}T_{RM,2}} & -\frac{1}{T_{RM,1}T_{RM,2}} \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}_{RM}} \begin{bmatrix} \dot{y}_{RM} \\ y_{RM} \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & 0 \\ \frac{1}{T_{RM,1}T_{RM,2}} & -1 \end{bmatrix}}_{\mathbf{B}_{RM}} \begin{bmatrix} y_C \\ v_H \end{bmatrix}, \quad (5.14)$$

$$\begin{bmatrix} y_{RM} \\ v_{RM} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{C}_{RM}} \begin{bmatrix} \dot{y}_{RM} \\ y_{RM} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{D}_{RM}} \begin{bmatrix} y_C \\ v_H \end{bmatrix}. \quad (5.15)$$

where $T_{RM,1}$ and $T_{RM,2}$ are the time constants of the second-order reference dynamics.

The characteristics of the path loop dynamics can be regarded as the characteristics of an inner loop from the perspective of the next outer loop. In general, this means that a fast response to commands is desired, as well as an adequate *phase reserve* provided to the next outer loop. The phase reserve here considers the phase curve of the closed path loop and its provisions for the loop design of the next outer loop and not the phase margin (and thus robustness) of the opened path loop. The path control structure here is primarily intended to process direct and stepwise path loop commands from the pilot via the MCP or from higher-level automation such as flight or mission management systems, and not primarily as an inner loop for cascaded trajectory control loops. A parallel trajectory controller is implemented as part of the modular FGCS, see Figure 1.1, and description in Section 1.1. As trajectory control is provided by a separate controller and not as cascaded loops around the path controller, the trajectory control bandwidth is not impacted by the added path control dynamics.

The time constants $T_{RM,1}$ and $T_{RM,2}$ may be chosen such that a slower pole dominates the reference dynamics for a desired frequency spectrum, and a faster pole is added to smoothen the pseudo-control. A comparison of first- and second-order reference model dynamics in terms of step responses, Bode gain and phase plots for the transfer functions from path command to path state and pseudo-control, respectively, as well as closed loop pole/zero maps is given in Figure 5.13, with a first-order reference model time constant $T_{RM} = 1.5$, second-order reference model time constants $T_1 = 1.5$ s and $T_2 = 0.1$ s. The closed inner loop dynamics is modeled according to Eq. (5.1) with natural frequency $\omega_{0,SP} = 2$ rad/s, relative damping $\zeta_{SP} = 0.7$ and zero $T_z = 0.1$. The effect on the path loop step response overshoot and settling time is small, and the desired smooth pseudo-control command is achieved. The closed path loop phase reserve is somewhat reduced for the second-order reference model, for larger frequencies, as expected due to the added dynamics.

The reference models produce consistent and smooth trajectories for the reference states and their command variables and incorporate limitation of the reference states and their derivatives according to aircraft configuration and desired operational limitations. The limiters allow for cross-feeds of limit values between the reference models for control objective prioritization and further protections and limitations, see Sections 5.2.8 and 5.2.9.

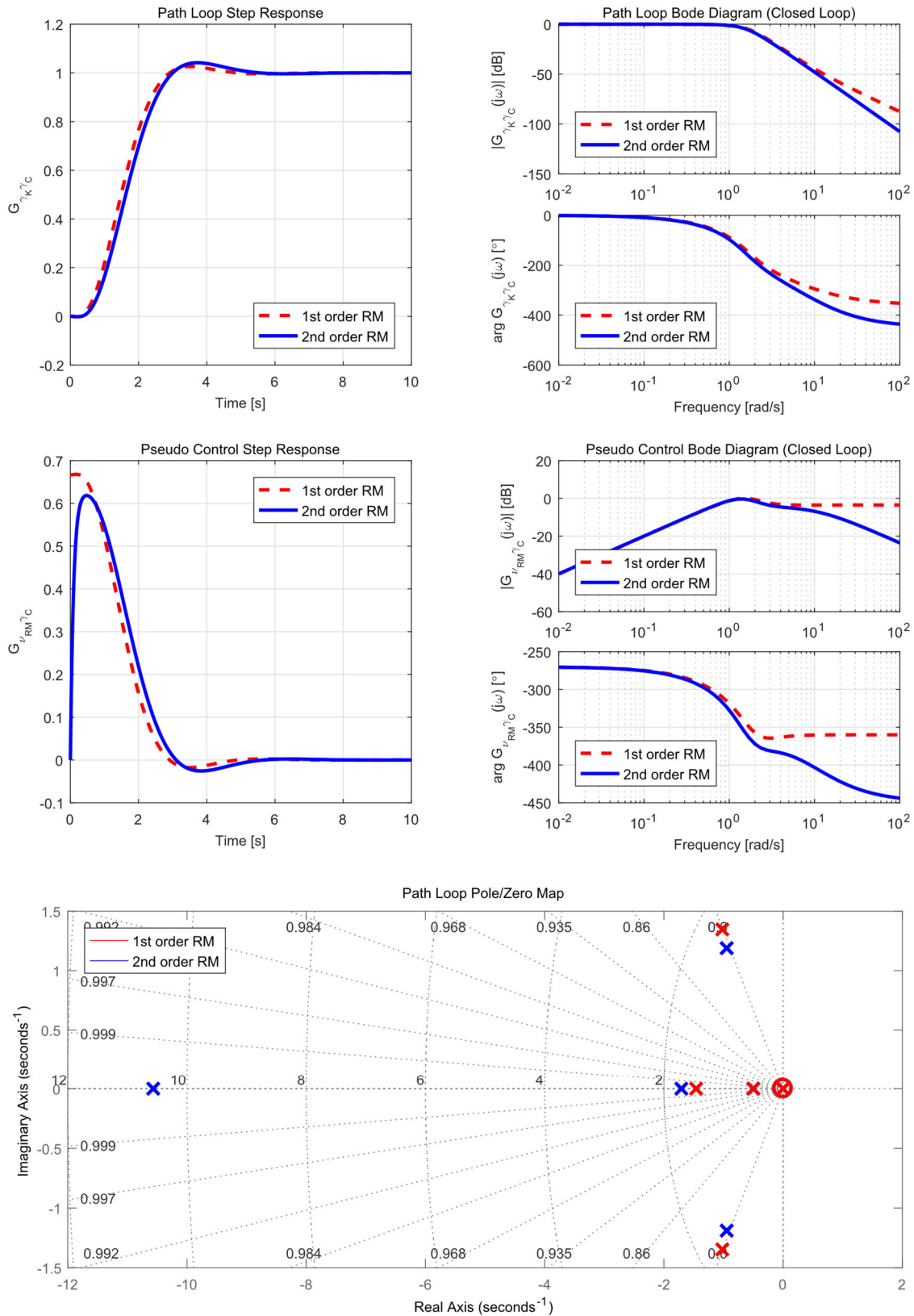


Figure 5.13: First- and second-order reference model comparison, with command to output state and pseudo-control step responses and closed loop Bode diagrams, respectively.

The design of the time constants is based on linear models of the closed inner loop dynamics with desired step response overshoot requirements as primary design criteria and adequate closed-loop stability as monitor criteria, see Section 5.3.3.

Plant Response Estimation and Pseudo-Control Hedging

The estimated plant response is the estimated specific forces in the kinematic frame and is calculated from the specific forces in the body axis frame. From the measured specific forces including gravity in the body axis frame, the specific force vector including gravity in the kinematic frame is calculated according to

$$\begin{bmatrix} f_x \\ f_y \\ f_x \end{bmatrix}_{K,meas} = \mathbf{M}_{KB} \begin{bmatrix} f_x \\ f_y \\ f_x \end{bmatrix}_{B,meas} = \mathbf{M}_{KO} \mathbf{M}_{OB} \begin{bmatrix} f_x \\ f_y \\ f_x \end{bmatrix}_{B,meas}, \quad (5.16)$$

with $\mathbf{M}_{KO}(\gamma_{K,meas}, \chi_{K,meas})$ and $\mathbf{M}_{OB}(\Psi_{meas}, \Theta_{meas}, \Phi_{meas})$ being the transformation matrices from the O into the K frame, and from the B into the O frame, respectively, see Appendix B.

With the incremental specific forces defined as

$$\Delta f_{x,K,meas} = f_{x,K,meas} - g \sin \gamma_{K,meas}, \quad (5.17)$$

$$\Delta f_{y,K,meas} = f_{y,K,meas}, \quad (5.18)$$

$$\Delta f_{z,K,meas} = f_{z,K,meas} + g \cos \gamma_{K,meas}, \quad (5.19)$$

the plant response is subsequently estimated using the equations Eq. (3.57)-(3.59),

$$\hat{v}_{\dot{V}_K} = \Delta f_{x,K,meas}, \quad (5.20)$$

$$\hat{v}_{\dot{\chi}_K} = \frac{1}{V_{K,meas} \cos \gamma_{K,meas}} \Delta f_{y,K,meas} \quad (5.21)$$

$$\hat{v}_{\dot{\gamma}_K} = \frac{1}{V_{K,meas}} \Delta f_{z,K,meas}. \quad (5.22)$$

The estimated plant responses are further low-pass filtered to reduce the impact of specific force measurement and turbulence noise on the command signal to the inner loops, with second-order low-pass filters according to

$$\hat{v}_{filt} = \frac{\omega_{0,\hat{v}}^2}{s^2 + 2 \cdot \zeta_{\hat{v}} \cdot \omega_{0,\hat{v}} + \omega_{0,\hat{v}}^2} \cdot \hat{v}. \quad (5.23)$$

The hedging signals $v_H = v - \hat{v}$, i.e. the difference between total pseudo control v and filtered estimated plant response \hat{v}_{filt} , are fed back to the reference models and added to the pseudo control before the reference state integrator to slow down the reference state and thus the reference model output. The pseudo-control hedging ensures that the reference model state is adjusted to the estimated response of the plant.

Path Loop Error Controllers

To stabilize the error between the reference model and the measured flight path state,

$$e = y_{RM} - y_{meas}, \quad (5.24)$$

PI feedback controllers are included, and a proportional part v_P and an integral part v_I are added to the pseudo-control from the reference model, before entering the inversion, according to

$$\begin{aligned} v &= v_{RM} + v_P + v_I \\ &= v_{RM} + k_P \cdot e + \int k_I \cdot e \, dt, \end{aligned} \quad (5.25)$$

or on matrix/vector form

$$\begin{bmatrix} \dot{v}_I \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{\mathbf{A}_{EC}} [v_I] + \underbrace{\begin{bmatrix} k_I \end{bmatrix}}_{\mathbf{B}_{EC}} [e], \quad (5.26)$$

$$\begin{bmatrix} v_P \\ v_I \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{C}_{EC}} [v_I] + \underbrace{\begin{bmatrix} k_P \\ 0 \end{bmatrix}}_{\mathbf{D}_{EC}} [e]. \quad (5.27)$$

The design of the error dynamics is decoupled from the design of the reference dynamics, see Section 4.3, and the error controller gains are based on linear models of the closed inner loop dynamics with disturbance settling times and closed loop stability margins as design criteria.

5.2.3 Control of the Vertical Flight Path

Reference Model for the Vertical Flight Path

The second-order reference dynamics for the flight path angle is given by

$$\begin{aligned} \begin{bmatrix} \ddot{\gamma}_{RM} \\ \dot{\gamma}_{RM} \end{bmatrix} &= \underbrace{\begin{bmatrix} -\frac{T_{RM,1,\gamma} + T_{RM,2,\gamma}}{T_{RM,1,\gamma}T_{RM,2,\gamma}} & -\frac{1}{T_{RM,1,\gamma}T_{RM,2,\gamma}} \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}_{RM,\gamma}} \begin{bmatrix} \dot{\gamma}_{RM} \\ \gamma_{RM} \end{bmatrix} \\ &+ \underbrace{\begin{bmatrix} \frac{1}{T_{RM,1,\gamma}T_{RM,2,\gamma}} & \frac{T_{RM,1,\gamma} + T_{RM,2,\gamma}}{T_{RM,1,\gamma}T_{RM,2,\gamma}} \\ 0 & -1 \end{bmatrix}}_{\mathbf{B}_{RM,\gamma}} \begin{bmatrix} \gamma_C \\ v_{H,\dot{\gamma}} \end{bmatrix}, \end{aligned} \quad (5.28)$$

$$\begin{bmatrix} \gamma_{RM} \\ v_{RM,\dot{\gamma}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{B}_{RM,\gamma}} \begin{bmatrix} \dot{\gamma}_{RM} \\ \gamma_{RM} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{C}_{RM,\gamma}} \begin{bmatrix} \gamma_C \\ v_{H,\dot{\gamma}} \end{bmatrix}. \quad (5.29)$$

with $T_{RM,1,\gamma}$ and $T_{RM,2,\gamma}$ being the time constants of the reference dynamics, the reference flight path angle γ_{RM} and flight path angle rate/vertical flight path curvature $\dot{\gamma}_{RM}$ the states, flight path angle command γ_C and hedging signal $v_{H,\dot{\gamma}}$ the inputs and reference flight path angle and pseudo control $v_{RM,\dot{\gamma}}$ the outputs.

The reference model integrators are reset whenever the flight path angle or vertical speed mode is activated. The reference flight path angle is initialized at the current measured flight path angle. The reference flight path curvature is initialized at the estimated one.

Control of Flight Path Angle or Vertical Speed

The AFCS provides two alternate modes for the control of the vertical flight path: flight path angle mode (FPA) or vertical speed mode (V/S), and the option to toggle between the two modes via the MCP. For the vertical speed mode, an equivalent flight path angle command is calculated from the vertical speed command according to

$$\gamma_{C,VS} = \sin^{-1} \left(\frac{\dot{h}_C}{V_K} \right). \quad (5.30)$$

The equivalent flight path angle command is then fed to the reference model.

Reference Model Limiters

The flight path angle reference model contains limitations of the input command, reference state, and the reference state derivatives. The flight path angle command is limited according to

$$\gamma_{min} \leq \gamma_C \leq \gamma_{max}, \quad (5.31)$$

with the maximum or minimum flight path angle command limited according to nominal descent angle $\gamma_{min/max,nom}$ or descent rate $\dot{h}_{min/max,nom}$ operational limits or desired operational envelope, or maximum/minimum flight angle from the energy-rate prioritization and energy protection,

$$\gamma_{max,\gamma_C} = \min \left\{ \begin{array}{l} \gamma_{max,nom} \\ \sin^{-1} \left(\frac{\dot{h}_{max,nom}}{V_K} \right) \\ \gamma_{max,prot} \end{array} \right\}, \quad (5.32)$$

$$\gamma_{min,\gamma_C} = \max \left\{ \begin{array}{l} \gamma_{min,nom} \\ \sin^{-1} \left(\frac{\dot{h}_{min,nom}}{V_K} \right) \\ \gamma_{min,prot} \end{array} \right\}. \quad (5.33)$$

The desired flight path angle rate $v_{RM,\dot{\gamma}}$ is limited according to

$$\dot{\gamma}_{min} \leq v_{RM,\dot{\gamma}} \leq \dot{\gamma}_{max}, \quad (5.34)$$

with the maximum and minimum flight path angle rate given by the desired or permissible specific normal force $\Delta f_{z,K,max/min}$ or the permissible vertical path curvature from the force prioritization,

$$\dot{\gamma}_{max} = \min \left\{ \begin{array}{l} \frac{1}{V_K} (-\Delta f_{z,K,min}) \\ \dot{\gamma}_{max,lprio} \end{array} \right\}, \quad (5.35)$$

$$\dot{\gamma}_{min} = \max \begin{cases} \frac{1}{V_K} (-\Delta f_{z,K,max}) \\ \dot{\gamma}_{min,lprio} \end{cases} \quad (5.36)$$

The second derivative of the flight path angle, i.e., the rate of change of the flight path curvature may also be limited,

$$\ddot{\gamma}_{min} \leq \ddot{\gamma}_{RM} \leq \ddot{\gamma}_{max}, \quad (5.37)$$

for example, according to the available normal specific force rate of change, limited by the bandwidth of the inner loop controller. Finally, the reference flight path angle integrator is limited according to the same limits as in Eq. (5.43)-(5.44), to ensure that the reference flight path angle stays within the permissible flight path range after the hedging signal added after the flight path angle rate limiter,

$$\gamma_{min} \leq \gamma_{RM} \leq \gamma_{max}. \quad (5.38)$$

Control of the Vertical Flight Path Error

To stabilize the vertical flight path error dynamics,

$$e_\gamma = \gamma_{RM} - \gamma_{K,meas}, \quad (5.39)$$

the error controller adds a proportional and integral part to the pseudo-controls entering the inversion, according to

$$\begin{aligned} v_{\dot{\gamma}} &= v_{RM,\dot{\gamma}} + v_{P,\dot{\gamma}} + v_{I,\dot{\gamma}} \\ &= v_{RM,\dot{\gamma}} + k_{P,\gamma} \cdot e_\gamma + \int k_{I,\gamma} \cdot e_\gamma dt. \end{aligned} \quad (5.40)$$

or on matrix/vector form

$$\begin{bmatrix} \dot{v}_{I,\dot{\gamma}} \\ v_{I,\dot{\gamma}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{A}_{EC,\gamma}} \begin{bmatrix} v_{I,\dot{\gamma}} \\ v_{I,\dot{\gamma}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{I,\gamma} \\ 0 \end{bmatrix}}_{\mathbf{B}_{EC,\gamma}} [e_\gamma], \quad (5.41)$$

$$\begin{bmatrix} v_{P,\dot{\gamma}} \\ v_{I,\dot{\gamma}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{C}_{EC,\gamma}} \begin{bmatrix} v_{I,\dot{\gamma}} \\ v_{I,\dot{\gamma}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{P,\gamma} \\ 0 \end{bmatrix}}_{\mathbf{D}_{EC,\gamma}} [e_\gamma]. \quad (5.42)$$

The error controller integrator is reset whenever the flight path angle or vertical speed mode is activated. The integrator state is initialized at zero.

5.2.4 Control of the Lateral Flight Path

Reference Model for the Lateral Flight Path

The second-order reference dynamics for the flight path angle is given by

$$\begin{bmatrix} \ddot{\chi}_{RM} \\ \dot{\chi}_{RM} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{T_{RM,1,\chi} + T_{RM,2,\chi}}{T_{RM,1,\chi}T_{RM,2,\chi}} & -\frac{1}{T_{RM,1,\chi}T_{RM,2,\chi}} \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}_{RM,\chi}} \begin{bmatrix} \dot{\chi}_{RM} \\ \chi_{RM} \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & T_{RM,1,\chi} + T_{RM,2,\chi} \\ T_{RM,1,\chi}T_{RM,2,\chi} & T_{RM,1,\chi}T_{RM,2,\chi} \\ 0 & -1 \end{bmatrix}}_{\mathbf{B}_{RM,\chi}} \begin{bmatrix} \chi_C \\ v_{H,\dot{\chi}} \end{bmatrix}, \quad (5.43)$$

$$\begin{bmatrix} \chi_{RM} \\ v_{RM,\dot{\chi}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{C}_{RM,\chi}} \begin{bmatrix} \dot{\chi}_{RM} \\ \chi_{RM} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{D}_{RM,\chi}} \begin{bmatrix} \chi_C \\ v_{H,\dot{\chi}} \end{bmatrix}. \quad (5.44)$$

with $T_{RM,1,\chi}$ and $T_{RM,2,\chi}$ being the time constants of the reference dynamics, the reference track angle χ_{RM} and track angle rate / lateral path curvature $\dot{\chi}_{RM}$ the states, track angle command χ_C and hedging signal $v_{H,\dot{\chi}}$ the inputs and reference track angle and pseudo control $v_{RM,\dot{\chi}}$ the outputs.

The reference model integrators are reset whenever the track or heading mode is activated (or toggled between them). The reference track or heading is set to the current track or heading, and the reference turn or heading rate is set to the current turn or heading rate.

Control of Heading or Track

The AFCS provides two alternate modes for the control of the lateral flight path: track angle mode (TRK) or heading mode (HDG), and the option to toggle between the two modes via the MCP. The ability to toggle between TRK and HDG can be found in modern airliner AFCS, with INS providing the flight path information. For the heading mode, the lateral path reference model takes the heading command Ψ_C as the desired target, and the current heading Ψ is fed back to calculate the control error as the difference from the desired heading. The heading rate for the reference model reset and pseudo-control hedging is calculated from Eq. (3.30) according to

$$\dot{\Psi} = q \frac{\sin \Phi}{\cos \theta} + r \frac{\cos \Phi}{\cos \theta}. \quad (5.45)$$

Whereas the kinematic flight path modes, i.e., FPA and TRK, are controlling the flight path and commanding a desired flight path curvature, in HDG mode, the lateral path controller effectively operates as an aircraft attitude controller. This contradicts the architecture principle illustrated in Figure 5.10 with a kinematic flight path control independent of aircraft attitude. However, the HDG mode is included here as it is a standard mode with the same reference dynamics as the TRK mode.

Reference Model Flip Blocks and Limiters

The reference model for the lateral path includes flip blocks instead of limiters for the input command and the reference state integrator to ensure a commanded angle $\chi_C \in [0, 2\pi]$ and reference track angle $\chi_{RM} \in [0, 2\pi]$.

The difference between the track angle command and the reference track angle is flipped to the range $(\chi_C - \chi_{RM}) \in [-\pi, \pi]$ to ensure that a turn to acquire a new track angle target is made in the direction of the smallest angle error. This is the default behavior of the track angle mode. In selected operation however, the pilot or operator is able to select a desired turn direction (see description of HMI design and operation in Section 5.4), in which case the sign of the difference between the track angle command and the reference track angle is enforced according to

$$\text{sgn}(v_{RM,\dot{\chi}}) = \begin{cases} -1, & \text{Left turn desired} \\ 1, & \text{Right turn desired} \\ \text{sgn}(\chi_C - \chi_{RM}), & \text{No turn direction specified} \end{cases} \quad (5.46)$$

The desired turn rate pseudo control $v_{RM,\dot{\chi}}$ integrator is limited according to

$$\dot{\chi}_{min} \leq v_{RM,\dot{\chi}} \leq \dot{\chi}_{max} \quad (5.47)$$

with the maximum and minimum turn rate given by the desired or permissible nominal turn rate $\dot{\chi}_{min/max,nom}$ or the permissible lateral path curvature from the force prioritization,

$$\dot{\chi}_{max} = \min \begin{cases} \dot{\chi}_{max,nom} \\ \dot{\chi}_{max,prio} \end{cases} \quad (5.48)$$

$$\dot{\chi}_{min} = \max \begin{cases} \dot{\chi}_{min,nom} \\ \dot{\chi}_{min,prio} \end{cases} \quad (5.49)$$

The desired or permissible maximum turn rate $\dot{\chi}_{max,nom}$ may be set to a fixed value or defined as for example speed dependent rate half, rate one, or rate two turn.

The second derivative of the track angle, i.e., the rate of change of the lateral flight path curvature may also be limited,

$$\ddot{\chi}_{min} \leq \ddot{\chi}_{RM} \leq \ddot{\chi}_{max} \quad (5.50)$$

for example, according to the maximum kinematic bank angle rate $\dot{\mu}_K$, limited by the roll bandwidth of the inner loop controller.

Control of the Lateral Flight Path Error

To stabilize the lateral flight path error dynamics,

$$e_\chi = \chi_{RM} - \chi_{K,meas} \quad (5.51)$$

the error controller adds a proportional and integral part to the pseudo controls entering the inversion, according to

$$\begin{aligned} v_{\dot{\chi}} &= v_{RM,\dot{\chi}} + v_{P,\dot{\chi}} + v_{I,\dot{\chi}} \\ &= v_{RM,\dot{\chi}} + k_{P,\chi} \cdot e_\chi + \int k_{I,\chi} \cdot e_\chi dt. \end{aligned} \quad (5.52)$$

or on matrix/vector form

$$\begin{bmatrix} \dot{v}_{I,\dot{\chi}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{\mathbf{A}_{EC,\chi}} \begin{bmatrix} v_{I,\dot{\chi}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{I,\chi} \end{bmatrix}}_{\mathbf{B}_{EC,\chi}} \begin{bmatrix} e_{\chi} \end{bmatrix}, \quad (5.53)$$

$$\begin{bmatrix} v_{P,\dot{\chi}} \\ v_{I,\dot{\chi}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{C}_{EC,\chi}} \begin{bmatrix} v_{I,\dot{\chi}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{P,\chi} \\ 0 \end{bmatrix}}_{\mathbf{D}_{EC,\chi}} \begin{bmatrix} e_{\chi} \end{bmatrix}. \quad (5.54)$$

The error controller integrator is reset whenever the track or heading mode is activated. The integrator state is initialized at zero.

5.2.5 Control of the Aerodynamic Speed

Reference Model for Aerodynamic Speed

The second-order reference dynamics for the aerodynamic speed is given by

$$\begin{bmatrix} \dot{V}_{RM} \\ \dot{v}_{RM,\dot{\psi}} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{T_{RM,1,V} + T_{RM,2,V}}{T_{RM,1,V}T_{RM,2,V}} & -\frac{1}{T_{RM,1,V}T_{RM,2,V}} \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}_{RM,V}} \begin{bmatrix} V_{RM} \\ v_{RM,\dot{\psi}} \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & \frac{T_{RM,1,V} + T_{RM,2,V}}{T_{RM,1,V}T_{RM,2,V}} \\ \frac{1}{T_{RM,1,V}T_{RM,2,V}} & -1 \end{bmatrix}}_{\mathbf{B}_{RM,V}} \begin{bmatrix} V_C \\ v_{H,\dot{\psi}} \end{bmatrix}, \quad (5.55)$$

$$\begin{bmatrix} V_{RM} \\ v_{RM,\dot{\psi}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{C}_{RM,V}} \begin{bmatrix} \dot{V}_{RM} \\ \dot{v}_{RM,\dot{\psi}} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{D}_{RM,V}} \begin{bmatrix} V_C \\ v_{H,\dot{\psi}} \end{bmatrix}. \quad (5.56)$$

with $T_{RM,1,V}$ and $T_{RM,2,V}$ being the time constants of the reference dynamics, the reference airspeed V_{RM} and acceleration \dot{V}_{RM} the states, the airspeed command V_C and hedging signal $v_{H,\dot{\psi}}$ the inputs, and reference airspeed and pseudo-control $v_{RM,\dot{\psi}}$ the outputs.

The reference model integrators are reset if speed control is activated (speed by pitch or by thrust), but not when switching between the two speed modes. The initial condition for the speed reference value is the current indicated airspeed. The initial condition for the speed reference rate value is the current estimated acceleration.

Speed by Thrust or Speed by Pitch

The AFCS provides two alternate modes for the control of the (indicated) aerodynamic speed, coupled with the control mode for the vertical flight path: speed by thrust or speed by pitch, see the principal coupling between automatic flight control modes and autopilot/aut thrust functions illustrated in Figure 5.2. In speed by thrust mode, the reference model produces a desired acceleration that is inverted to a desired incremental specific force along the flight path as command to the thrust loop controller. In speed by pitch mode, the desired incremental specific force along the flight path is converted to an equivalent flight path angle command that is fed directly to the flight path error controller, from Eq. (3.100),

$$\gamma_{K,v\dot{V}} = \sin^{-1} \left(\frac{\dot{V}_K - v\dot{V}}{g} + \sin \gamma_K \right). \quad (5.57)$$

The acceleration along the flight path achievable by the inner loop controller through the $\Delta f_{z,K}$ dynamics will be different than the acceleration achievable by the thrust loop controller, i.e., the $\Delta f_{x,K}$ dynamics. Thus, the dynamics of the reference model is adjusted depending on the active speed mode, and the time constants $T_{1,V}$ and $T_{2,V}$ as well as the error controller gains are designed separately for the speed by thrust and the speed by pitch modes.

Reference Model Limiters

The speed reference model contains limitations of the input command, reference state, and the reference state derivatives. The aerodynamic speed command is limited according to

$$V_{IAS,min} \leq V_C \leq V_{IAS,max}, \quad (5.58)$$

with the aerodynamic speed limits adjusted according to the current aircraft configuration (flap and gear configuration).

The acceleration pseudo-control $v_{RM,\dot{V}}$ is limited according to

$$\dot{V}_{min} \leq v_{RM,\dot{V}} \leq \dot{V}_{max}, \quad (5.59)$$

with acceleration limits set to either the desired nominal acceleration to acquire a new speed target $\dot{V}_{min/max,nom}$, or the acceleration limits from the energy-rate prioritization for either speed or path priority,

$$\dot{V}_{max} = \min \left\{ \begin{array}{l} \dot{V}_{max,nom}, \\ \dot{V}_{max,prio}, \end{array} \right. \quad (5.60)$$

$$\dot{V}_{min} = \max \left\{ \begin{array}{l} \dot{V}_{min,nom}, \\ \dot{V}_{min,prio}, \end{array} \right. \quad (5.61)$$

The acceleration rate of change \dot{V}_{RM} may also be limited according to

$$\dot{V}_{min} \leq \dot{V}_{RM} \leq \dot{V}_{max}, \quad (5.62)$$

for example, according to the desired thrust rate of change, limited by the bandwidth of the thrust controller, or when in speed by pitch mode, by the bandwidth of the inner loop controller, determining the available flight path rate of change and thus the acceleration rate of change along the flight path. Finally, the reference aerodynamic speed integrator is limited according to the same limits as in Eq. (5.85), to ensure that the reference speed stays within the permissible speed range after the hedging signal added after the acceleration pseudo control limiter,

$$V_{min} \leq V_{RM} \leq V_{max}. \quad (5.63)$$

Open Climb/Descent

When the speed by pitch mode is coupled with an engaged autothrust, the autothrust provides a climb/descent thrust setting depending on the altitude target set, and the speed by pitch mode results in an open climb/descent.

The speed by pitch flight path angle command is limited according to:

- In open climb, the minimum flight path angle is zero (no descent allowed in order to increase speed)
- In open descent, the maximum flight angle is zero (no climb allowed in order to decrease speed)

For example, when in an open climb and setting a new airspeed target larger than the current airspeed, the AFCS will reduce the climb angle in order to redistribute the energy rate to accelerate the aircraft at the desired acceleration limit, or at flight path angle equal to zero, whichever is largest.

In order to prevent a step in the flight path angle error in case of a direct switch between open climb and open descent, the maximum and minimum flight path angle limits are filtered through rate-limited PT1 filters, initialized at the current flight path angle when the climb or descent mode is activated,

$$\gamma_{min/max,CLB/DES,filtered} = \frac{1}{T_{CLB/DES} \cdot s + 1} \gamma_{min/max,CLB/DES} \quad (5.64)$$

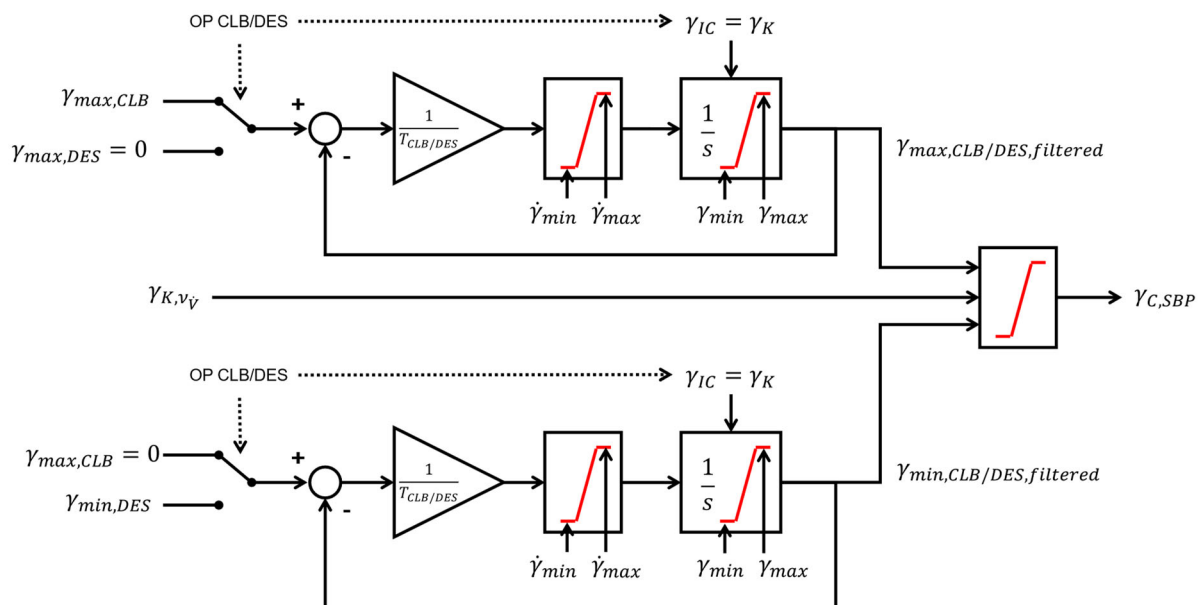


Figure 5.14: Flight path angle limit filtering at open climb/descent activation.

The limits are opened up at the maximum/minimum permissible flight path angle rate limits, see Figure 5.14. The rate limits are based on the kinematic speed and the permissible incremental specific forces in the vertical plane.

Complimentary Airspeed Filtering and Thrust Activity

A complimentary filtering of the measured indicated airspeed $V_{IAS,meas}$ and the estimated kinematic acceleration $\hat{v}_{\dot{V}_K}$ is added, to attenuate the influence of turbulence on the speed error signal and thus on the path acceleration command fed to the autothrust controller. The complimentary filtered airspeed V_{CF} is given by

$$V_{CF} = \frac{1}{T_{V,CF}S + 1} \cdot V_{IAS,meas} + \frac{T_{V,CF}}{T_{V,CF}S + 1} \cdot \hat{v}_{\dot{V}_K} \quad (5.65)$$

with $T_{V,CF}$ being the filter time constant.

To avoid noise in the specific force measurements directly impacting the incremental throttle position command, the estimated kinematic acceleration $\hat{v}_{\dot{V}_K}$ is low-pass filtered, see Section 5.2.11.

Control of the Speed Error

To stabilize the speed error dynamics,

$$e_V = V_{RM} - V_{CF}, \quad (5.66)$$

the error controller adds a proportional and integral part to the pseudo controls entering the inversion, according to

$$\begin{aligned} v_{\dot{V}} &= v_{RM,\dot{V}} + v_{P,\dot{V}} + v_{I,\dot{V}} \\ &= v_{RM,\dot{V}} + k_{P,V} \cdot e_V + \int k_{I,V} \cdot e_V dt. \end{aligned} \quad (5.67)$$

or on matrix/vector form

$$\begin{bmatrix} \dot{v}_{I,\dot{V}} \\ v_{I,\dot{V}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{A}_{EC,V}} \begin{bmatrix} v_{I,\dot{V}} \\ v_{I,\dot{V}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{I,V} \\ 0 \end{bmatrix}}_{\mathbf{B}_{EC,V}} [e_V], \quad (5.68)$$

$$\begin{bmatrix} v_{P,\dot{V}} \\ v_{I,\dot{V}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{C}_{EC,V}} \begin{bmatrix} v_{I,\dot{V}} \\ v_{I,\dot{V}} \end{bmatrix} + \underbrace{\begin{bmatrix} k_{P,V} \\ 0 \end{bmatrix}}_{\mathbf{D}_{EC,V}} [e_V]. \quad (5.69)$$

The error controller integrator is reset whenever the speed by thrust or speed by pitch mode, but not when switching between the two speed modes. The integrator state is initialized at zero.

5.2.6 Trajectory Control Modes

A dedicated trajectory control module is embedded in the AFCS module, as illustrated in Figure 5.10. The trajectory controller, developed at TUM-FSD in conjunction with the AFCS and inner loop controller as part of the modular FGCS, is described in detail in Schatz et al. [76, 111]. It

provides the same inner loop interface, i.e., the incremental specific forces in the kinematic frame. The trajectory controller takes the desired trajectory angles χ_T and γ_T , defined in the trajectory frame T , see Figure 5.15, the desired trajectory angular rates $\dot{\chi}_T$ and $\dot{\gamma}_T$, the desired trajectory angular accelerations $\ddot{\chi}_T$ and $\ddot{\gamma}_T$, as well as the deviations between aircraft and desired trajectory $(\vec{\mathbf{r}}^{FR})_T$, and corresponding time derivatives of the deviations $(\vec{\mathbf{V}}_K^{FR})_T^T$, according to

$$(\vec{\mathbf{r}}^{FR})_T = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}_T \triangleq \begin{bmatrix} x^R - x^F \\ y^R - y^F \\ z^R - z^F \end{bmatrix}_T, \quad (5.70)$$

$$(\vec{\mathbf{V}}_K^{FR})_T^T = \begin{bmatrix} \Delta \dot{x} \\ \Delta \dot{y} \\ \Delta \dot{z} \end{bmatrix}_T^T \triangleq \begin{bmatrix} \dot{x}^R - \dot{x}^F \\ \dot{y}^R - \dot{y}^F \\ \dot{z}^R - \dot{z}^F \end{bmatrix}_T^T. \quad (5.71)$$

From the perspective of the trajectory controller, any source of trajectory deviations and additional trajectory information is permitted. Thus, for the purpose of the modular FGCS, the trajectory controller is utilized to realize the altitude hold and vertical/lateral trajectory navigation modes, as well as desired trajectory and deviations from the automatic landing and takeoff module [84]. The trajectory controller is embedded in the AFCS module, as the AFCS provides the mode transition logic and output command selection for the path and trajectory control modes.

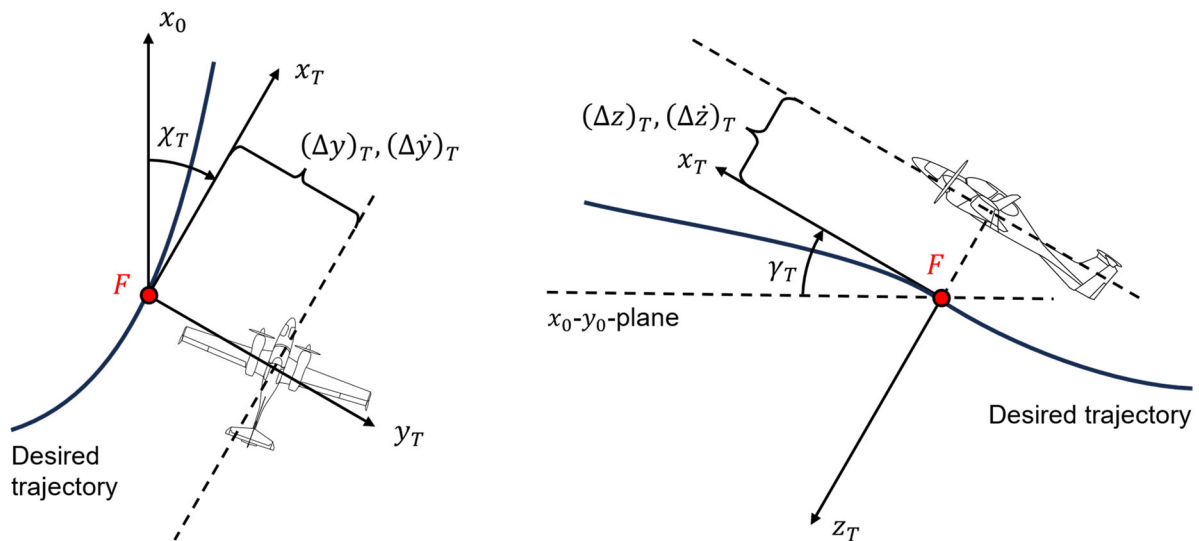


Figure 5.15: Trajectory frame as utilized by the trajectory controller with the trajectory deviations and their corresponding time derivatives, adopted from [111].

Vertical/Lateral Trajectory Navigation

When in managed operation, the AFCS forwards desired trajectory and deviation commands based on waypoints and commands from a trajectory generation module [78] (VNAV and LNAV modes in Table 5.3).

Altitude Capture/Hold

When in selected or managed operation, the AFCS automatically arms the altitude capture mode (ALT*) if climbing or descending towards a preselected altitude, except when in glideslope mode.

The altitude capture mode is calculating a transition arc to the desired altitude, based on the altitude deviation $\Delta h = h - h_C$ and a constant desired capture-specific force increment $\Delta f_{z,K,ALT^*}$, see Figure 5.16. Desired for the transition arc is a flight path angle command γ_{C,ALT^*} as function of the altitude deviation.

The vertical speed and its rate of change are given by

$$\dot{h} = V_K \sin \gamma_K \quad (5.72)$$

$$\ddot{h} = V_K \cos \gamma_K \cdot \dot{\gamma}_K = V_K \cos \gamma_K \cdot \left(-\frac{1}{V_K} \cdot \Delta f_{z,K} \right) \approx -\Delta f_{z,K} \quad (5.73)$$

with the flight path curvature $\dot{\gamma}_K$ in terms of incremental specific force $\Delta f_{z,K}$ according to Eq. (3.59) and the small angle approximation $\cos \gamma_K \approx 1$.

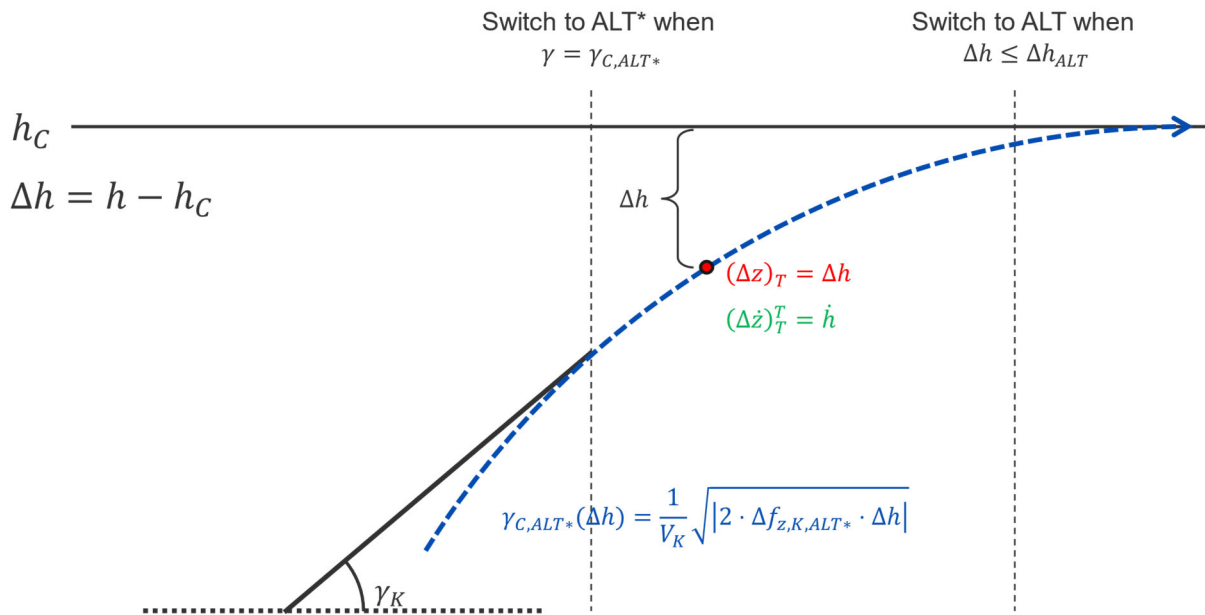


Figure 5.16: Altitude capture and track geometry.

Integration of Eq. (5.73) with respect to time gives

$$\dot{h}(t) = -\Delta f_{z,K} \cdot t \quad (5.74)$$

$$h(t) = -\frac{1}{2} \cdot \Delta f_{z,K} \cdot t^2 + h_C \quad (5.75)$$

$$\Delta h = h(t) - h_C = -\frac{1}{2} \cdot \Delta f_{z,K} \cdot t^2 \quad (5.76)$$

$$t^2 = -\frac{2 \cdot \Delta h}{\Delta f_{z,K}} \Rightarrow t = \sqrt{-\frac{2 \cdot \Delta h}{\Delta f_{z,K}}} \quad (5.77)$$

An altitude capture from above, with $\Delta h > 0$, requires a positive flight path curvature, with $\Delta f_{z,K} < 0$. An altitude capture from below, with $\Delta h < 0$, requires a negative flight path curvature, with $\Delta f_{z,K} > 0$. Thus, the expression under the root sign in Eq. (5.77) is always positive. Inserting this into Eq. (5.74) gives the desired vertical speed as a function of the altitude deviation,

$$\dot{h}(\Delta h) = \begin{cases} -\sqrt{-2 \cdot \Delta f_{z,K} \cdot \Delta h}, & \Delta h < 0 \\ \sqrt{-2 \cdot \Delta f_{z,K} \cdot \Delta h}, & \Delta h > 0 \end{cases} \quad (5.78)$$

With the small angle approximation $\sin \gamma_K \approx \gamma_K$, the resulting flight path angle command for the altitude capture transition arc is given by

$$\gamma_{C,ALT^*}(\Delta h) = \begin{cases} -\frac{1}{V_K} \sqrt{|2 \cdot \Delta f_{z,K,ALT^*} \cdot \Delta h|}, & \Delta h < 0 \\ \frac{1}{V_K} \sqrt{|2 \cdot \Delta f_{z,K,ALT^*} \cdot \Delta h|}, & \Delta h > 0 \end{cases} \quad (5.79)$$

with a specified altitude capture specific force increment $\Delta f_{z,K,ALT^*}$. The altitude capture mode is activated when the flight path angle command for the transition arc equals the current climb or descent flight path angle, $\gamma_K = \gamma_{C,ALT^*}$, and feeds the capture command to the flight path angle error controller.

The altitude hold mode is realized by the trajectory controller, with the altitude deviation being the deviation from the desired trajectory, and the vertical speed the corresponding deviation time derivative, according to

$$(\Delta z)_T = (z^R - z^F)_T = h - h_C = \Delta h, \quad (5.80)$$

$$(\Delta \dot{z})_T^T = (\dot{z}^R - \dot{z}^F)_T^T = \dot{h} - 0 = \dot{h}. \quad (5.81)$$

The altitude hold mode is activated when the deviation threshold reaches a certain value.

5.2.7 Radio Navigation & Approach Modes

The AFCS module provides a basic implementation of the state-of-the-art radio navigation and approach modes

- Glideslope Capture/Track (G/S*, G/S)
- VOR/Localizer Capture/Track (VOR/LOC*, VOR/LOC)

The glideslope capture and track modes convert the glideslope deviation into a flight path angle command that is fed directly to the flight path angle error controller, bypassing the flight path angle reference model. The VOR/localizer capture and track modes convert the course angle deviation to a track angle error that is fed directly to the track angle error controller, bypassing the track angle reference model.

For the flare maneuver, the AFCS module provides a dedicated command input, utilized by the ATOL module during automated landing maneuvers. The ATOL module commands a vertical speed during the flare maneuver, which is converted in the AFCS module to an equivalent flight path angle command that is fed directly to the flight path angle error controller, bypassing the flight path angle reference model similar to the glideslope mode. The flare mode has a dedicated set of gains, as the feedback structure and thus the closed loop dynamics changes when the reference model is bypassed.

The functionality of the radio navigation and approach modes have been demonstrated in MIL and HIL simulations. As the functional design and implementation of these modes do not constitute a significant contribution beyond the state-of-the-art, a further description of these modes is omitted here.

5.2.8 Energy-Based Prioritizations

As shown in Section 3.5, a given energy flow rate can be distributed into either an acceleration along the flight path or a flight path angle, or some combination thereof. Since the energy flow rate provided by the propulsion system is physically limited, arbitrary combinations of acceleration and flight path angle cannot be achieved, and thus a control objective conflict in the tracking of speed and flight path commands will occur when the energy flow rate control is saturated or disengaged. Based on the energy principles derived in Section 3.5, an active prioritization of flight path or acceleration (and thus speed) command tracking is achieved by a dynamic distribution of the energy flow rate in favor of either flight path or acceleration, or some combination thereof. Furthermore, in order to protect the airspeed envelope and the energy integrity of the aircraft, the energy flow rate can automatically be distributed in favor of the acceleration at the envelope edges.

The energy flow rate distribution is realized by cross-coupling the flight path angle and speed reference models according to Figure 5.17, and setting the flight path angle command and integrator limiters, and the acceleration limiter, respectively [72, 74].

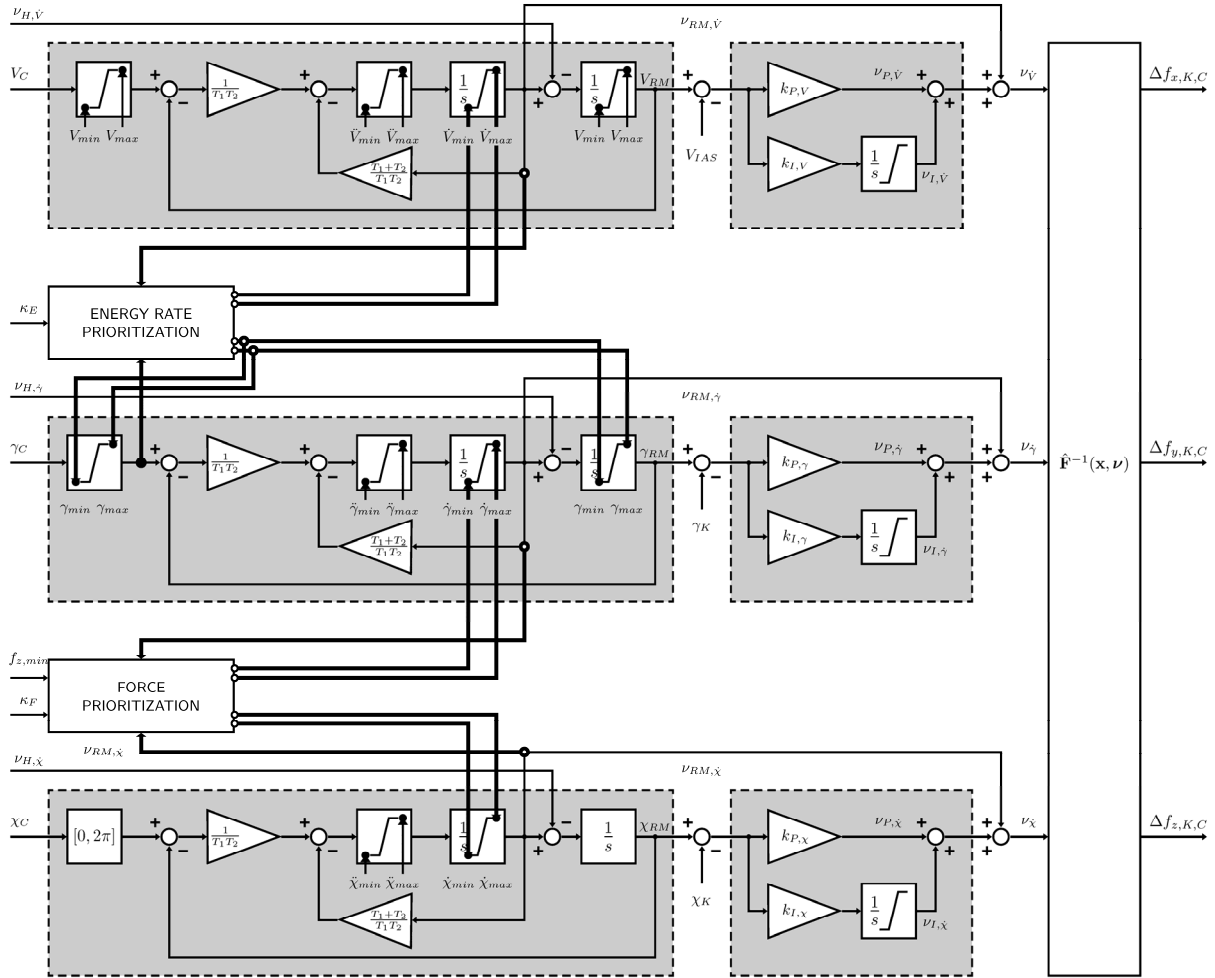


Figure 5.17: Basic flight path control structure with control plane coupling for energy rate and force prioritizations.

Speed vs Vertical Path Prioritization

The Eq. (3.100) and (3.101) may now be used to set the limiters in the corresponding reference model, in order to prioritize either the commanded flight path angle (by limiting the pseudo control acceleration in the speed reference model) or the commanded speed (by limiting the reference flight path angle).

In the case autothrust is off, the constraints for speed priority (denoted VPRIO) on the flight path angle are given by

$$\gamma_{max,vprio} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{V}} - \hat{v}_{\dot{V}}}{g} \right), \quad (5.82)$$

$$\gamma_{min,vprio} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{V}} - \hat{v}_{\dot{V}}}{g} \right), \quad (5.83)$$

where $v_{RM,\dot{V}}$ is the acceleration command from the speed reference model, and $\hat{v}_{\dot{V}}$ the estimated acceleration.

In path priority mode (denoted PPRIO) the constraints on the acceleration are given by

$$v_{RM,\dot{V},max,pprio} = \hat{v}_{\dot{V}} - g(\sin \gamma_C - \sin \gamma_K), \quad (5.84)$$

$$v_{RM,\dot{V},min,pprio} = \hat{v}_{\dot{V}} - g(\sin \gamma_C - \sin \gamma_K). \quad (5.85)$$

The Eq. (5.82) to (5.85) realizes a binary speed or path priority mode control where only the current energy flow (which results from the current thrust setting) is considered. In the case where the autothrust (energy flow rate control) is active, the potential acceleration or flight path angle may be amended by an estimation of the additional energy flow rate $\Delta \dot{h}_{E,max/min} = V_K \Delta f_{x,K,max/min}/g$ automatically provided by the autothrust, according to Eq. (3.102), with $\Delta f_{x,K,max/min}$ being the maximum or minimum additional available specific force along the x_K -axis, see Eq. (5.150). The autothrust function is further described in Section 5.2.11.

In the case of autothrust on, i.e. active energy flow rate control, the consideration of anticipated available excess thrust can be included in the calculation of Eq. (5.82) to (5.85) according to

$$\gamma_{max,vprio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{V}} - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,max}}{g} \right), \quad (5.86)$$

$$\gamma_{min,vprio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{V}} - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,min}}{g} \right), \quad (5.87)$$

and

$$v_{RM,\dot{V},max,pprio,ATHR} = \hat{v}_{\dot{V}} - g(\sin \gamma_C - \sin \gamma_K) + \Delta f_{x,K,max}, \quad (5.88)$$

$$v_{RM,\dot{V},min,pprio,ATHR} = \hat{v}_{\dot{V}} - g(\sin \gamma_C - \sin \gamma_K) + \Delta f_{x,K,min}. \quad (5.89)$$

Mixed Energy Rate Authority Prioritization

The previous section introduced the speed versus flight path angle prioritization, where the energy rate distribution always is performed in favor of one of the two variables. In this section, an integration of the two modes of prioritization is presented, where an available energy rate distribution authority, based on the permissible maximum and minimum acceleration limits, is budgeted between acceleration and flight path angle.

The specific speed-normalized energy rate distribution authority, a measure of how fast energy may be traded between acceleration and flight path, is given by

$$\frac{\dot{E}_{dist,max}}{m \cdot V_K} = (\dot{V}_{max,nom} - \dot{V}_{min,nom}), \quad (5.90)$$

i.e., the energy rate distribution between flight path and acceleration is limited so that the permissible nominal acceleration limits are not violated. An energy rate distribution factor $\kappa_E \in [0, 1]$ is introduced, defining the amount of energy rate distribution authority allocated to flight path and acceleration control, where $\kappa_E = 0$ indicates full flight path priority, and $\kappa_E = 1$ full speed priority, and a value in between a mixed priority. The flight path angle and acceleration limits are then a function of κ_E , according to

$$\gamma_{max,prio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{\left((1 - \kappa_E) \dot{V}_{min,nom} + \kappa_E v_{RM,\dot{V}} \right) - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,max}}{g} \right), \quad (5.91)$$

$$\gamma_{min,prio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{\left((1 - \kappa_E) \dot{V}_{max,nom} + \kappa_E v_{RM,\dot{V}} \right) - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,min}}{g} \right), \quad (5.92)$$

and

$$v_{RM,\dot{V},max,prio,ATHR} = \hat{v}_{\dot{V}} + g \left(\sin \gamma_K - \left((1 - \kappa_E) \sin \gamma_C - \kappa_E \frac{\dot{V}_{max,nom}}{g} \right) \right) + \Delta f_{x,K,max}, \quad (5.93)$$

$$v_{RM,\dot{V},min,prio,ATHR} = \hat{v}_{\dot{V}} + g \left(\sin \gamma_K - \left((1 - \kappa_E) \sin \gamma_C - \kappa_E \frac{\dot{V}_{min,nom}}{g} \right) \right) + \Delta f_{x,K,min}. \quad (5.94)$$

If the energy rate distribution factor κ_E equals 0, i.e. full path priority, Eq. (5.91) and (5.92) reduce to

$$\gamma_{max,prio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{\dot{V}_{min,nom} - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,max}}{g} \right), \quad (5.95)$$

$$\gamma_{min,prio,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{\dot{V}_{max,nom} - \hat{v}_{\dot{V}}}{g} + \frac{\Delta f_{x,K,min}}{g} \right), \quad (5.96)$$

i.e., the flight path angles are limited to ensure an energy redistribution within the maximum and minimum nominal acceleration limits. Eq. (5.93) and (5.94) are reduced to Eq. (5.88) and (5.89), respectively, ensuring that the desired flight path angle may be tracked. If the energy rate distribution factor κ_E equals 1, i.e. full speed priority, Eq. (5.91) and (5.92) reduce to Eq.(5.86) and (5.87), respectively, ensuring that the desired pseudo-control acceleration may be tracked.

Energy Integrity Protection

In order to ensure the energy integrity of the aircraft, the energy distribution must be automatically prioritized in favor of the airspeed at the edges of its envelope. The acceleration limits $v_{RM,\dot{V},max/min}$ for the speed reference model are determined as a function of the distance to the airspeed limit $V_{IAS,max/min}$ and the current estimated acceleration $\hat{v}_{\dot{V}}$, and the flight path angle limits $\gamma_{max/min}$ adjusted so that the energy distribution ensures the limit acceleration is maintained, thereby preventing an over- or undershoot of the limit airspeed.

The maximum and minimum indicated airspeeds may be set at fixed values for verification and validation purposes in a limited airspeed envelope or implemented as functions of the stall speed and the aircraft configuration (clean/flaps/landing), with some safety margins depending on the operational scenario and the airspeed envelope needed.

Speed/Acceleration Phase Plane Regions

Two regions in the speed/acceleration phase plane are defined:

- A *transition region*, defined by the distance ΔV_{trans} to the limit speed, where the energy distribution is actively controlled in favor of airspeed
- A *protection region*, defined by the distance ΔV_{prot} beyond the speed limit, where the throttle limits are additionally adjusted beyond nominal limits to prevent a low energy state (from e.g., Maximum Continuous to Take-Off/Go-Around Thrust)

The acceleration limits as function of the speed margin to the respective region are defined as linear slopes in the speed/acceleration phase plane, given by:

$$\dot{V}_{max} = k_{V,max}(V_{IAS,max} - V_{IAS}) \quad (5.97)$$

$$\begin{aligned} \dot{V}_{trans,max} &= k_{V,max}(V_{IAS,trans,max} - V_{IAS}) \\ &= k_{V,max}(V_{IAS,max} - \Delta V_{trans} - V_{IAS}) \end{aligned} \quad (5.98)$$

$$\begin{aligned} \dot{V}_{prot,max} &= k_{V,max}(V_{IAS,prot,max} - V_{IAS}) \\ &= k_{V,max}(V_{IAS,max} - \Delta V_{prot} - V_{IAS}) \end{aligned} \quad (5.99)$$

$$\dot{V}_{min} = k_{V,min}(V_{IAS,min} - V_{IAS}) \quad (5.100)$$

$$\begin{aligned} \dot{V}_{trans,min} &= k_{V,min}(V_{IAS,trans,min} - V_{IAS}) \\ &= k_{V,min}(V_{IAS,min} - \Delta V_{trans} - V_{IAS}) \end{aligned} \quad (5.101)$$

$$\begin{aligned} \dot{V}_{prot,min} &= k_{V,min}(V_{IAS,prot,min} - V_{IAS}) \\ &= k_{V,min}(V_{IAS,min} - \Delta V_{prot} - V_{IAS}) \end{aligned} \quad (5.102)$$

where $k_{V,max}$ and $k_{V,min}$ determines the slope of the limits in the high-speed and low-speed regions, respectively. The phase plane regions with their corresponding limit accelerations are visualized in Figure 5.18. Alternative designs of the phase plane regions using more complex parametrizations than a straight line would provide additional degrees of freedom for optimization but are not explored further within the scope of this thesis.

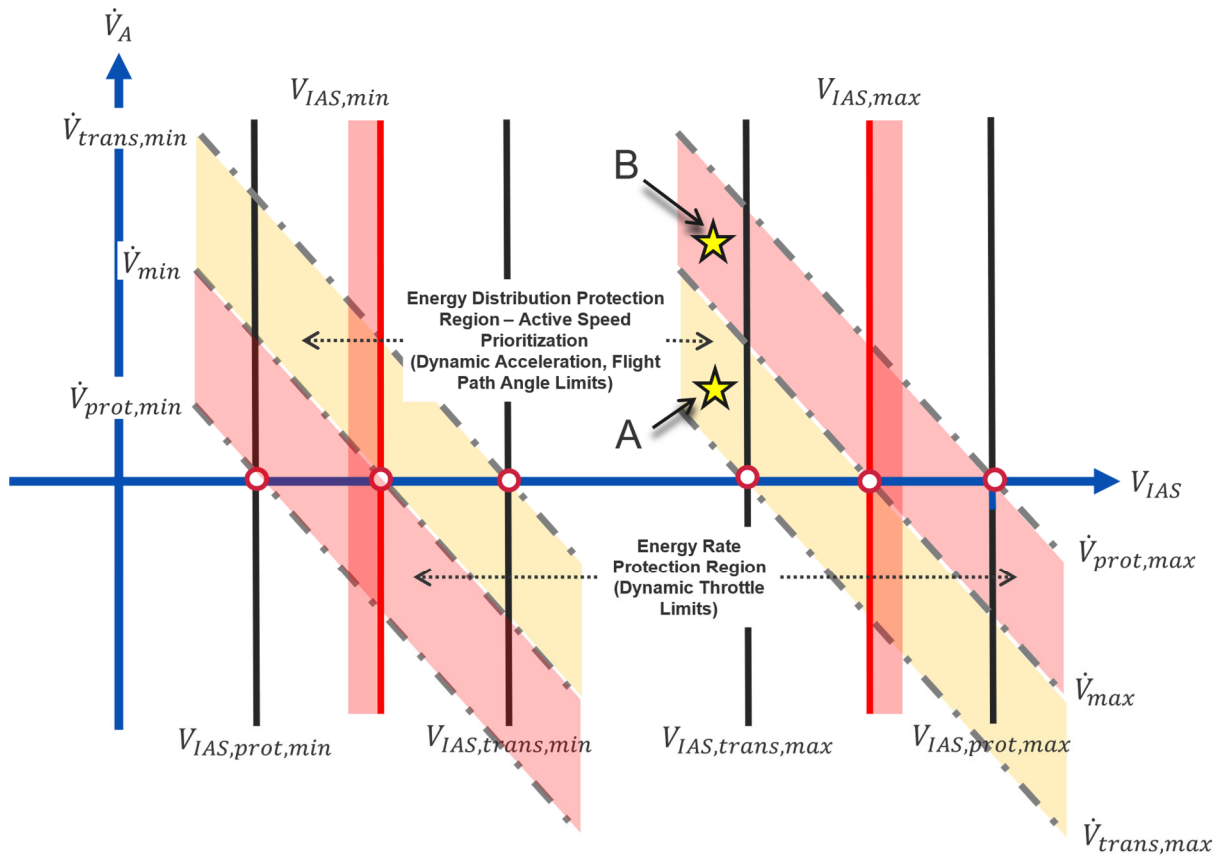


Figure 5.18: Phase plane regions for energy protection.

In Figure 5.18, point A represents a situation where the aircraft approaches its upper speed limit (positive acceleration) and has entered the transition region where the acceleration and flight path limits are adjusted to prevent an overshoot. In case the acceleration is even larger, point B, the throttle limits are further adjusted (in this case to their absolute minimum) in order to further utilize the possible energy flow rate.

Within the transition region, the acceleration limits are reduced from their nominal limits to the protection limits. The implemented protection functionality uses a linear transition according to the pattern illustrated in Figure 5.19, according to

$$y = y_1 + (y_0 - y_1) \left(\frac{\dot{v} - \dot{v}_1}{\dot{v}_0 - \dot{v}_1} \right) \quad (5.103)$$

where

- y_0 is the nominal limit (fixed parameter),
- y_1 is the protection limit (fixed or calculated),
- \dot{v} is the linear acceleration,
- \dot{v}_0 is the acceleration at the transition region start (based on current and transition airspeed) and
- \dot{v}_1 is the acceleration at the transition region end (based on current and limit airspeed).

So as the acceleration \dot{v} goes from \dot{v}_0 to \dot{v}_1 the output limit y is reduced/increased from the nominal limit y_0 to the protection limit y_1 , see Figure 5.19.

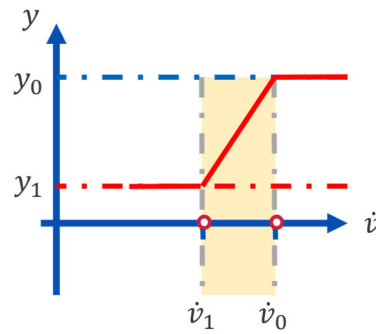


Figure 5.19: Protection limits scaling principle.

As with the design of the phase plane regions, a more complex parametrization of the limit transitions, for example, using higher order polynomials, would provide additional degrees of freedom for optimizing the energy protections, but are not explored further within the scope of this thesis.

Acceleration Limits

For the acceleration, Eq. (5.103) yields

$$v_{RM,\dot{v},max} = 0 + (\dot{V}_{max,nom} - 0) \left(\frac{\hat{v}_{\dot{v}} - \dot{V}_{max}}{\dot{V}_{trans,max} - \dot{V}_{max}} \right), \quad (5.104)$$

$$v_{RM,\dot{v},min} = 0 + (\dot{V}_{min,nom} - 0) \left(\frac{\hat{v}_{\dot{v}} - \dot{V}_{min}}{\dot{V}_{trans,min} - \dot{V}_{min}} \right). \quad (5.105)$$

with $v_{RM,\dot{v},max/min}$ being the acceleration limit set in the speed reference model, $\dot{V}_{max/min,nom}$ the nominal limits, $\hat{v}_{\dot{v}}$ the estimated linear acceleration, and $\dot{V}_{trans,max/min}$ and $\dot{V}_{max/min}$ the transition region limit accelerations according to Eq. (5.97) to (5.102). (For the sake of clarity, the zero, being the permissible acceleration at maximum/minimum speed, is included.) The acceleration limits are visualized in Figure 5.20.

Flight Path Angle Limits

The protection flight path angles equivalent to the minimum/maximum acceleration in the low/high-speed transition regions can be derived from Eq. (3.103), again with the available thrust increment considered in case of autothrust active,

$$\gamma_{max,prot,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{v},min} - \hat{v}_{\dot{v}}}{g} + \frac{\Delta f_{x,K,max}}{g} \right), \quad (5.106)$$

$$\gamma_{min,prot,ATHR} = \sin^{-1} \left(\sin \gamma_K - \frac{v_{RM,\dot{v},max} - \hat{v}_{\dot{v}}}{g} + \frac{\Delta f_{x,K,min}}{g} \right). \quad (5.107)$$

The flight path angle limits and their relation to the acceleration limits are visualized in Figure 5.20.

Throttle Limits

Analogously, the throttle limits $\delta_{T,min/max}$ are adjusted within the high/low-speed protection region according to Eq. (5.103), from the nominal limits $\delta_{T,nom,min/max}$ to their absolute limits $\delta_{T,abs,min/max}$ in order to provide maximum or minimum available thrust, see Figure 5.20. In the low-speed protection region (denoted LS), the maximum and minimum throttle limits are given by

$$\delta_{T,max,prot,LS} = \delta_{T,max,abs} + (\delta_{T,max,nom} - \delta_{T,max,abs}) \left(\frac{\hat{v}_{\dot{V}} - \dot{V}_{prot,min}}{\dot{V}_{min} - \dot{V}_{prot,min}} \right), \quad (5.108)$$

$$\delta_{T,min,prot,LS} = \delta_{T,max,abs} + (\delta_{T,min,nom} - \delta_{T,max,abs}) \left(\frac{\hat{v}_{\dot{V}} - \dot{V}_{prot,min}}{\dot{V}_{min} - \dot{V}_{prot,min}} \right), \quad (5.109)$$

and in the high-speed protection region (denoted HS) analogously by

$$\delta_{T,max,prot,HS} = \delta_{T,min,abs} + (\delta_{T,max,nom} - \delta_{T,min,abs}) \left(\frac{\hat{v}_{\dot{V}} - \dot{V}_{prot,max}}{\dot{V}_{max} - \dot{V}_{prot,max}} \right), \quad (5.110)$$

$$\delta_{T,min,prot,HS} = \delta_{T,min,abs} + (\delta_{T,min,nom} - \delta_{T,min,abs}) \left(\frac{\hat{v}_{\dot{V}} - \dot{V}_{prot,max}}{\dot{V}_{max} - \dot{V}_{prot,max}} \right), \quad (5.111)$$

It is possible to extend the throttle limit adjustment within the protection region to include means of drag control, e.g., speed brakes, to further control the energy flow rate in the case of high-speed protection.

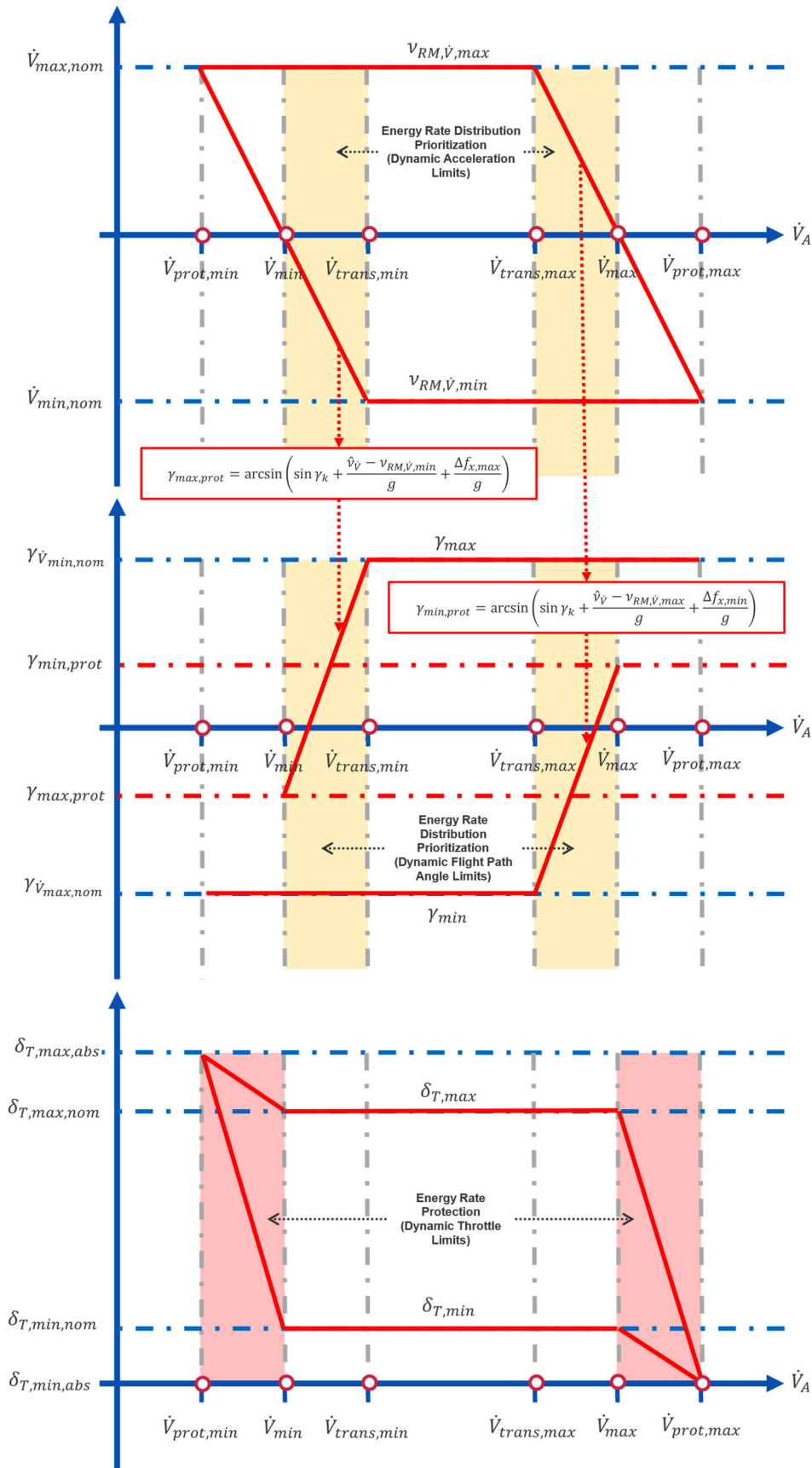


Figure 5.20: Acceleration, flight path angle, and throttle limits for energy integrity protection.

5.2.9 Force-Based Prioritizations

A curvature of the vertical ($\dot{\gamma}_K$) or lateral flight path ($\dot{\chi}_K$) is achieved by changing the magnitude of the total force perpendicular to the flight path in the vertical or lateral plane, respectively. As shown in Section 3.6, the limited available transverse force precludes an arbitrary curvature of the vertical and lateral flight path. For concurrent maneuvering in the vertical and lateral plane, the permissible transverse acceleration must be taken into account and, in case the transverse acceleration is saturated, a prioritization or trade-off between desired curvature in the vertical and lateral plane becomes necessary.

The specific force distribution is realized by cross-coupling the flight path angle and track angle reference models according to Figure 5.17, and setting the flight path angle rate and track angle rate pseudo-control command limiters, respectively [72, 74]. As in Section 3.6, for the formulation of the force prioritization constraints, the more intuitive load factors $n_{z,K}$, $n_{y,K}$ and $n_{z,\bar{K}}$ are used in place of the normal specific forces $f_{z,K}$, $f_{y,K}$ and $f_{z,\bar{K}}$, according to Eq. (3.108)-(3.109) and (3.114).

The magnitude of the commanded path transverse load factor $n_{z,\bar{K},C}$ for achieving the desired vertical and lateral path curvatures, respectively, must be equal to or less than the maximum permissible magnitude of the transverse load factor $n_{z,\bar{K},max}$,

$$n_{z,\bar{K},C} = \sqrt{(n_{z,K,C})^2 + (n_{y,K,C})^2} \leq n_{z,\bar{K},max}, \quad (5.112)$$

with

$$n_{z,K,C} = \frac{V_K}{g} v_{RM,\dot{\gamma}} + \cos \gamma_K, \quad (5.113)$$

$$n_{y,K,C} = \frac{V_K}{g} \cos \gamma_K v_{RM,\dot{\chi}}. \quad (5.114)$$

The maximum magnitude of the transverse specific force $n_{z,\bar{K},max}$ may be set at a fixed value for verification and validation purposes or implemented as a function of the maximum achievable lift coefficient or the maximum permissible structural load, as well as on the aircraft configuration (clean/flaps/landing), depending on the operational scenario and the maneuvering bandwidth needed.

Vertical vs Lateral Plane Maneuvering Prioritization

From the Eq. (3.115)-(3.116) and (3.119)-(3.120), the constraints on the desired flight path curvatures $v_{RM,\dot{\gamma}}$ and $v_{RM,\dot{\chi}}$ are given, in order to achieve the desired curvature in the prioritized plane. For prioritized maneuvering in the vertical plane, the constraints on the lateral path curvature commands are given by

$$\begin{aligned}
 v_{RM,\dot{\chi},max,vert} &= \frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - (n_{z,K,C})^2} \\
 &= \frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - \left(\frac{V_K}{g} v_{RM,\dot{\gamma}} - \cos \gamma_K\right)^2},
 \end{aligned} \tag{5.115}$$

$$\begin{aligned}
 v_{RM,\dot{\chi},min,vert} &= -\frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - (n_{z,K,C})^2} \\
 &= -\frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - \left(\frac{V_K}{g} v_{RM,\dot{\gamma}} - \cos \gamma_K\right)^2}.
 \end{aligned} \tag{5.116}$$

In order for the expression under the square root to be positive, the following constraints apply to the commanded vertical path curvature,

$$v_{RM,\dot{\gamma},max,vert} = \frac{g}{V_K} (n_{z,\bar{K},max} - \cos \gamma_K), \tag{5.117}$$

$$v_{RM,\dot{\gamma},min,vert} = \frac{g}{V_K} (-n_{z,\bar{K},max} - \cos \gamma_K). \tag{5.118}$$

As with Eq. (3.118), Eq. (5.118) represents the case when the aircraft is inverted and pulling with maximum load factor to achieve a negative vertical flight path curvature. For the development and verification of the AFCS as part of the modular FGCS, the integrated inner loop command interface, described in Section 5.2.10, does not foresee bank angles exceeding 90 degrees. Thus, the vertical flight path curvature is instead limited by the *minimum* load factor, i.e., a push-over maneuver, according to

$$v_{RM,\dot{\gamma},min,vert} = \frac{g}{V_K} (n_{z,\bar{K},min} - \cos \gamma_K). \tag{5.119}$$

Just as for the vertical plane, maneuvering in the lateral plane may be prioritized, either by allowing the full transverse load factor to produce a lateral path curvature (i.e. with a bank angle equal to 90 degrees), or by allowing the residual load factor when maintaining the current flight path angle for maneuvering in the lateral plane, thereby inhibiting further positive curvature of the vertical flight path. The constraint on the positive vertical path curvature is given by

$$\begin{aligned}
 v_{RM,\dot{\chi},max,lat} &= \frac{g}{V_K} \sqrt{(n_{z,\bar{K},max})^2 - (n_{y,K,C})^2} - \frac{g}{V_K} \cos \gamma_K \\
 &= \frac{g}{V_K} \sqrt{(n_{z,\bar{K},max})^2 - \left(\frac{V_K}{g} \cos \gamma_K v_{RM,\dot{\chi}}\right)^2} - \frac{g}{V_K} \cos \gamma_K.
 \end{aligned} \tag{5.120}$$

The constraints on the lateral curvature when utilizing full transverse force are given by

$$v_{RM,\dot{\chi},max,lat,full} = \frac{g}{V_K \cos \gamma_K} n_{z,\bar{K},max}, \tag{5.121}$$

$$v_{RM,\dot{\chi},min,lat,full} = -\frac{g}{V_K \cos \gamma_K} n_{z,\bar{K},max}. \quad (5.122)$$

When utilizing transverse force limited for lateral maneuvering, leaving a residual transverse force for maintaining the current climb angle, the constraints on the lateral curvature are given by

$$v_{RM,\dot{\chi},max,lat} = \frac{g}{V_K} \sqrt{\left(\frac{n_{z,\bar{K},max}}{\cos \gamma_K}\right)^2 - 1}, \quad (5.123)$$

$$v_{RM,\dot{\chi},min,lat} = -\frac{g}{V_K} \sqrt{\left(\frac{n_{z,\bar{K},max}}{\cos \gamma_K}\right)^2 - 1}. \quad (5.124)$$

The constraints in the vertical and lateral planes are illustrated in Figure 5.21.

Mixed Force Authority Prioritization

The previous section introduced the vertical versus lateral flight path curvature prioritization, where the transverse specific force distribution always is performed in favor of one of the two planes. In this section, an integration of the two path curvature modes of prioritization is presented, where the available transverse specific force, is budgeted between vertical and lateral flight path curvature.

Analog to the energy-based prioritizations, a force distribution factor $\kappa_F \in [0,1]$ is introduced, defining the amount of transverse force authority is allocated to vertical and lateral flight path curvature control. A value $\kappa_F = 0$ indicates full vertical flight path curvature priority and $\kappa_F = 1$ full lateral path curvature maneuver priority. A value in between results in a mixed priority. The permissible flight path curvature limits are then a function of κ_F . For the vertical plane, the permissible positive flight path curvature limit is determined by

$$v_{RM,\dot{\chi},max,prio} = \frac{g}{V_K} \sqrt{(n_{z,\bar{K},max})^2 - \left(\kappa_F \frac{V_K}{g} \cos \gamma_K \cdot v_{RM,\dot{\chi}}\right)^2} - \frac{g}{V_K} \cos \gamma_K. \quad (5.125)$$

For $\kappa_F = 0$, Eq. (5.125) reduces to Eq. (5.117). For $\kappa_F = 1$, Eq. (5.125) reduces to Eq. (5.120).

For maneuvering in the lateral plane, leaving a residual transverse force for maintaining the current climb angle, the permissible lateral flight path curvature limits are determined by

$$v_{RM,\dot{\chi},max,prio} = \frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - \left((1 - \kappa_F) \frac{V_K}{g} \cdot v_{RM,\dot{\chi}} + g \cos \gamma_K\right)^2}, \quad (5.126)$$

$$v_{RM,\dot{\chi},min,prio} = -\frac{g}{V_K \cos \gamma_K} \sqrt{(n_{z,\bar{K},max})^2 - \left((1 - \kappa_F) \frac{V_K}{g} \cdot v_{RM,\dot{\chi}} + g \cos \gamma_K\right)^2}. \quad (5.127)$$

certain plane but provide some residual maneuvering authority in the other plane. The priority can be set as a function of the aircraft state, with κ_F varying with for example altitude, speed, or aircraft configuration.

5.2.10 Inner Loop – Normal and Lateral Specific Force Control

This section describes the control of the normal and lateral forces, perpendicular to the aircraft flight path. The normal and lateral force control is performed by the inner loop controller, which is not the subject of the thesis at hand, see Chapter 1.2 and Figure 1.1. The content of this section is summarized from [85].

The inner loop controller utilizes only body-fixed measurements, thus a transformation between the kinematic flight path coming from the flight path controller, and the body-fixed commands entering the inner loop controller is necessary. Additional feed-forward elements are also described.

Command Selection and Transformation

The flight path controller outputs path curvature and acceleration commands for the inner loop and thrust controller. As control of the path curvature is allocated to the inner loop, only the lateral and vertical curvature commands, $\Delta f_{y,K,C}$ and $\Delta f_{z,K,C}$, respectively, are forwarded. These commands are transformed into a body-fixed frame bank angle command and a normalized specific force command.

First, the incremental specific force commands in the kinematic frame are transformed into the O -frame according to

$$\frac{1}{g} \begin{bmatrix} \Delta f_{x,C} \\ \Delta f_{y,C} \\ \Delta f_{z,C} \end{bmatrix}_O = \frac{1}{g} \mathbf{M}_{OK} \begin{bmatrix} \Delta f_{x,C} \\ \Delta f_{y,C} \\ \Delta f_{z,C} \end{bmatrix}_K. \quad (5.128)$$

The resulting command vector contains only the specific forces required for curvature, without the gravity component. Hence, the steady state normalized specific force component is added, which in the O -frame equals -1 along the z -axis,

$$\frac{1}{g} \begin{bmatrix} f_{x,C} \\ f_{y,C} \\ f_{z,C} \end{bmatrix}_O = \frac{1}{g} \begin{bmatrix} \Delta f_{x,C} \\ \Delta f_{y,C} \\ \Delta f_{z,C} \end{bmatrix}_O - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_O. \quad (5.129)$$

The full specific force command vector is then transformed into the intermediate body-fixed system Z ,

$$\frac{1}{g} \begin{bmatrix} f_{x,C} \\ f_{y,C} \\ f_{z,C} \end{bmatrix}_Z = \frac{1}{g} \mathbf{M}_{ZO} \begin{bmatrix} f_{x,C} \\ f_{y,C} \\ f_{z,C} \end{bmatrix}_O. \quad (5.130)$$

The intermediate body-fixed system Z is equal to the standard body-fixed frame, excluding the rotation around the x_B -axis, i.e., the bank angle. The inner loop achieves the path curvature by increasing or decreasing the lift force magnitude, i.e., the z_B -component, and adjusting its

direction, i.e., the bank angle Φ . These can be obtained from the Z -frame components by trigonometric relations. Consider the following transformation between the intermediate body-fixed (Z) and the body-fixed (B) system,

$$\frac{1}{g} \begin{bmatrix} f_{x,C} \\ f_{y,C} \\ f_{z,C} \end{bmatrix}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \frac{1}{g} \begin{bmatrix} f_{x,C} \\ f_{y,C} \\ f_{z,C} \end{bmatrix}_Z. \quad (5.131)$$

The x -axis is not controlled by the inner loop since aerodynamic speed control is used. From the second and third rows, the following equations for the normal and lateral specific force commands can be obtained,

$$\frac{f_{y,B,C}}{g} = \frac{f_{y,Z,C}}{g} \cos \Phi + \frac{f_{z,Z,C}}{g} \sin \Phi, \quad (5.132)$$

$$\frac{f_{z,B,C}}{g} = -\frac{f_{y,Z,C}}{g} \sin \Phi + \frac{f_{z,Z,C}}{g} \cos \Phi. \quad (5.133)$$

In order to fly a coordinated curve, i.e. $f_{y,B} = 0$, the required bank angle for achieving the lateral force command can be calculated from Eq.(5.132) by setting $f_{y,B,C} = 0$ and solving for Φ ,

$$\Phi_C = \tan^{-1} \left(-\frac{f_{y,Z,C}}{f_{z,Z,C}} \right). \quad (5.134)$$

In order to obtain the body-fixed specific force command $f_{z,B,C}/g$, Eq. (5.134) rewritten as $f_{y,Z,C} = -f_{z,Z,C} \tan \Phi_C$ can be inserted into Eq. (5.133), which gives

$$\begin{aligned} \frac{f_{z,B,C}}{g} &= - \left(-\frac{f_{z,Z,C}}{g} \tan \Phi_C \right) \sin \Phi_C + \frac{f_{z,Z,C}}{g} \cos \Phi_C \\ &= \frac{f_{z,Z,C}}{g} \left(\frac{\sin \Phi_C}{\cos \Phi_C} \sin \Phi_C + \cos \Phi_C \right) \\ &= \frac{f_{z,Z,C}}{g} \frac{1}{\cos \Phi_C}. \end{aligned} \quad (5.135)$$

A normal specific force command calculated according to Eq. (5.135), based on a simultaneous bank angle command according to Eq. (5.134), would lead to a velodrome-shaped curve, since the pitch, i.e. $f_{z,B}$ dynamics is normally much faster than the bank angle dynamics. The aircraft would reach the required normal specific force for curve compensation, as calculated based on the commanded bank angle, before the corresponding bank angle has been built up, producing to a "pull-up then bank" behavior. This phenomenon is in the inner loop command interface mitigated by calculating the normal specific force command based on the actual bank angle instead of the commanded, i.e.

$$\frac{f_{z,B,C}}{g} = \frac{f_{z,Z,C}}{g} \frac{1}{\cos \Phi}. \quad (5.136)$$

However, this results in a small but acceptable delay in the normal specific force response. An alternative would be to use a modified bank angle command $\tilde{\Phi}_C$ for the calculation of the normal specific force command, e.g., according to

$$\tilde{\Phi}_C = \Phi + \Delta T \cdot p, \quad (5.137)$$

where T is some time constant and p the current roll rate, or according to some function of the difference between the bank angle command according to Eq. (5.134) and the measured bank angle,

$$\tilde{\Phi}_C = f(\Phi_C - \Phi). \quad (5.138)$$

The inner loop command transformation geometry is illustrated in Figure 5.22.

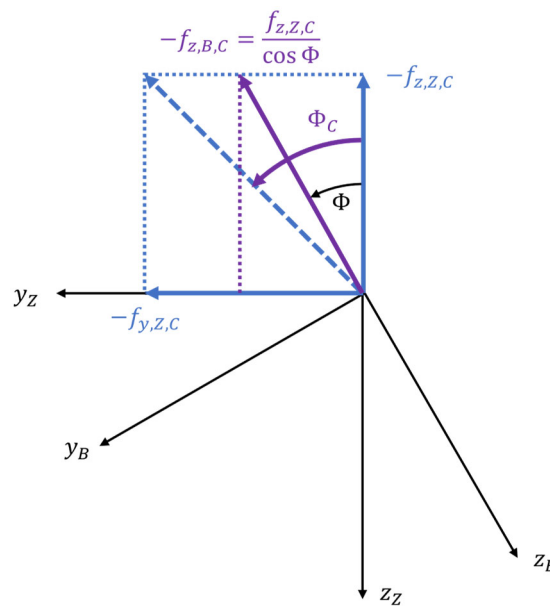


Figure 5.22: Body axes specific force and bank angle command generation.

Rate Feedforward Commands

The bank angle and specific force commands are amended by body-fixed frame rate commands, in order to improve the inner loop control performance, thus allowing for enhanced path control performance.

The rotation of the body-fixed frame with respect to the kinematic frame is given by

$$(\vec{\omega}^{KB})_B = (\vec{\omega}^{KO})_B + (\vec{\omega}^{OB})_B, \quad (5.139)$$

giving for the body-fixed rates relative to the O -frame

$$(\vec{\omega}^{OB})_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_B = (\vec{\omega}^{KB})_B - (\vec{\omega}^{KO})_B. \quad (5.140)$$

The rotation of the kinematic frame relative to the O -frame, i.e., the kinematic rates, are given by

$$(\vec{\omega}^{KO})_K = \begin{bmatrix} \dot{\chi}_K \\ -\dot{\gamma}_K \\ -\dot{\chi}_K \cos \gamma_K \end{bmatrix}_K. \quad (5.141)$$

The kinematic rates given in body-fixed coordinates can be obtained using the transformation matrices \mathbf{M}_{BO} , Eq (D.11), and \mathbf{M}_{OK} , Eq. (D.3),

$$\begin{aligned} (\vec{\omega}^{KO})_B &= \mathbf{M}_{BO}(\Phi, \Theta, \Psi) \mathbf{M}_{OK}(\chi_K, \gamma_K) (\vec{\omega}^{KO})_K \\ &= \dot{\chi}_K \begin{bmatrix} \sin \Theta \\ \cos \Theta \sin \Phi \\ \cos \Theta \cos \Phi \end{bmatrix}_K - \dot{\gamma}_K \begin{bmatrix} \sin(\Psi - \chi_K) \cos \Theta \\ \cos(\Psi - \chi_K) \cos \Phi + \sin(\Psi - \chi_K) \sin \Theta \sin \Phi \\ -\cos(\Psi - \chi_K) \sin \Phi + \sin(\Psi - \chi_K) \sin \Theta \cos \Phi \end{bmatrix}_K. \end{aligned} \quad (5.142)$$

The first term in Eq. (5.142) denotes the rates corresponding to a turn, and the second term is related to the rates required for a pull-up. Together with the rates of the body-fixed relative to the kinematic frame, given by $(\vec{\omega}^{KB})_B$, Eq. (D.16), corresponding to the dynamic changes of the kinematic angles of attack, α_K , sideslip, β_K , and bank, μ_K , the total rate feedforward can be divided as

$$(\vec{\omega}^{OB})_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_B = \begin{bmatrix} p_{turn} \\ q_{turn} \\ r_{turn} \end{bmatrix} + \begin{bmatrix} p_{pull} \\ q_{pull} \\ r_{pull} \end{bmatrix} + \begin{bmatrix} p_{dyn} \\ q_{dyn} \\ r_{dyn} \end{bmatrix}, \quad (5.143)$$

with

$$\begin{bmatrix} p_{turn} \\ q_{turn} \\ r_{turn} \end{bmatrix} = \dot{\chi}_{K,C} \begin{bmatrix} \sin \Theta \\ \cos \Theta \sin \Phi \\ \cos \Theta \cos \Phi \end{bmatrix}, \quad (5.144)$$

$$\begin{bmatrix} p_{pull} \\ q_{pull} \\ r_{pull} \end{bmatrix} = \dot{\gamma}_{K,C} \begin{bmatrix} \sin(\Psi - \chi_K) \cos \Theta \\ \cos(\Psi - \chi_K) \cos \Phi + \sin(\Psi - \chi_K) \sin \Theta \sin \Phi \\ -\cos(\Psi - \chi_K) \sin \Phi + \sin(\Psi - \chi_K) \sin \Theta \cos \Phi \end{bmatrix}, \quad (5.145)$$

and

$$\begin{bmatrix} p_{dyn} \\ q_{dyn} \\ r_{dyn} \end{bmatrix} = \begin{bmatrix} \dot{\mu}_K \cos \alpha_K \cos \beta_K + \dot{\beta}_K \sin \alpha_K \\ \dot{\alpha}_K - \dot{\mu}_K \sin \beta_K \\ \dot{\mu}_K \sin \alpha_K \cos \beta_K - \dot{\beta}_K \cos \alpha_K \end{bmatrix}. \quad (5.146)$$

The rate feedforward command block calculates the steady state turn and pull up rates, Eq. (5.144) to (5.145), based on the desired lateral and vertical specific forces according,

$$\dot{\chi}_{K,C} = \frac{1}{V_K \cos \gamma_K} \cdot \Delta f_{y,K,C}, \quad (5.147)$$

$$\dot{\gamma}_{K,C} = -\frac{1}{V_K} \Delta f_{z,K,C}, \quad (5.148)$$

and omits the dynamic rates, Eq. (5.146).

Normal Specific Force and Bank Angle Control

The design of the inner loop controller for the normal specific force and bank angle control is highly specific to the application aircraft. For the different demonstration platforms described in Section 1.4, different inner loop controllers were developed at TUM-FSD. For the conceptual development of the modular FGCS for the DA42 OE-FSD demonstration platform, the longitudinal inner loop controller employs the normal body specific force and the pitch rate as feedback for a PI controller [85]. The lateral inner loop controller is a MIMO control structure designed to track the bank angle command Φ_C and reduce the body lateral acceleration $f_{y,B}$ to zero. For this purpose, the roll and pitch rates are fed back along with the lateral acceleration and the bank angle command. The inner loop for the Do-228 D-CODE is based on the inner loop for the DA42, with aircraft-specific parametrization.

The longitudinal and lateral inner loop for the SAGITTA Research Demonstrator utilizes a different controller design due to the unconventional geometry and control surface design. The inner loop controller for the SAGITTA Research Demonstrator is described in [86].

5.2.11 Autothrust – Longitudinal Acceleration Control

As for the inner loop controlling the body axis specific force and bank angle, an automatic thrust controller controlling the acceleration along the flight path is designed around the magnitude and dynamics of the propulsion system, which is dependent on the type of propulsion, e.g., jet engine, turbofan, turboprop, or electric propulsion.

For the development and validation of the AFCS concept, due to a lack of an underlying model of the engine dynamics, a simplified first-order model according to Eq. (3.88) is the basis for the design of a basic autothrust controller, with the maximum thrust available $T_{max}(V_A, h)$ and the current thrust T determined by the normalized throttle position $\delta_T \in [0,1]$ according to $T = \delta_T \cdot T_{max}$.

Autothrust in Speed by Thrust Mode

In speed by thrust mode, the required change in throttle position $\Delta\delta_{T,C}$ to achieve a desired incremental specific force $\Delta f_{x,K,C}$ is calculated according to

$$\Delta\delta_{T,C} = \frac{1}{\frac{\Delta f_{x,K}}{\Delta\delta_T}} \cdot (\Delta f_{x,K,C} - \Delta f_{x,K}) = \frac{1}{\frac{\Delta f_{x,K}}{\Delta\delta_T}} \cdot (v_{\hat{V}_K} - \hat{v}_{\hat{V}_K}) \quad (5.149)$$

where $v_{\hat{V}}$ is the pseudo control acceleration command from the speed loop, and $\hat{v}_{\hat{V}}$ is the estimated acceleration according to Eq. (5.20). The ratio of the incremental specific force per throttle increment $\Delta f_{x,K}/\Delta\delta_T$ is calculated from the maximum thrust available T_{max} and the current thrust T , which for the simplified engine model in Eq. (3.88) is given by

$$\frac{\Delta f_{x,K}}{\Delta\delta_T} = \frac{1}{m} \cdot \frac{T_{max} - T}{\delta_{T,max} - \delta_T} = \frac{1}{m} \cdot \frac{T_{max}(1 - \delta_T)}{1 - \delta_T} = \frac{T_{max}}{m} = \Delta f_{x,K,max}. \quad (5.150)$$

Thus, Eq. (5.149) is reduced to

$$\Delta\delta_{T,C} = \frac{1}{\Delta f_{x,K,max}} \cdot (\Delta f_{x,K,C} - \Delta f_{x,K}). \quad (5.151)$$

The normalized throttle command $\delta_{T,C}$ is calculated from the estimated required throttle increment according to a PI control structure given by

$$\begin{aligned} \delta_{T,C} &= \Delta\delta_{T,P,C} + \Delta\delta_{T,I,C} \\ &= k_{P,\delta_T} \Delta\delta_{T,C} + \int k_{I,\delta_T} \Delta\delta_{T,C} dt. \end{aligned} \quad (5.152)$$

The basic autothrust control structure is illustrated in Figure 5.23. The proportional and integral gains are selected based on a desired closed-loop thrust control bandwidth and the propulsion system time constant T_{δ_T} .

The initial condition of the incremental throttle command integrator is set to the trim throttle setting if the speed by thrust function is activated and there is no transition from a speed by pitch mode (e.g., during the initial initialization of the controller). Otherwise, the current throttle command is taken as the initial condition.

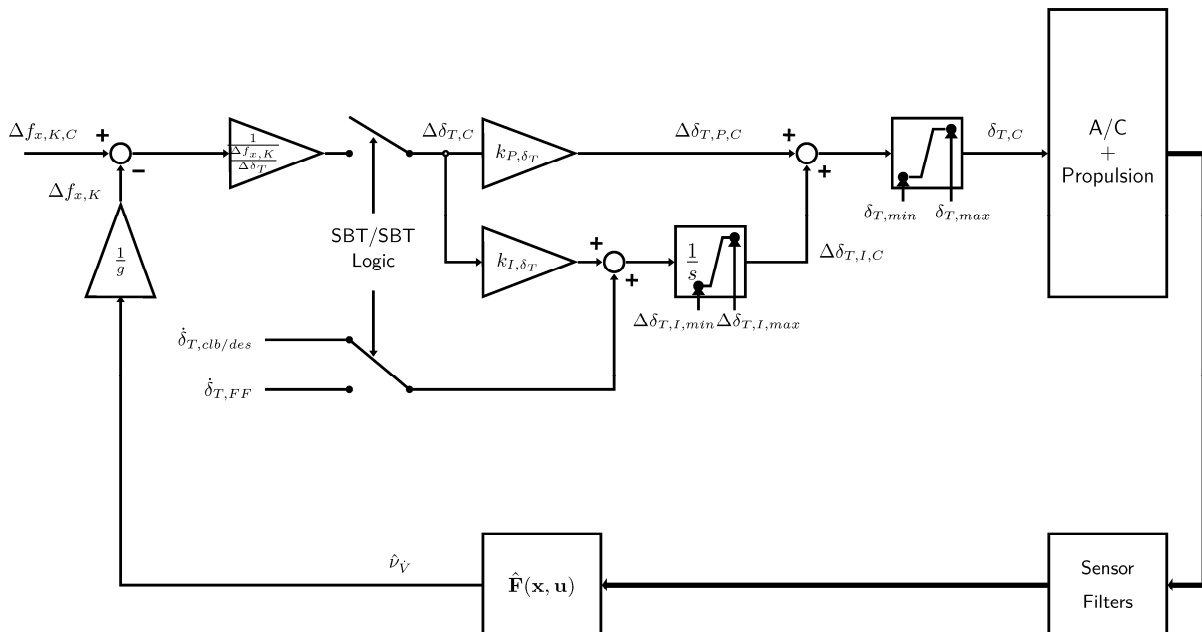


Figure 5.23: Autothrust controller structure.

The integrator part ensures static accuracy for the thrust control. The integrator further allows direct feedforward throttle rate commands, added to the integrator input. This can be used to compensate for the additional required thrust during maneuvering in the vertical plane, as well as for fixed thrust modes such as open climb and descent.

The acceleration along the flight path is a function of the linear specific force and the flight path angle, $\dot{V}_K = f_{x,K} - g \sin \gamma_K$. For the speed to remain unchanged during a maneuver in the vertical plane, i.e.

$$\dot{V}_K = 0 \Rightarrow f_{x,K} = g \sin \gamma_K, \quad (5.153)$$

the relation between the rate of change of the specific force along the flight path $\dot{f}_{x,K}$ and the flight path curvature $\dot{\gamma}_K$, and in turn the normal incremental specific force $\Delta f_{z,K}$, is given by

$$\begin{aligned} \dot{f}_{x,K} &= g \cos \gamma_K \cdot \dot{\gamma}_K \\ &= g \cos \gamma_K \cdot \left(-\frac{1}{V_K} \cdot \Delta f_{z,K} \right). \end{aligned} \quad (5.154)$$

From (5.154) a feedforward normal specific force rate command, and thereby a feedforward throttle rate command $\dot{\delta}_{T,FF}$, can be derived, that is fed directly to the thrust loop integrator,

$$\begin{aligned} \dot{\delta}_{T,FF} &= \frac{1}{\Delta f_{x,K,max}} \cdot m \cdot \dot{f}_{x,K,C} \\ &= \frac{1}{\Delta f_{x,K,max}} \cdot m \cdot g \cos \gamma_K \cdot \left(-\frac{1}{V_K} \cdot \Delta f_{z,K,C} \right) \end{aligned} \quad (5.155)$$

Autothrust in Speed by Pitch / Open Climb/Descent Mode

In speed by pitch mode, the autothrust provides a fixed normalized throttle setting. If an altitude target is selected, the speed by pitch is combined with a throttle setting corresponding to climb or descent thrust, $\delta_{T,CLB}$ or $\delta_{T,DES}$. The throttle command is achieved by adjusting the upper or lower limits of the integrator to the desired climb or descent throttle setting and transitioning the throttle command to that limit at a fixed climb or descent throttle rate, $\dot{\delta}_{T,CLB}$ or $\dot{\delta}_{T,DES}$. If in an open climb, the upper limit of the integrator is set to the climb thrust setting, allowing the positive throttle rate command to advance the throttle to this setting but not beyond. If in an open descent, the lower limit of the integrator is set to the descent thrust setting, allowing the negative throttle rate command to reduce the throttle to this setting but not beyond. If no climb or descent is commanded, the corresponding throttle rate is set to zero.

For the fixed rate throttle modes (climb and descent thrust), the fixed throttle rate commanded to the integrator is known, and thus the resulting excess energy rate that is to be converted into a flight path angle to realize the desired climb or descent. As the acceleration rate is known, the corresponding flight path curvature can be approximated from the same relations as in Eq. (5.155), and converted to a feedforward incremental normal specific force command to the vertical path control and incremental vertical load factor command,

$$\Delta f_{z,K,C,CLB/DES} = -\frac{V_K \cdot \Delta f_{x,K,max}}{m \cdot g \cos \gamma_K} \cdot \dot{\delta}_{T,CLB/DES}. \quad (5.156)$$

5.2.12 Direct Path Curvature Control

The controller structure further allows for an external guidance system to directly command a curvature of the vertical or lateral plane, i.e., $\dot{\gamma}_C$ and $\dot{\chi}_C$. The path curvature commands are converted to the corresponding specific forces in the kinematic frame, analog to the pseudo control vector, Eq. (5.7),

$$\Delta f_{y,K,C} = V_K \cdot \cos \gamma_K \cdot \dot{\chi}_C \quad (5.157)$$

$$\Delta f_{z,K,C} = -V_K \cdot \dot{\gamma}_C \quad (5.158)$$

The specific force commands are then fed to the command transformation described in Section 5.2.10.

To protect the energy integrity of the aircraft also in the direct path curvature command mode, the flight path angle limitations calculated in Section 5.2.8 are enforced, by imposing a linear reduction of the permissible vertical flight path curvature, so that at the maximum and minimum flight path angles, the allowed path curvature is equal to zero, according to

$$\dot{\gamma}_{C,max} = k_{\dot{\gamma}_C}(\gamma_{max} - \gamma_K) \quad (5.159)$$

$$\dot{\gamma}_{C,min} = k_{\dot{\gamma}_C}(\gamma_{min} - \gamma_K) \quad (5.160)$$

The direct path curvature control mode has been primarily used for manual control of the flight path curvature, by mapping a passive stick deflection to a specific curvature range, a so-called “direct path curvature stick mode.” The basic controller structure for this mode is visualized in Figure 5.24.

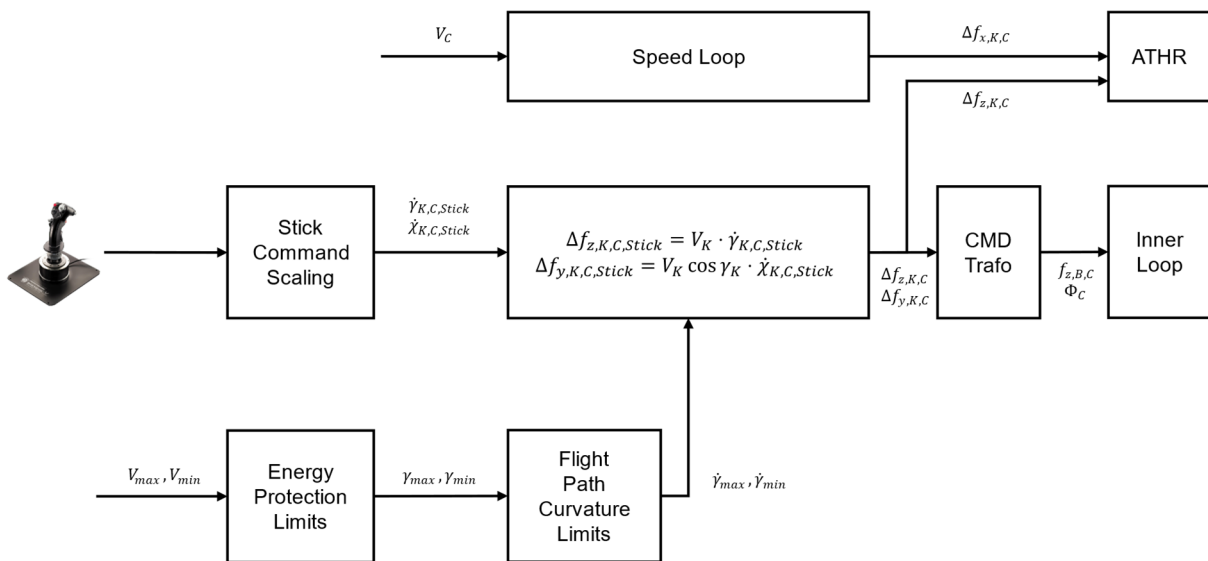


Figure 5.24: Direct path curvature stick mode with limitations for energy protection.

5.3 Gain and Parameter Design

This section describes the design strategy and framework for the AFCS gain and parameters of the control laws and functional elements described in Section 5.2.

5.3.1 Gain and Parameter Design Preliminary Considerations

The strategy for the AFCS gain and parameter design depends on the application scenario. For cases where the plant and closed inner loop dynamics are well known, gain and parameter design may be optimized for performance, with gains and parameters scheduled over the operational speed and altitude envelopes, as well as aircraft configuration (gear and flaps settings), in order to fully utilize the available plant dynamics. For cases where the plant and inner loop dynamics are less known, the gain and parameter design may be optimized for robustness against plant uncertainties, perhaps using fixed gains. For the demonstration platform for the main results in this thesis, the DA42 OE-FSD, the plant and closed inner loop dynamics are rather well known. Thus, a gain and parameter scheduling approach is chosen that corresponds to the scheduling of the inner loop gains, i.e., over static and dynamic pressure, as well as aircraft gear and flap configuration settings. Robustness against uncertainties in the closed inner loop dynamics and signal processing is provided by assuring adequate gain and phase margin stability criteria for the closed path control loops, including the pseudo-control hedging feedback structure, as well as actuation and sensor delays.

The gain and parameter design framework shall also manage different application scenarios through easy adaptation to the set of control modes, configuration of parameters, and verification of the set of functionalities without extensive retesting, in line with research objective 1, *a modular control system architecture for configurability and testability*, see Section 1.3.2. For the gain and parameter design framework, this results in the following design objectives:

- Maintain a configuration generic setup with generic routines and data structures, but with the necessity to manage configuration-specific elements and input data on different forms for multiple platforms
- The framework must include the full set of control modes but with the ability to configure the controller modes for design and test of the relevant subsets based on the configuration-specific input data and settings
- High degree of automation and computational effectiveness, to allow for quick development iterations with the concurrent inner loop and other control module developments

The gain and parameter design framework makes use of linear representations of the controller elements as presented in Section 5.2, implemented as MATLAB models. The design of the reference model dynamics and error controller gains is based on linear models of the closed inner loop and plant dynamics, and looped over the scheduling grid of the inner loop, with overshoot and settling time requirements as design criteria according to Section 5.1.3, and gain and phase margin requirements at the actuators as monitoring criteria.

5.3.2 Controller Design and Assessment Framework

Tunable Model Structures

The design and assessment framework consists of a suite of MATLAB scripts for automated controller gain and parameter design, as well as linear assessment of the longitudinal and lateral control systems. The script suite generates a linear representation of the AFCS in the form of generalized state space models (genss) with tunable gains and parameters, using the MATLAB Control System Toolbox.

The model structure for the vertical plane is illustrated in Figure 5.25 and the model structure for the lateral plane in Figure 5.26. The model structure for the vertical plane does not include the trajectory controller module, as this module and its gains and parameters are designed separately, see Section 5.2.6. Thus, the altitude hold mode and the vertical navigation mode are omitted in this section, as these are implemented by the trajectory controller. The radio navigation and approach modes have been prototypically implemented as part of the design model, with their nominal functionality demonstrated in Model- and Hardware-in-the-Loop simulations but are not included in the formal design and assessment framework presented here.

The pitch hold mode structure seen in Figure 5.25 is an application-specific basic control mode implemented for the DA42 OE-FSD demonstration platform, and not part of the set of application-generic AFCF path/trajectory functions defined in the system functions, Section 5.1.2, and is thus not further elaborated in this section.

The generalized state space models include loop switches that allow for the configuration of control modes (for example by switching between speed by thrust and speed by pitch modes), or to open loops for gain and phase margin analysis. The loop switches are illustrated in Figure 5.25 and Figure 5.26 as red boxes. The tunable models allow for an efficient gain and parameter design, as the complete model is only needed to be built once for each design point, with the individual design scripts for specific modes configuring the switches for the loop under design and assigning the designed mode-specific parameter values to the model structure, without the need to rebuild blocks around parameters subject to design.

AFCS MATLAB Linear Model Structure

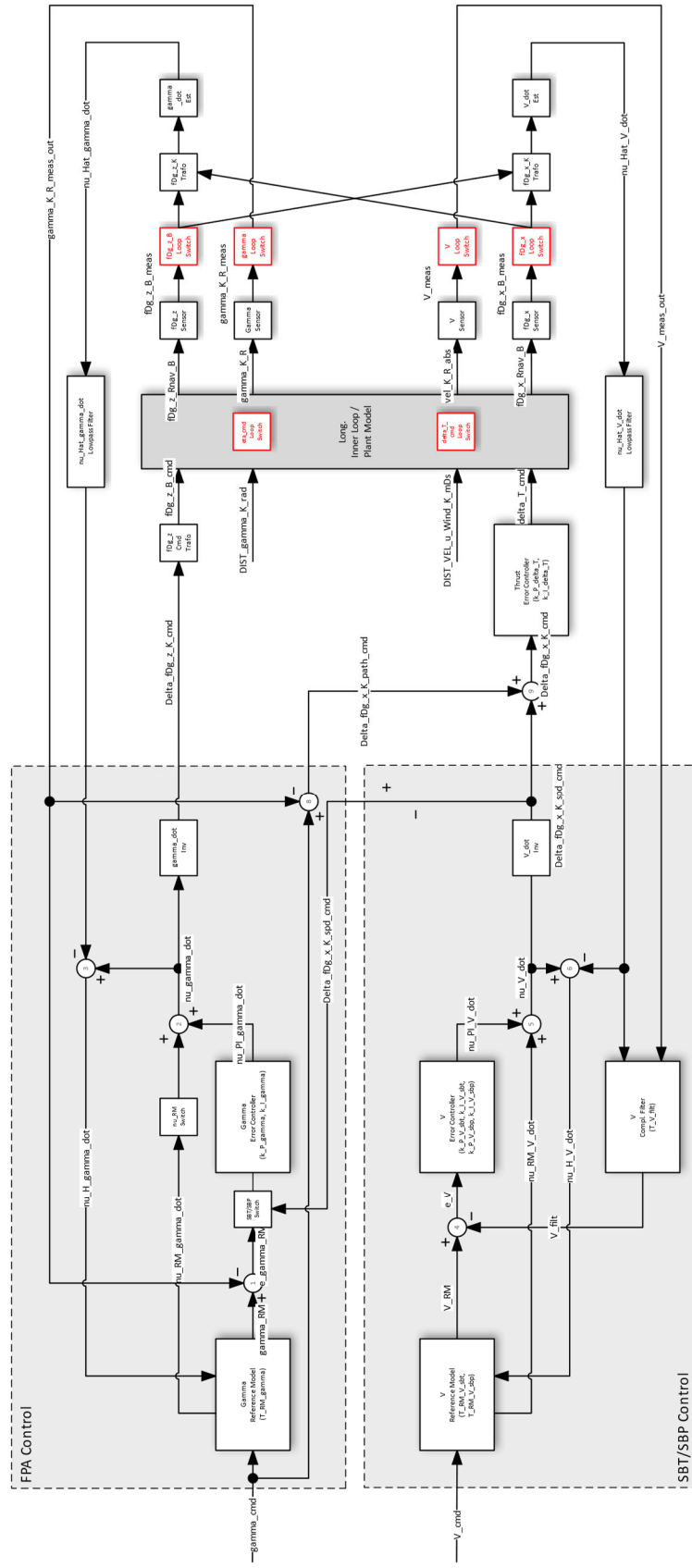


Figure 5.25: Linear MATLAB model structure for vertical plane design and assessment, including closed plant and inner loop dynamics.

AFCs MATLAB Linear Model Structure

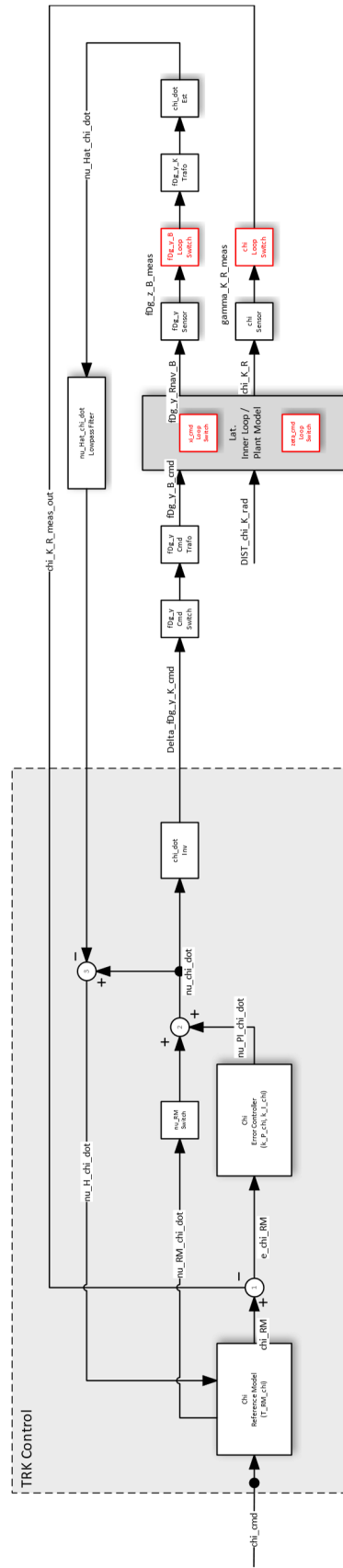


Figure 5.26: Linear MATLAB model structure for lateral plane design and assessment, including closed plant and inner loop dynamics.

Linearization of Controller Elements

The path dynamics inversion together with the plant response estimation form a set of input-output linearized SISO systems, i.e., the transfer functions from the pseudo control v to the derivative of the plant state \dot{y} . For the design of desired input-output dynamics of the closed path loops, the path dynamics inversion and the plant response estimation may thus be omitted, and the design of the dynamics performed for a model structure of nominally linear elements, i.e. the reference model, the error controller, a linearized closed inner loop dynamics for the kinematic frame specific force dynamics, as well as the integrator chain as plant.

The plant response estimation and inner loop command transformation, however, are performed based on measurements of the body axis specific forces, and the linearized closed inner loops represent the body axis dynamics. Thus, the model structures for the lateral and vertical planes include linearized versions of the path dynamics inversion, the path dynamics response estimation, as well as the inner loop command transformation from the kinematic to the body-axis frames. This is done to capture the effects of sensor measurement delays and noise. The inclusion of linearized versions of the path dynamics inversion also produces a specific force command to the inner loop of correct magnitude, which is not the case for the kinematic linear model representation discussed above.

In general, a linearization may be performed for multiple steady-state, quasi-steady-state or transient reference flight conditions. Depending on the application scenario, desired envelope, and mission profile, a set of linearized flight conditions underlie the inner loop gain design and scheduling approach. For the conceptual development of the modular FGCS including the inner loop and the AFCS, reference flight conditions are limited to straight and level flight over the envelope.

The inversion of the path dynamics, Eq. (5.7), is linearized as

$$\delta\Delta f_{x,K,C} = \delta v_{\dot{V}_K}, \quad (5.161)$$

$$\begin{aligned} \delta\Delta f_{y,K,C} = & V_{K,0} \cos \gamma_{K,0} \cdot \delta v_{\dot{\chi}_K} + \cos \gamma_{K,0} v_{\dot{\chi}_{K,0}} \cdot \delta V_K \\ & - V_{K,0} \cos \gamma_{K,0} v_{\dot{\chi}_{K,0}} \cdot \delta \gamma_K \end{aligned} \quad (5.162)$$

$$\Delta f_{z,K,C} = V_{K,0} \cdot \delta v_{\dot{\gamma}_K} - v_{\dot{\gamma}_{K,0}} \cdot \delta V_K \quad (5.163)$$

With the simplification of steady-state incremental specific forces in the kinematic frame equal to zero, i.e., steady-state path curvatures and path acceleration equal to zero,

$$\Delta f_{x,K,0} = 0 \Rightarrow v_{\dot{V}_{K,0}} = 0, \quad (5.164)$$

$$\Delta f_{y,K,0} = 0 \Rightarrow v_{\dot{\chi}_{K,0}} = 0, \quad (5.165)$$

$$\Delta f_{z,K,0} = 0 \Rightarrow v_{\dot{\gamma}_{K,0}} = 0, \quad (5.166)$$

the path curvature commands are analyzed with respect to the steady-state flight path angle $\gamma_{K,0}$ and track angle $\chi_{K,0}$, with the linearized inversion reduced to

$$\delta \Delta f_{x,K,C} = \delta v_{\dot{V}_K}, \quad (5.167)$$

$$\delta \Delta f_{y,K,C} = V_{K,0} \cos \gamma_{K,0} \cdot \delta v_{\dot{\chi}_K}, \quad (5.168)$$

$$\delta \Delta f_{z,K,C} = -V_{K,0} \cdot \delta v_{\dot{\gamma}_K}. \quad (5.169)$$

For horizontal, wings level flight, Eq. (5.168) is further reduced to

$$\delta \Delta f_{y,K,C} = V_{K,0} \cdot \delta v_{\dot{\chi}_K}. \quad (5.170)$$

The path dynamics response is estimated from the measured specific forces in the body axis frame including gravity, transformed into the kinematic frame, according to Eq. (5.16)-(5.22). The path dynamics response in the kinematic frame is linearized as

$$\delta \hat{v}_{\dot{V}_K} = \delta f_{x,K} - g \cos \gamma_{K,0} \cdot \delta \gamma_K, \quad (5.171)$$

$$\delta \hat{v}_{\dot{\chi}_K} = V_{K,0} \cos \gamma_{K,0} \cdot \delta f_{y,K} + f_{y,K,0} \cos \gamma_{K,0} \cdot \delta V_K - f_{yK,0} V_{K,0} \sin \gamma_{K,0} \cdot \delta \gamma_K, \quad (5.172)$$

$$\delta \hat{v}_{\dot{\gamma}_K} = -V_{K,0} \cdot \delta f_{z,K} - (f_{z,K,0} + g \cos \gamma_{K,0}) \cdot \delta V_K + V_{K,0} g \sin \gamma_{K,0} \cdot \delta \gamma_K. \quad (5.173)$$

Assuming steady-state path curvatures and path acceleration equal to zero, Eq. (5.164)-(5.166), as well as no bank angle or lateral force, i.e.

$$\Delta f_{y,B,0} = 0, \quad (5.174)$$

$$\Phi_0 = 0, \quad (5.175)$$

the steady-state specific forces in the kinematic frame are given by

$$f_{x,K,0} = f_{x,B,0} \cos(\Theta_0 - \gamma_{K,0}) + f_{z,B,0} \sin(\Theta_0 - \gamma_{K,0}), \quad (5.176)$$

$$f_{y,K,0} = 0, \quad (5.177)$$

$$f_{z,K,0} = -f_{x,B,0} \sin(\Theta_0 - \gamma_{K,0}) + f_{z,B,0} \cos(\Theta_0 - \gamma_{K,0}). \quad (5.178)$$

The incremental specific forces in the kinematic frame are given by

$$\begin{aligned} \delta f_{x,K} &= \cos(\Theta_0 - \gamma_{K,0}) \cdot \delta f_{x,B} + \sin(\Theta_0 - \gamma_{K,0}) \cdot \delta f_{z,B} \\ &\quad + (f_{x,B,0} \sin(\Theta_0 - \gamma_{K,0}) + f_{x,B,0} \cos(\Theta_0 - \gamma_{K,0})) \cdot \delta \Theta \\ &\quad - (f_{x,B,0} \sin(\Theta_0 - \gamma_{K,0}) + f_{x,B,0} \cos(\Theta_0 - \gamma_{K,0})) \cdot \delta \gamma_K \end{aligned} \quad (5.179)$$

$$\delta f_{y,K} = \delta f_{y,B} - f_{z,B,0} \cdot \delta \Phi, \quad (5.180)$$

$$\begin{aligned} \delta f_{z,K} &= -\sin(\Theta_0 - \gamma_{K,0}) \cdot \delta f_{x,B} + \cos(\Theta_0 - \gamma_{K,0}) \cdot \delta f_{z,B} \\ &\quad - (f_{x,B,0} \cos(\Theta_0 - \gamma_{K,0}) + f_{x,B,0} \sin(\Theta_0 - \gamma_{K,0})) \cdot \delta \Theta \\ &\quad - (f_{x,B,0} \cos(\Theta_0 - \gamma_{K,0}) + f_{x,B,0} \sin(\Theta_0 - \gamma_{K,0})) \cdot \delta \gamma_K. \end{aligned} \quad (5.181)$$

For horizontal flight, i.e. $\gamma_{K,0} = 0$, Eq. (5.171)-(5.173) are reduced further to

$$\delta \hat{v}_{\dot{V}_K} = \delta f_{x,K} - g \cdot \delta \gamma_K, \quad (5.182)$$

$$\delta \hat{v}_{\dot{\chi}_K} = V_{K,0} \cdot \delta f_{y,K} + f_{y,K,0} \cdot \delta V_K, \quad (5.183)$$

$$\delta \hat{v}_{\dot{Y}_K} = -V_{K,0} \cdot \delta f_{z,K} - (f_{z,K,0} + g) \cdot \delta V_K, \quad (5.184)$$

as well as similarly for Eq. (5.174)-(5.181).

Main Design Routine

A generic main design routine, illustrated in Figure 5.27, takes the desired aircraft identification and desired control plane for design (vertical plane, lateral plane of both) as inputs. The main design routine is then sequenced as follows:

- The design is initialized by getting the path to the applicable directory with aircraft-specific input design data and creating a corresponding output directory for the output design data.
- For the desired control plane, the routine loads requirements data, as well as the closed inner loop and plant dynamics as a structure with models over the envelope, and builds a plant model for the AFCS design with generic input and output names.
- The routine then builds the AFCS model, based on the desired set of control modes defined in the requirements set, and populates the AFCS model with parameters such as limits according to the requirements and a default set of tunable gains.
- For each of the desired control modes, separate design scripts take the AFCS and plant models as well as requirements as inputs, perform the gain design, and output a tuned AFCS model for that specific control mode.
- When all the desired modes are successfully designed, the gain set and design results such as design metadata, tuned models, step response data, gain and phase margin data, are stored in the output directory.
- If desired, an automated design report is generated with time and frequency domain analysis documented.

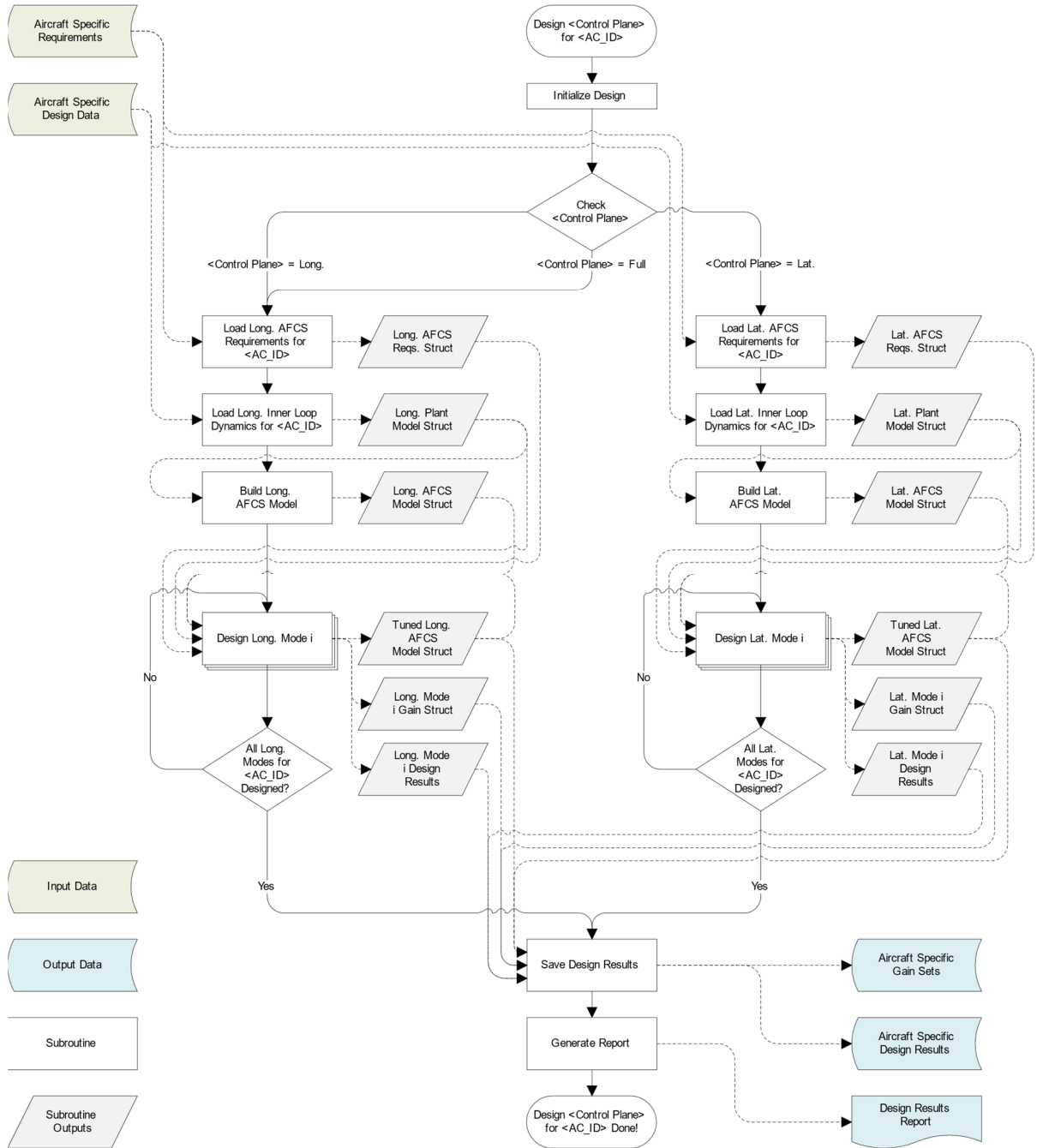


Figure 5.27: Main gain and parameter design routine for longitudinal/lateral AFCS.

5.3.3 Path Control Parameters

The gains and parameters for the path control include the reference model time constants, the error controller proportional, and integral gains.

Reference Model Time Constants

The reference model time constants are the parameters that influence the desired behavior of the path loops the most. The time constants $T_{RM,1}$ and $T_{RM,2}$ are chosen such that a slower pole dominates the reference dynamics, and a faster pole is added to smoothen the pseudo control. A comparison of first- and second-order reference model dynamics in terms of step responses, Bode gain and phase plots for the transfer functions from path command to path state and pseudo control, respectively, is given in Figure 5.13.

Figure 5.28 illustrates the effect of an increase of the dominating reference model time constant $T_{RM,1}$ while the faster time constant is kept constant. The overshoot increases and the rise time decreases with decreasing time constant. The settling time is affected negatively by a too-small or too-large time constant. A smaller time constant leads to an oscillating step response that takes additional time to settle, whereas a larger time constant simply slows down the step response. The Bode diagram shows a decrease in maximum magnitude for an increasing time constant but a reduction in phase reserve (the phase curve drops).

The objective for the design of the reference model time constants is to obtain a similar behavior with desired characteristics for each flight condition, for example, the step response overshoot. The desired time constant is determined using the bisection method starting with a large time constant with a very slow step response and large settling time, and then gradually reducing the time constant until a desired step response and settling time criteria are met. As mentioned above, overshoot will increase (from zero to a desired value) and settling time will decrease when the time constant is decreased. The stability of the closed loop system is ensured by monitoring that the maximum real part of all poles is less than zero.

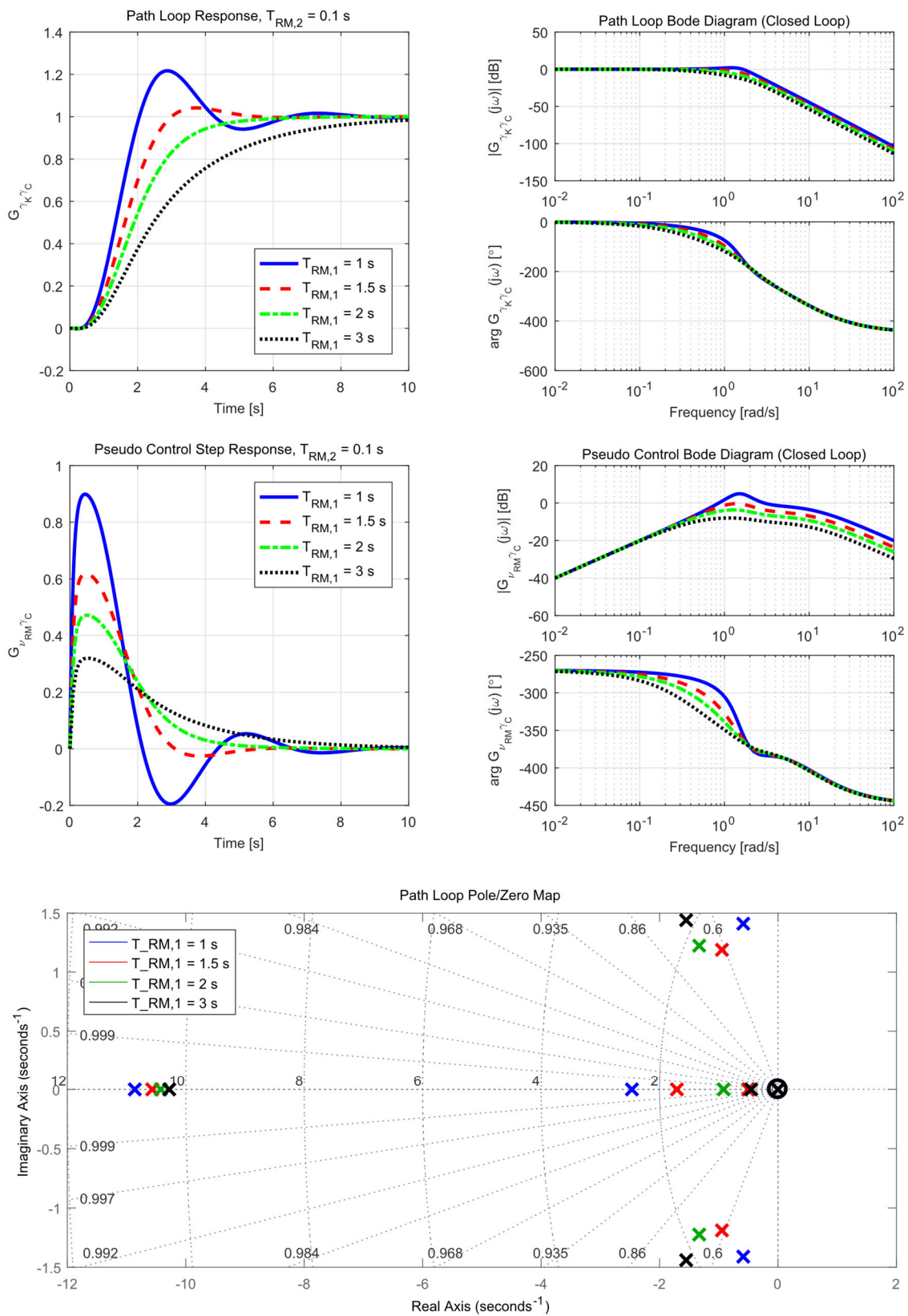


Figure 5.28: Effect of reference model time constant on path step response.

Plant Model with Disturbance Inputs

The error controller affects the path loop dynamics only in the presence of disturbances, as the reference model otherwise tracks the plant output. To illustrate the effect of disturbances on the closed loop dynamics for different values of proportional and integral error controller gains, consider the reduced plant model with a set of disturbance inputs in Figure 5.29. The possible source of errors is discussed in Section 4.3.

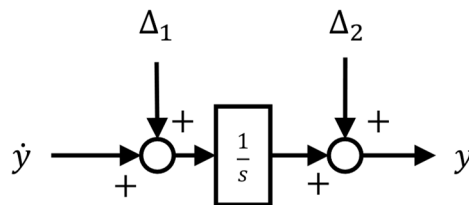


Figure 5.29: Simplified plant model with disturbances.

The disturbance input Δ_1 , added to the plant dynamics before the integrator can be interpreted as an inner loop steady state error that is integrated and thus with an increasing effect on the output. The disturbance input Δ_2 , added directly to the output signal, can be interpreted as an external disturbance causing a step in the flight path angle, track angle, or aerodynamic speed, such as wind disturbance.

Error Controller Proportional Gain

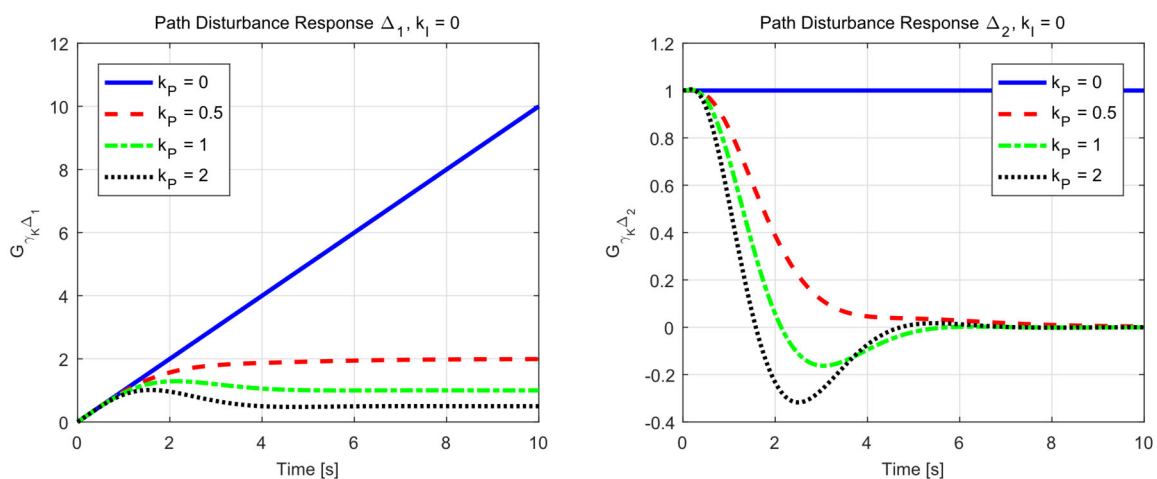


Figure 5.30: Effect of different values of proportional gain k_P on flight path state and pseudo control disturbance responses, with integral gain $k_I = 0$.

The effect of an increasing proportional gain k_P on the flight path state and the pseudo control is illustrated in Figure 5.30, for integral gain $k_I = 0$. For zero proportional gain, the disturbance Δ_1 is integrated and leads to a ramp in the flight path state. For increasing gain, the error is driven to a steady state value. The steady-state value is calculated from Eq. (4.43) according to

$$k_P \cdot e = -\Delta_1 \Rightarrow e = y_{RM} - y = -\frac{\Delta_1}{k_P}. \quad (5.185)$$

For the external disturbance Δ_2 , the proportional gain alone drives the error to zero, without a need for an integral gain.

Similar to the design of the time constant, the proportional gain is determined using the bisection method with the Δ_2 disturbance overshoot and settling time as design criteria, and closed loop stability as monitoring criteria. Starting with a small gain resulting in a slow step response and large settling time, the gain is gradually increased until desired step response and settling time criteria are met.

Error Controller Integral Gain

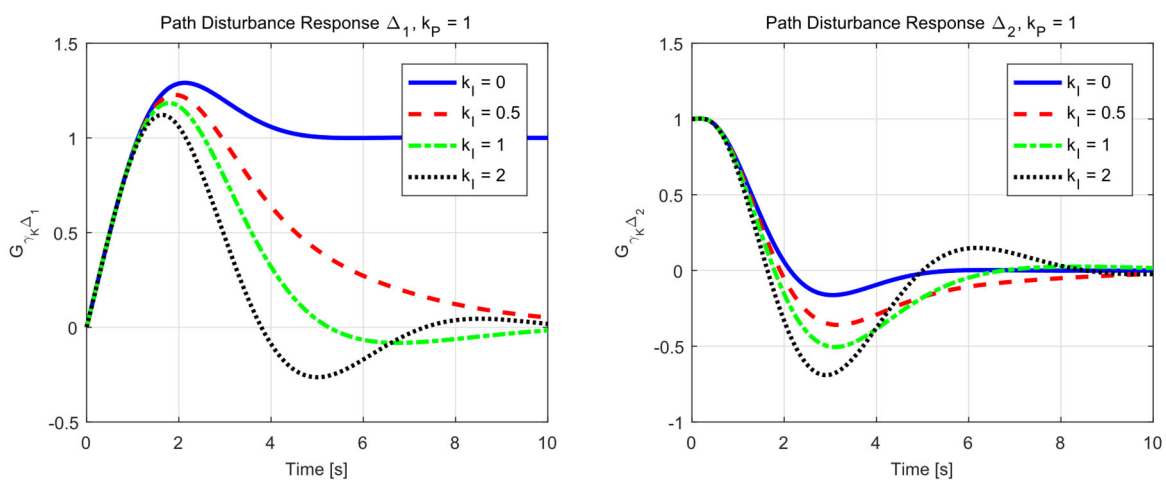


Figure 5.31: Effect of different values of integral gain k_I on flight path state and pseudo control disturbance responses, with proportional gain $k_P = 1$.

The effect of an increasing proportional gain k_I on the flight path state and the pseudo control is illustrated in Figure 5.31, for proportional gain $k_P = 1$. As expected, the steady-state error due to a Δ_1 is driven to zero by the integral gain. Thus, the integral gain is only required when a steady-state accuracy in the presence of the disturbance Δ_1 is required. Assuming that the inner loop controller is steady-state accurate, the integral gain is set to zero.

5.3.4 Speed Control Parameters

The time constants for the speed control reference model for speed by thrust and speed by pitch are designed analogously to the time constants for the vertical and lateral path reference models. For the speed by pitch loop, the normal specific force dynamics of the inner loop becomes limiting for speed control bandwidth, and for the speed by thrust the assumed thrust loop dynamics.

5.3.5 Energy Protection Parameters

When the aircraft enters the energy protection regions defined in Section 5.2.8, the feedback structure of the path loop changes, with the airspeed, flight path acceleration, and flight path

angle fed back to the flight path angle limit calculations. The maximum or minimum permissible flight path angles now function as the reference flight path angle.

To assess the energy protection parameters, i.e. the transition regions $\Delta V_{trans,min/max}$ and their inclination in the speed/acceleration phase plane determined by $k_{V,min/max}$, linearized models of the energy protection limits are used to design the protection parameters.

The linearized energy protection situation is the steady state situation where the aircraft is kept at the maximum or minimum airspeed. The transient situation where the aircraft approaches the maximum or minimum airspeed is validated in the nonlinear simulation.

For the steady state flight at maximum or minimum airspeed, the reference acceleration is given by $\hat{v}_{V,0} = 0$, and the reference deviations from maximum/minimum speeds $\Delta V_{min,0}$ and $\Delta V_{max,0}$ are given by

$$\Delta V_{min,0} = V_{IAS,0} - V_{IAS,min} \quad (5.186)$$

$$\Delta V_{max,0} = V_{IAS,0} - V_{IAS,max} \quad (5.187)$$

The linearized maximum and minimum flight path angles from Eq. (5.106)-(5.107), with reference acceleration $\hat{v}_{V,0} = 0$, and reference deviations from maximum/minimum speeds $\Delta V_{min,0}$ and $\Delta V_{max,0}$, are given by

$$\begin{aligned} \delta\gamma_{max} = & \operatorname{sgn}(\cos \gamma_{K,0}) \cdot \delta\gamma_K \\ & + \left(\frac{\gamma_{max,nom} - \gamma_{K,0}}{\Delta V_{trans,min}} - \frac{\dot{V}_{min,nom}}{g \cdot \Delta V_{trans,min} \sqrt{1 - \sin^2 \gamma_{K,0}}} \right) \cdot \delta V_{IAS} \\ & + \left(\frac{\gamma_{max,nom} - \gamma_{K,0}}{k_{V,min} \cdot \Delta V_{trans,min}} - \frac{\dot{V}_{min,nom} - k_{V,min} \cdot \Delta V_{trans,min}}{g \cdot k_{V,min} \cdot \Delta V_{trans,min} \sqrt{1 - \sin^2 \gamma_{K,0}}} \right) \\ & \cdot \delta\hat{v}_V \end{aligned} \quad (5.188)$$

$$\begin{aligned} \delta\gamma_{min} = & \operatorname{sgn}(\cos \gamma_{K,0}) \cdot \delta\gamma_K \\ & + \left(\frac{\gamma_{min,nom} - \gamma_{K,0}}{\Delta V_{trans,max}} - \frac{\dot{V}_{max,nom}}{g \cdot \Delta V_{trans,max} \sqrt{1 - \sin^2 \gamma_{K,0}}} \right) \cdot \delta V_{IAS} \\ & + \left(\frac{\gamma_{min,nom} - \gamma_{K,0}}{k_{V,max} \cdot \Delta V_{trans,max}} - \frac{\dot{V}_{max,nom} - k_{V,max} \cdot \Delta V_{trans,max}}{g \cdot k_{V,max} \cdot \Delta V_{trans,max} \sqrt{1 - \sin^2 \gamma_{K,0}}} \right) \\ & \cdot \delta\hat{v}_V \end{aligned} \quad (5.189)$$

5.4 Mode Control and Monitoring Interface Design

In conjunction with the system functions and control law/logic design, a compact Human-Machine-Interface (HMI) was developed, consisting of a Mode Control Panel (MCP) and a Mode Control and Monitoring Display (MCMD) to enable control of the AFCS and provide state awareness of the experimental flight control system. The MCP layout and logic as well as MCMD layout and display application were designed by the author and presented in this thesis. The hardware and embedded software framework were built by AEE Aircraft Electronics Engineering in close cooperation with TUM-FSD.

MCP and MCMD layouts and functionalities were iteratively developed as executable desktop applications, interfacing with the model-in-the-loop simulation environment, before detailed specification and hardware/software production. This rapid prototyping of the HMI greatly supported control law prototyping and early validation of system operation. The design was iteratively improved based on test pilot feedback before hardware manufacturing and HIL integration. Display software and mode control logic were fine-tuned in HILS and AILS, and validated in flight tests and remote-control operation.

5.4.1 Preliminary Design Considerations

The MCP and MCMD layouts share design principles with typical state-of-the-art autopilot mode control interfaces and PFD. This was important to ensure familiarity and intuitive handling for test pilots and operators. The MCP and MCMD however include design elements, information density, and form factors driven by the experimental nature of the flight control system:

- Separated target selection, synchronization and confirmation logic and annunciation to increase target selection awareness
- Individual engagement/disengagement control and status of actuation resources for each axis, together with control surface commands and current deflections to support single/multi-axis control system verification and validation
- Dynamic indications of flight path limits and targets when control objective prioritization or protections become active

To validate the sensor inputs to the control laws, the MCMD displays the aircraft states (attitude, altitude, speed, etc.) on the PFD from the FCC, rather than directly from the sensors.

A compact single panel, single display design allows MCP and MCMD installation into an application aircraft for test pilot control of the AFCS functionalities in-flight or configuration as a UAV or OPV ground station providing a remote operator with AFCS control and awareness via data link.

5.4.2 Architecture

The principal MCP/MCMD architecture, architecture elements, and interfaces are illustrated in Figure 5.32.

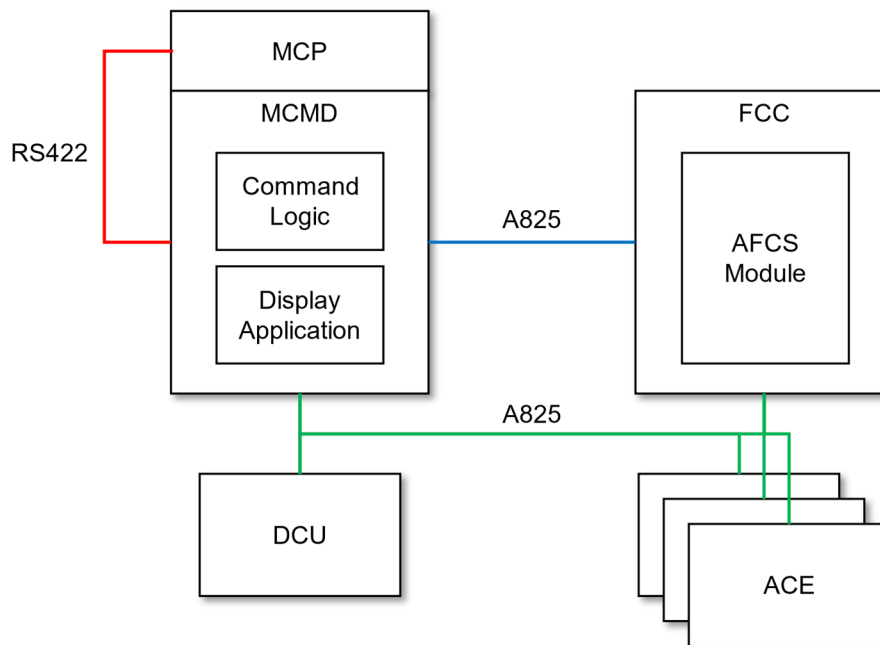


Figure 5.32: Overview of HMI architecture and MCP-MCMD-FCC-ACE-DCU interfaces.

Mode Control Panel (MCP)

The MCP is the main interface of the pilot to select automatic flight control functions, target sources, and actuation resources. The MCP processes

- pushbutton actions, selector knob turning increments, and selector knob push/pull actions into corresponding flags sent to the MCMD Command Logic via RS422 and to the FCC via ARINC825
- desired actuation resource engagement/disengagement for the ACEs

The MCP receives

- active modes and guidance source from the FCC
- actuation resource engagement status from the ACE

The MCP layout and operation are described in Section 5.4.3.

Mode Control and Monitoring Display (MCMD)

The MCMD contains two functional elements: the MCMD Command Logic and the MCMD Display Application, which are described in the following. The MCMD layout is described in Section 5.4.4.

MCMD Command Logic

The MCMD Command Logic receives

- pushbutton and selector knob actions from the MCP via RS422

- sensor data from FCC via ARINC825

and calculates

- the preselected commands and flags for the display application
- the desired (pulled/confirmed) commands and flags sent to the FCC via ARINC825

A detailed specification for the MCMD Command Logic is included in Appendix C.1

MCMD Display Application

The MCMD Display Application receives

- the preselected commands and flags from the MCMD Command Logic
- active commands and modes from the FCC via ARINC825
- sensor data from FCC via ARINC825
- control surface deflections from the ACEs via ARINC825
- subsystem status information from the FCC, ACEs, and DCU via ARINC825

and presents this information according to the layout presented in Section 5.4.4.

Flight Control Computer (FCC)

The FCC contains the functional modules of the modular FGCS, see Figure 1.1, Section 1.1, including the AFCS module. The functional architecture of the AFCS module is illustrated in Figure 5.10, Section 5.2.1.

In selected operation, when the AFCS takes the commands from the MCP, parts of the associated mode control logic are performed within the AFCS module. The FCC/AFCS receives

- pushbutton actions from the MCP via ARINC825
- desired commands, command set status flags, turn direction flags, and course from the MCMD Command Logic via ARINC825
- toggle flags from the MCMD Command Logic via ARINC825
- actuation resource engagement status from the ACE via ARINC825

The FCC outputs

- active commands and modes to the MCMD Display Application
- sensor data that the FCC processes to the MCMD Display Application
- control surface/actuator position commands to the MCMD Display Application
- actuator position commands to the ACE

The MCMD Command Logic and the inputs to the FCC/AFCS are detailed in Appendix C.1.

Actuator Control Electronics (ACE)

The ACE receives

- desired actuation resource engagement/disengagement from the MCP
- actuator position commands from the FCC

The ACE outputs

- actuation resource engagement status to the MCP/MCMD and FCC
- actuator positions to the MCMD Display Application and FCC

Data Concentrator Unit (DCU)

The DCU collects additional sensor data from for example control surface sensors, pitch trim force sensors and outputs these to the MCMP Display Application and the FCC via ARINC 825.

5.4.3 Mode Control Panel Design

Principal Layout

The MCP layout is illustrated in Figure 5.33.

Master Power Switch

The master power switch turns on the power to the AFCS and the monitoring display.

Target Selector Knobs

The MCP has four push/pull target selectors for preselecting, confirming, or synchronizing the indicated airspeed, heading/track, altitude, and vertical speed/flight path angle target.

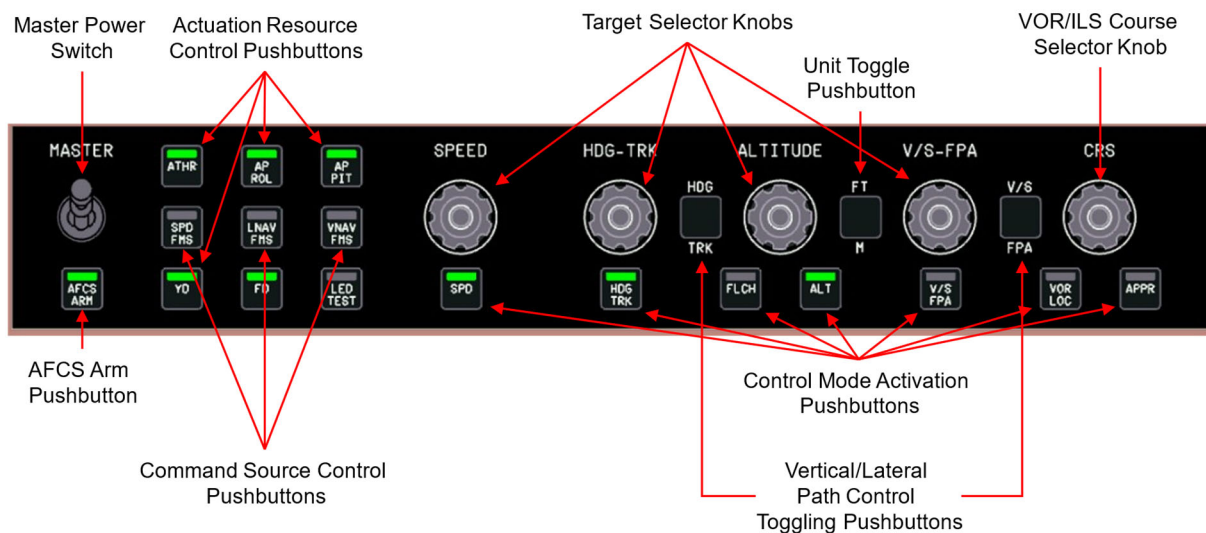


Figure 5.33: Mode Control Panel (MCP) layout.

A new target is *preselected* by turning the corresponding selector knob. The preselected value is displayed in the first line on the MCMD, see Figure 5.34, Section 5.4.4. A preselected target is *confirmed* by pulling the selector knob. The confirmed target is sent to the FCC. The second line on the MCMD always displays the target from the FCC, either the target selected on the MCP or managed by the external guidance system. If the selector knob is pushed, the

corresponding preselected target is *synchronized* to the current measured aircraft state (indicated airspeed, heading or track, altitude, vertical speed or flight path angle).

The MCP also has one selector knob for setting the desired ILS or VOR course. Pushing the selector synchronizes the desired course to the current heading. The VOR/ILS course is indicated on the Horizontal Situation Indicator (HSI) section of the MCMD.

Function Activation Pushbuttons

The MCP has six pushbuttons to activate the different vertical and lateral functions:

- The FLCH pushbutton activates the Flight Level Change function if pressed when in Pitch Hold, Vertical Speed, Flight Path Angle or Approach, or reverts to Pitch Hold if pressed when Flight Level Change is active
- The HDG TRK pushbutton activates the Heading or Track function (depending on the status of the HDG/TRK mode switch) if pressed when in Roll Hold, VOR/Localizer Capture/Track or Approach, or reverts to Roll Hold if pressed when Heading or Track is active
- The ALT pushbutton activates the Altitude Hold function and resets the altitude target to the current altitude if pressed when in Pitch Hold, Flight Level Change, Vertical Speed, Flight Path Angle or Approach (i.e., a level off at current altitude), or reverts to Pitch Hold if pressed when Altitude Hold is active
- The VS FPA pushbutton activates the Vertical Speed or Flight Path Angle function (depending on the status of the VS/FPA mode switch) if pressed when in Pitch Hold, Flight Level Change or Approach, or reverts to Pitch Hold if pressed when Vertical Speed or Flight Path Angle is active
- The VOR LOC pushbutton arms the VOR/Localizer function (the active lateral function remains active until the VOR/Localizer is captured) or reverts to Roll Hold if pressed when the VOR/Localizer function is active
- The APPR pushbutton arms the Approach function (the active vertical and lateral functions remain active until the localizer and the glideslope respectively have been captured) or reverts to Pitch / Roll Hold if pressed when the Approach function is active

Vertical/Lateral Path Control Toggling Pushbuttons

The MCP has two pushbuttons for toggling lateral and vertical path modes. The HDG/TRK toggle pushbutton toggles the selected lateral path function between Heading and Track. The V/S/FPA toggle pushbutton toggles the selected vertical path function between Vertical Speed and Flight Path Angle. When switching from Vertical Speed to Flight Path Angle, the current flight path angle is set as preselected and active target, and vice versa when switching from Flight Path Angle to Vertical Speed.

Unit Toggle Pushbutton

The FT/M pushbutton toggles the indication of the preselected and active altitude target on the MCMD to meters, as well as an indication of the current altitude in meters in addition to the default indication in feet on the altitude indicator.

When the altitude unit is meters, the preselected altitude target is set in increments of 100 meters.

Airspeed and vertical speed and associated targets remain indicated in knots and feet per minute.

Actuation Resource Control Pushbuttons

The actuation resource control pushbuttons engage the individual actuation resources for the pitch axis (pitch autopilot or AP PIT), roll axis (roll autopilot or AP ROL), yaw axis (yaw damper or YD), and throttle/thrust setting (autothrust or ATHR). The availability of each axis is aircraft-specific, and engagement of the pitch or roll autopilot without previous engagement of the yaw damper may automatically engage the yaw damper or not, depending on the application-specific actuation control resource logic.

Command Source Control Pushbuttons

The VNAV FMS pushbutton engages/disengages managed control of the vertical profile.

The LNAV FMS pushbutton engages/disengages managed control of the lateral profile.

The SPD FMS pushbutton engages/disengages managed control of the speed profile.

Additional Pushbuttons

The FD pushbutton turns the Pitch/Roll Flight Director and Thrust Director on or off.

Pressing the LED Test pushbutton illuminates all pushbutton LEDs on the MCP.

The AFCS ARM pushbutton activates the power supply to the actuator clutches, thereby allowing the engagement of the actuation control resources. Pressing the AP DISC button on the control column disconnects the power supply to the clutches and they need to be “rearmed” in order to reengage an actuation control resource.

5.4.4 Mode Control and Monitoring Display Design

The MCMD annunciates the current state of the AFCS, i.e., the active actuation resources, functions, and targets, as well as the input signals from the sensors, with which the AFCS is working. The MCMD layout is illustrated in Figure 5.34.

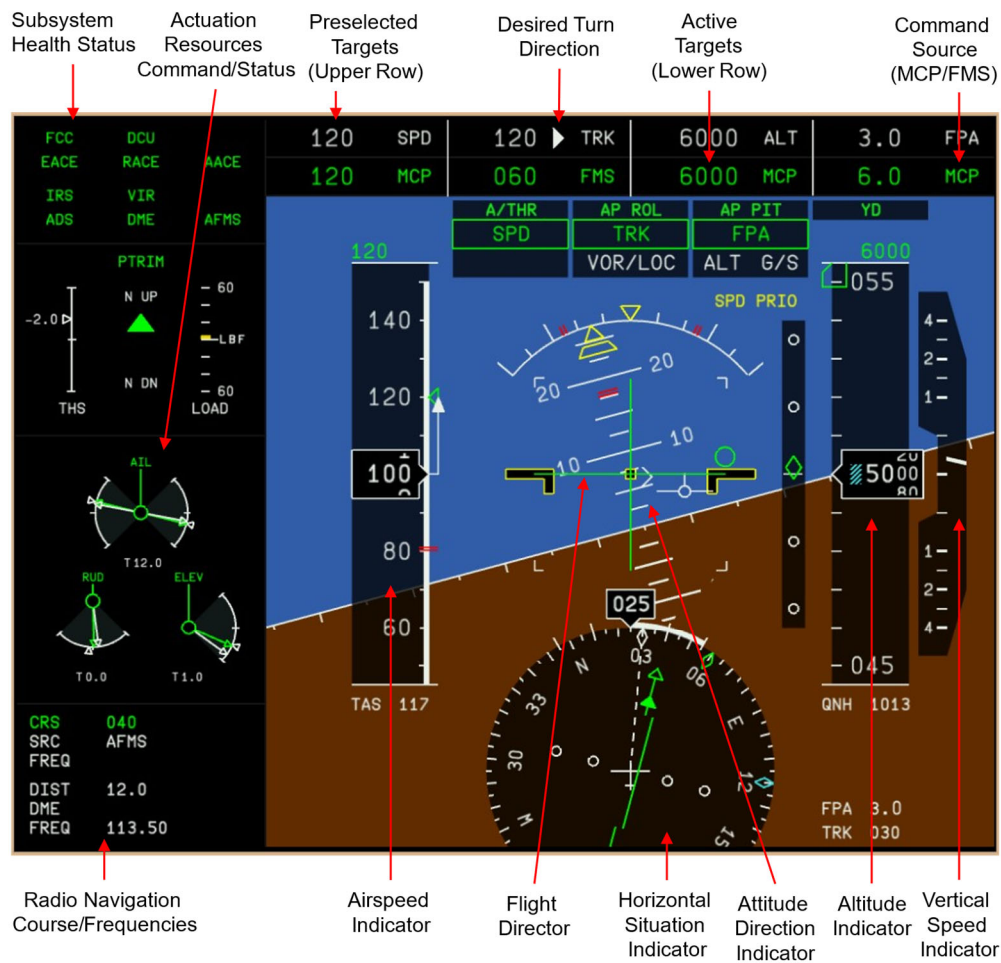
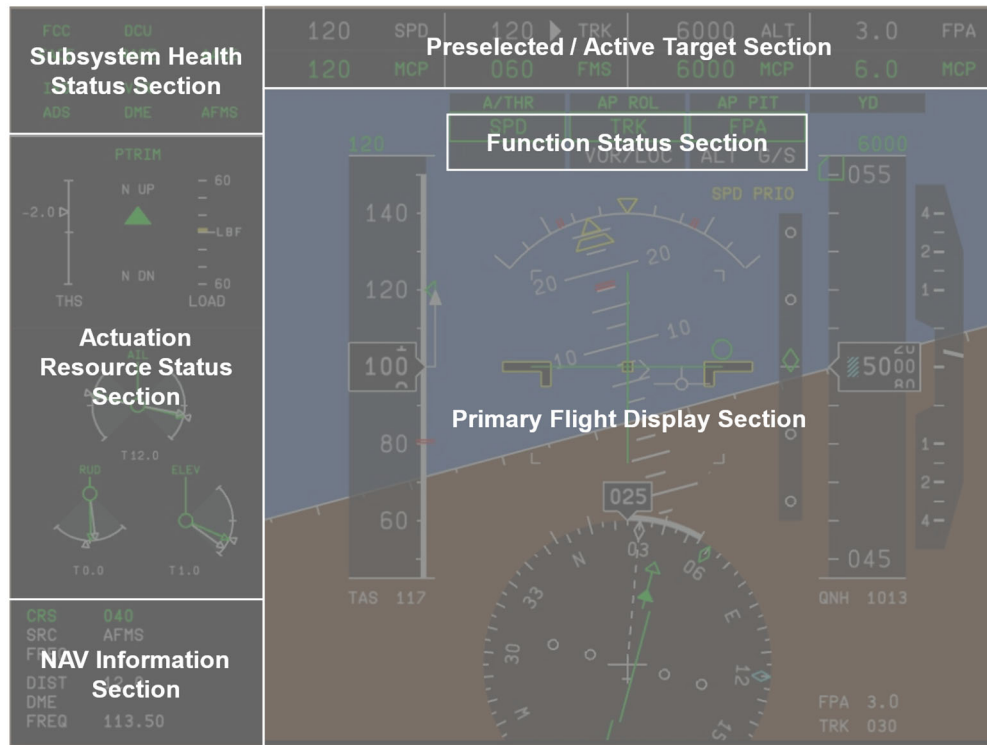


Figure 5.34: Mode Control and Monitoring Display (MCMD) layout with sections.

Preselected / Active Target Section

The Preselect / Active Target section at the upper part of the display indicates preselected, active, and inactive targets for indicated airspeed, heading/track, altitude, and vertical speed/flight path angle. The upper row indicates the preselected targets, whereas the lower row indicates the confirmed and thus active or passive targets. An active target, i.e., a target being acquired or tracked is shown in green color. A passive target, i.e., a target confirmed but currently not being acquired or tracked is shown in white color.

Primary Flight Display Section

The PFD section indicates the sensor inputs that the AFCS utilizes, together with targets, limits, navigation, and approach information, as well as FD.

The Attitude Direction Indicator (ADI) indicates the pitch and roll attitude of the aircraft with an extended artificial horizon, with pitch and roll angle limits. The bank angle pointer also indicates sideslip based on the measured lateral acceleration. A white flight path vector birdie indicates the flight path and drift angle, and a green flight path vector command birdie indicates the commanded flight path angle. A red, separated flight path vector birdie indicates the maximum (upper half of the birdie) and minimum (lower half of the birdie) flight path angle. A white chevron indicates the energy angle (based on the current acceleration and flight path angle according to Eq. (3.98)), and two larger green chevrons indicate the commanded energy angle (based on the desired acceleration and flight path angle).

The FD shows a horizontal and a vertical bar. The horizontal bar indicates a pitch up or down command if moving above or below the aircraft symbol, and the vertical bar indicates a roll left or right command if moving to the left or the right of the aircraft symbol center.

The Horizontal Situation Indicator (HSI) indicates the current heading and track angle, as well as the commanded heading or track angle, depending on active control mode. A heading command bug is visible when the heading/track toggle is set to heading, with a white heading bug for the preselected heading on the MCP and a green bug for the confirmed and active heading target. A track command bug is visible when the heading/track toggle is set to track, with a white track bug for the preselected track on the MCP and a green bug for the confirmed and active track target. A turn rate indicator on the outer boundary of the HSI indicates the current turn rate as a white tape, with ticks for half and standard rate turn. The HSI further includes a VOR/localizer course indicator, with a VOR/localizer deviation indicator visible when the VOR/localizer function is active, as well as a VOR/localizer to/from indication.

The airspeed tape indicator to the left on PFD indicates the current indicated airspeed along with a white arrow indicating the airspeed trend (current acceleration). The airspeed command bug indicates the preselected airspeed target (in white) or the active airspeed target (in green). The airspeed target is also numerically indicated above the speed tape. The tape further includes indications of minimum and maximum AFCS speed, flaps extended speed, stall speed, as well as a numerical indication of the true airspeed below.

The altitude tape indicator to the right on the PFD indicates the current barometric altitude. A white altitude bug indicates the preselected altitude target, and a green altitude bug indicates the confirmed and active altitude target being acquired or tracked. The altitude target is also numerically indicated above the altitude tape. The current QNH setting is indicated below the altitude tape. A vertical speed indicator to the right of the altitude tape indicates the current vertical speed, as well as the vertical speed target if the vertical path toggle is set to vertical speed mode.

Actuation Resource Status Section

The Actuation Resource Status section indicates the engagement status of each actuation resource (i.e., pitch, roll, and yaw actuation, and if installed, autothrust). The section further indicates the actuator commanded and actual positions (aileron, rudder, elevator actuators) as well as the actual primary control surface positions (ailerons, rudder, elevator). Depending on the trim design of the aircraft (trim tabs, trimmable horizontal stabilizer, etc.) and the implemented automatic trim functionalities, the MCMD indicates trim action and control forces. In Figure 5.34, the position of the trimmable horizontal stabilizer position is indicated, as well as the horizontal stabilizer trim nose-up/nose-down movement.

Function Status Section

The Function Status section in the upper part of the PFD indicates the active and armed speed, vertical, and lateral flight control modes, according to Table 5.3 in Section 5.1.2. A manual or automatic mode transition is indicated by a green box around the mode indication for 5 seconds.

NAV Information Section

The NAV information section down to the left on the MCMD indicates the VOR/ILS course setting, either set using the MCP CRS selector knob or, if VNAV/LNAV FMS is on, an approach course from an external guidance system, as well as the source as either MCP or FMS.

The NAV information section further indicates the selected VOR/ILS station frequency, the distance to the selected Distance Measuring Equipment (DME) station, the decoded DME station identifier, as well as the DME station frequency.

Subsystem Health Status Section

The Subsystem Health Status section indicates the health status of each control system component. These indicators are specific to the system architecture of the aircraft application – in Figure 5.34 the subsystem components are the Flight Control Computer (FCC), Data Concentrator Unit (DCU), the Elevator/Rudder/Aileron Actuation Control Electronics (EACE/RACE/AACE), the Inertial Reference System (IRS), the Air Data System (ADS), the navigation receiver (VIR), the DME receiver and the external guidance system (AFMS).

6 Safety Assessment

This chapter describes the safety assessment activities of the AFCS development, according to the safety assessment process track of the System Development Process defined in Section 2.3.3. The safety assessment process track for the AFCS focuses on the *system-level FHA*, according to the ARP4754B/ARP4761A system safety assessment process, Figure 2.5, see Section 2.2.1. The system-level FHA is function-oriented and architecture-agnostic with the purpose of identifying system hazards and determining the necessary level of safety. The objective AFCS system-level FHA is to demonstrate how the AFCS functions may be assessed with respect to safety and provide the necessary inputs to an application life cycle and the development of a system architecture that can meet the requirements of the specific operational scenario and regulatory environment. The system-level FHA activities include:

- Identification and description of failure conditions
- Determination of failure condition effects
- Classification of failure conditions
- Assignment of requirements to failure conditions
- Identification of method for compliance verification

Section 6.1 discusses the main inputs of the FHA: the set of system functions from the system definition track (see Section 5.1) together with the operational concept (operational scenarios, flight phases), and applicable certification regulations and guidance material (such as advisory circulars with classification guidance).

The FHA output is a set of *worksheets*, tables with rows for each assessed function and columns representing the outcome of each activity above. The FHA worksheet columns are explained in Table 2.1. The worksheets from the AFCS system-level FHA are included in Appendix E. Section 6.2 discusses the main outputs from the FHA: the significant failure conditions and the associated required functional development assurance levels (FDAL) as well as safety-related requirements regarding for example fail-safe behaviors in the presence of faults, fault detection mechanisms, warnings and cautions, or annunciations.

The reference platform for the development of the modular FGCS including the AFCS that is the subject of this thesis is the DA42 OE-FSD, a Class II aircraft according to AC 23-1309-1E [104], and the FHA and failure condition classification is performed in this context. The modular FGCS concurrently constitutes the basis for several application cases on other aircraft

platforms, each with its own life cycle model, illustrated in Figure 2.2. Section 6.3 discusses aspects of the physical system architecture that realizes the system functions. The architecture is dependent on the application scenario and aircraft platform and different valid system architecture solutions with varying levels of redundancy, dissimilarity, and independence may satisfy the required functional development assurance, resulting in architecture candidates with varying levels of hardware/software development assurance levels for primary and secondary systems.

6.1 FHA Preliminary Considerations

6.1.1 System Functions under Analysis

The scope of the AFCS system-level FHA is the application-generic system functions as presented Section 5.1.2. Table 6.1 summarizes the application-generic system function under analysis, with references to functional descriptions in Section 5.1.2 and the corresponding FHA work sheet, collected in Appendix E.

Table 6.1: Application-generic system function subject to FHA.

System Functions	Application-Generic Functions	Functional Description Reference (Section 5.1.2)	FHA Work Sheet Reference (Appendix E)	Comment
Automatic Flight Control Functions	Guidance modes	Table 5.2	Appendix E.3	
	Vertical/lateral path control modes) Vertical/lateral trajectory control modes Speed control modes	Table 5.3		Automatic control with AP/ATHR Manual control coupled with FD
	Envelope protection modes Control objective prioritization modes (maneuvering integrity)	Table 5.4		
Human-Machine-Interface Functions	Mode and target selection/ annunciation Actuation resource selection/ annunciation Flight director	Table 5.5	Appendix E.4	FD assessed coupled with AFCF
Autopilot Functions	Automatic pitch/roll/yaw control Autopilot engagement/disengagement	Table 5.6	Appendix E.5	Automatic pitch/roll/yaw control assessed coupled with AFCF
Autothrust Functions	Automatic thrust control Autothrust engagement/disengagement	Table 5.6	Appendix E.6	Automatic thrust control assessed coupled with AFCF
Safety Functions	Input monitoring functions Subsystem monitoring functions Autopilot/Autothrust automatic disengagement	Table 5.7	Appendix E.7	

As illustrated in the principal AFCS architecture with generic architecture elements, Figure 2.11, the AFCF for the vertical plane, lateral plane, and speed are coupled to the autopilot and

autothrust functions for automatic control of the flight path, as well as to the HMI and the Flight Director functions for manual control of the flight path. The principal automatic flight control mode and autopilot/autothrust couplings are illustrated in Figure 5.2. In the AFCS system-level FHA, the AFCF are assessed for the automatic control coupled with autopilot/autothrust and manual control with flight director guidance separately, as noted in Table 6.1.

6.1.2 Flight Phases

For the development of the AFCS concept as part of the modular FGCS described in Section 1.1, the research platform of the TUM-FSD, the Diamond DA42 OE-FSD, described in Section 1.4.1, is considered the reference platform for the safety assessment. Section 5.1.1 describes an example reference use case, an OPV, that operates both as a normal CS-23 aircraft as well as an emulated UAV with a safety pilot monitoring the experimental systems. An example mission profile with flight phases and typical vertical, lateral, and speed control functions for operation as a conventional CS-23 aircraft as well as operation as an OPV is illustrated in Figure 5.1.

The AFCS system-level FHA considers the system functions, failure conditions, flight phases, and applicable guidance material for the operation of the aircraft as a conventional CS-23 aircraft. The flight phases considered are:

- Departure / Descent
- Enroute / Return (Cruise)
- Mission (Maneuvering Flight)
- Approach

The take-off and landing phases are not considered, as they are performed manually during conventional operation. For an FGCS including automatic take-off and landing functions, to enable autonomous UAV emulation, an extended FHA would be necessary specifically concerning those flight phases.

6.1.3 Guidance Material

The ARP4754B [37] governs the scope of the safety assessment and the objectives of the FHA in relation to the development activities. Detailed guidance on the individual FHA steps is further provided in ARP4761A [38]. The FAA AC 23.1309-1E [104] is the main guidance material providing acceptable means of showing compliance with the certification specifications for CS-23 aircraft and providing additional detailed FHA guidance for CS-23 aircraft in general and automatic flight control systems in particular.

The AFCS system-level FHA leans on the AC 23.1309-1E for guidance on failure condition classification, FDAL assignment, and quantitative safety requirements. Table 2.3 summarizes the AC 23.1309-1E guidance on failure condition classification depending on the failure effect on airplane, occupants, and flight crew. Table 2.4 provides guidance on the classification of a hardover malfunction without warning, depending on the autopilot axes under control and the automatic control system authority. For the example reference use case, an OPV that operates both as a normal CS-23 aircraft as well as an emulated UAV, the AFCS is a multi-axis system

with unlimited authority. Thus, a hardover malfunction is classified as a catastrophic failure condition. Table 2.5 summarizes the AC 23.1309-1E guidance on the relation between failure condition classification, quantified occurrence probabilities, and DAL requirements for primary (P) and secondary (S) systems for different aircraft classes. According to Table 2.5 and as noted in Table 1.3, the DA42 OE-FSD is considered a Class II aircraft. This is adhered to for the FDAL classification. Table 6.2 summarizes the relevant guidance for the AFCS system-level FHA. For system architecture considerations, the ARP4754B provides guidance on the FDAL/IDAL assignment process and the dependencies between FDAL and IDAL for different system architecting approaches. This is further discussed in Section 6.3.

Table 6.2: Summary of relationship between probabilities, failure condition severity and DAL requirements for primary (P) and secondary (S) items for Class II aircraft (extract from [104] Figure 2).

Classification of Failure Conditions	No Safety Effect	Minor	Major	Hazardous	Catastrophic
Class II (Typically MRE, STE or MTE 6,000 pounds or less)	No Probability or IDAL Requirement	<10 ⁻³ / FH P=D	<10 ⁻⁵ / FH P=C, S=D	<10 ⁻⁶ / FH P=C, S=C	<10 ⁻⁷ / FH P=C, S=C

6.2 Summary of FHA Results

The FHA worksheets with the results of the AFCS system-level FHA are included in Appendix E. This section summarizes the significant failure conditions, the associated quantitative safety objectives, and functional development assurance levels.

6.2.1 Significant Failure Conditions

Table 6.3 summarizes the failure conditions that are considered significant and require a thorough and quantitative safety assessment during system design and implementation. The significant failure conditions are those identified as hazardous or catastrophic, as well as those major failure conditions related to a new and conceptual design where the classification itself should be validated (it may be considered hazardous, as details become known during the system design).

Table 6.3: Summary of significant failure conditions.

Function	Failure Condition	Phase	Failure Effect	Class.	Comment
Automatic Vertical Path Control	Erroneous vertical path control, hardover malfunction	Approach	Flight crew recognizes unexpected aircraft behavior and disconnects AFCS to recover flight promptly. Large reductions in safety margins and large increase in crew workload. For nose down hardover, risk of not meeting obstacle clearance. For nose up hardover, risk of reaching stall speed.	CAT	AC 23.1309-1E: Catastrophic for multi-axis, unlimited authority AFCS The exact effect on the aircraft in case of a runaway is not known and the classification should be validated through simulation.
	Total loss of vertical path control without indication	Approach	Crew must recognize by secondary means (deviation from flight path) and take over control. Significant increase in crew workload.	MAJ	
	Erroneous vertical path control, slowover malfunction	Approach	Flight crew monitors AFCS operation, recognizes erroneous or inconsistent vertical control or gradual departure from reference path. Flight crew disconnects AFCS manually to recover flight promptly. Reductions in safety margins and significant increase in crew workload.	MAJ	
Automatic Lateral Path Control	Erroneous lateral path control, hardover malfunction	Approach	Flight crew recognizes unexpected aircraft behavior, e.g., significant bank angle and deviation from intended runway track and disconnects AFCS to recover flight promptly. Large reductions in safety margins and large increase in crew workload.	CAT	AC 23.1309-1E: Catastrophic for multi-axis, unlimited authority AFCS The exact effect on the aircraft in case of a runaway is not known and the classification should be validated through simulation.
Automatic Thrust / Speed Control	Total loss of thrust / speed control without indication	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	
	Erroneous thrust / speed control without indication	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	
	Asymmetric thrust without annunciation, autopilot engaged	All	AFCS tries to compensate asymmetric thrust setting with roll/yaw commands, pilot may not immediately recognize the failure condition. Significant reduction in safety margins.	MAJ	The behavior of the aircraft during failure conditions related to this function must be validated

Function	Failure Condition	Phase	Failure Effect	Class.	Comment
Manual Control with Vertical Flight Guidance	Misleading vertical flight guidance command display	Approach	Guidance inconsistent with intended flight path, identified by the crew. Reduction of safety margins.	MAJ	
Energy Protection	Erroneous low-speed protection	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	
	Erroneous high-speed protection	Mission	Flight crew may not immediately recognize the failure condition. May cause overspeed situation during maneuvering in mission phase. Significant reduction in safety margins.	MAJ	
Autopilot Dis-engagement	Loss of the ability to disengage the autopilot with control column disconnect switch	Approach	Flight crew not able to disconnect autopilot at decision height or during autopilot or trim failure. Pilot recognizes situation and overcomes the autopilot servos to control the aircraft. Autopilot must be disengaged by other means (e.g., circuit breakers). Increase in crew workload.	MAJ	
	Inadvertent disengagement without annunciation	All	Pilot recognizes that autopilot has disengaged by deviation from intended flight path, takes manual action to retain control of the aircraft. Increase in crew workload and reduction of safety margins.	MAJ	
Autothrust Engagement	Inadvertent autothrust engagement during autopilot approach, low thrust setting	Approach	Aircraft suddenly loses speed, pilot recognizes unintended thrust change and overpowers autothrust servo, might cause stall warning. Increase in crew workload and reduction in safety margins.	MAJ	
Autothrust Dis-engagement	Inadvertent disengagement of autothrust, without annunciation	All	Pilot does not immediately recognize autothrust disengagement, aircraft speed drifts away from reference, may cause overspeed or stall warning.	MAJ	

6.2.2 Quantitative Safety Objectives

The dimensioning failure condition for the AFCS is a hardover malfunction during automatic path control in the approach phase, see Table 6.3. For a multi-axis system with unlimited

authority, this failure condition is classified as catastrophic, according to AC 23.1309-1E [104], Appendix 1. The associated allowable quantitative probability requirement is $P < 10^{-7}$ per flight hour, see Table 6.2.

Several AFCS failure conditions are classified as major, see Table 6.3. For these failure conditions, the associated allowable quantitative probability requirement is $P < 10^{-5}$ per flight hour, see Table 6.2.

For an aircraft-specific application, it shall be shown in the PSSA and the SSA, for example using a quantitative FTA, that the quantitative safety objectives are satisfied by the system design.

6.2.3 Functional Development Assurance Levels

Table 6.2 describes the FDAL associated with each failure condition classification. For catastrophic failure conditions for a Class II aircraft, the required development assurance level for a primary system is level C. For major failure conditions, the required assurance level is likewise level C. The stronger safety requirement for the catastrophic failure conditions is the probability of occurrence per flight hour, which for a catastrophic failure condition is two orders of magnitude lower than that for a major failure condition.

The FDAL is a requirement on the rigor of development activities associated with the function and items implementing the function, i.e., that a certain level of development rigor shall assure that to a sufficient degree, development errors are mitigated.

6.3 System Architecture Considerations

As discussed in the introduction to this chapter, the safety assessment process for the AFCS focuses on the system-level FHA, as summarized in Section 6.2. The modular FGCS of which the AFCS is a part is concurrently the basis for several application cases on other aircraft platforms, each with their life cycle model, illustrated in Figure 2.2. This section discusses aspects of the physical system architecture that realizes the system functions.

The physical system architecture is dependent on the application scenario and aircraft platform. Depending on the application aircraft classification according to AC 23.1309-1E as well as the operational and performance requirements of the AFCS, with resulting operational envelope, control authority, and bandwidth, the failure condition classifications as well as the development assurance levels are subject to change. This is illustrated for the hardover malfunction failure condition in Table 6.4. Different valid system architecture solutions with varying levels of redundancy, dissimilarity, and independence may satisfy the required functional development assurance, resulting in architecture candidates with varying levels of hardware/software development assurance levels for primary and secondary systems. AC 23.1309-1E requires that no single failure will result in a catastrophic failure condition, i.e., for a multi-axis AFCS with unlimited authority, a redundant system will be required to ensure that a single failure does not lead to a hardover malfunction.

Table 6.4: Hardover malfunction classification and required development assurance for different application platform classifications.

AFCS Authority	Application Platform Classification			
	Class II (e.g., DA42 OE-FSD)		Class IV (e.g., Dornier Do-228 D-CODE)	
Single-axis, limited authority	MAJ	FDAL C	MAJ	FDAL C
Multi-axis, limited authority	HAZ	FDAL C	HAZ	FDAL B
Multi-axis, unlimited authority	CAT	FDAL C	CAT	FDAL A

6.3.1 Failure Condition Mitigation Strategies

To prevent failures leading to catastrophic or hazardous conditions, multiple strategies and methods may be employed to reduce or nullify the risk for severe failure conditions. Leveson [55] discusses four principal failure condition mitigation strategies: hazard elimination, hazard reduction, hazard control, and damage reduction. The mitigation strategies are discussed in the following in light of the identified AFCS failure conditions.

Hazard Elimination (Intrinsic Safety)

An intrinsically safe system design is incapable, under normal as well as abnormal operating conditions, to cause danger, either by for example limitation of the energy generated or an inherent inability of causing a harmful exposure. For the AFCS, the severity of the failure condition of a hardover malfunction is associated with the number of axes and authority of the automatic control function. The necessary AFCS control authority will be dependent on the aircraft-specific application scenario and required control bandwidth to meet performance requirements. For an AFCS with unlimited authority, utilizing the full dynamics of the aircraft to track flight path commands for example in a maneuvering mission phase, the failure condition classification follows as in Table 6.3.

For a more conventional operational scenario, where a lower authority and lower bandwidth AFCS actuation system (low deflection rate and range) is sufficient to meet performance criteria, the corresponding failure classification is reduced to a hazardous or major condition. The catastrophic failure condition is eliminated as even in the case of an AFCS hardover or actuation system jamming, the system does not have sufficient impact over time to put the aircraft in a hazardous flight condition.

The AFCS may also be physically inhibited for flight phases where it is not intended to be used, for example through disconnected power supply, in order to eliminate the possibility of inadvertent engagement. This is incorporated into the MCP design with a Master Power Switch, see Figure 5.33, that allows manual control of the AFCS power supply (as implemented for example on the Do-228 D-CODE).

Hazard Reduction

The hazard occurrence probability can be reduced by implementing constraints and safety interlocks or minimizing the operating conditions that could lead to a hazard. A constraint or safety interlock in the form of a physical limitation or inhibition as described above may be regarded as a hazard elimination, whereas a software-based safety interlock may be regarded as a hazard reduction. A software-based limitation or safety interlock depends on inputs in the form of sensor measurements with the objective of constraining for example operational envelope, or prohibiting engagement of functions under certain conditions, reducing the probability of potentially hazardous conditions. As the sensor measurements themselves may be erroneous, a software-based hazard reduction strategy is not fool-proof.

Hazard Control

Hazard control includes passive or active control measures that may mitigate the effects of a hazard once it has occurred. Examples of typical hazard control measures are relief valves in case of over-pressurization (passive measure) or smoke detectors and sprinkler system (active measures). For the AFCS, different passive and active hazard control measures are possible. Active monitoring functions and automatic AFCS disengagement provide added safety in case of a detected failure condition that could lead to a hazardous state, such as a sensor fault or an actuation system fault. Overload clutches act as a passive hazard control measure in case of an actuation system hardover, allowing the flight crew to disengage the AFCS by overpowering the actuation system.

Damage Reduction

Warning devices and emergency procedures may reduce the damage caused by a hazard, in case of ineffective hazard control measures or unidentified hazards.

6.3.2 Architecture and Item Development Assurance Approaches

From the system functions and their FDAL, the system architecture allocates hardware and software development assurance requirements as Item Development Assurance Level (IDAL), corresponding to DO-178C software level. This section discusses a set of architecture approaches relevant to consider for aircraft-specific AFCS applications.

FDAL/IDAL Assignment with or without System Architecture Consideration

ARP4754B [37] distinguishes two main approaches to assign development assurance to functions and items: *with* or *without* system architecture consideration.

Without system architecture consideration, the required top-level FDAL identified during the FHA is directly allocated to everything under that function, i.e., the same FDAL requirement applies for all functions supporting the top-level function, and the corresponding IDAL requirement applies for all items in the architecture realizing the function.

With system architecture consideration, the degree of independence within the architecture of the functions and items supporting the top-level function allows for reduced FDAL or IDAL. For

a top-level function supported by two or more independent sub-functions, it is for example possible to assign an FDAL for at least one of the sub-functions to a level lower than the top-level function FDAL.

The impact of functional and item independence on the FDAL/IDAL assignment during system architecture development may be analyzed using Functional Failure Sets (FFS). Analog to a Fault Tree Analysis, where a minimal cut set is the smallest set of basic events that together lead to a top-level event in a fault tree (see ARP4761A [38]), an FFS is the equivalent to a fault tree minimal cut set, where the tree members represent potential development errors rather than item or component failures that lead to a failure condition. A failure condition may have one or more FFS, and an FFS may have one or more members.

Functional and Item Independence

The purpose of functional independence is to minimize the likelihood of common requirement errors and common requirement interpretation errors between two or more functions realizing a top-level function. This may for example be claimed when the sets of functional requirements are different or the principles for the realization of the functions are physically different. ARP4754B [37] lists several examples, including deceleration of the aircraft on ground using wheel brakes vs. reverse thrust vs. ground spoilers and control of the aircraft in the air using flight control surfaces vs. thrust vectoring.

The purpose of item independence is to minimize the likelihood of common mode error between individually developed items. Errors mitigated by item independence include

- Software or hardware design errors (e.g., requirements errors, architecture errors)
- Software or hardware development errors (e.g., process errors, configuration control errors)
- Software or hardware tool errors (e.g., coder error, compiler error, linker error)

Item development independence may for example be claimed for:

- Dissimilar technologies (e.g., hydraulic vs. electric power)
- Dissimilar software
 - Source code implemented in different programming languages
 - Object code generated using different compilers
 - Software requirements, design, and source code developed by different development teams or in different software development environments
- Dissimilar hardware
 - Different microprocessors
 - Hardware requirements, design, and fabrication by different teams or in different development environments

Architecture Levels of Redundancy

Three principal AFCS architectures with or without functional and item independence and the effect on their development assurance levels are discussed in the following:

- Simplex architecture without functional independence
- Fail-passive duplex architecture with functional independence
- Fail-operational dual duplex architecture with functional independence

The three architectures are illustrated in Figure 6.1. Many other redundant architectures and architecture redundancy management approaches exist, for example, triplex or quadruplex architectures with cross-strapped sensor signal voting, actuator command voting, or actuator force voting. Rice et al. [112] provide an overview of principal redundant flight control system architectures.

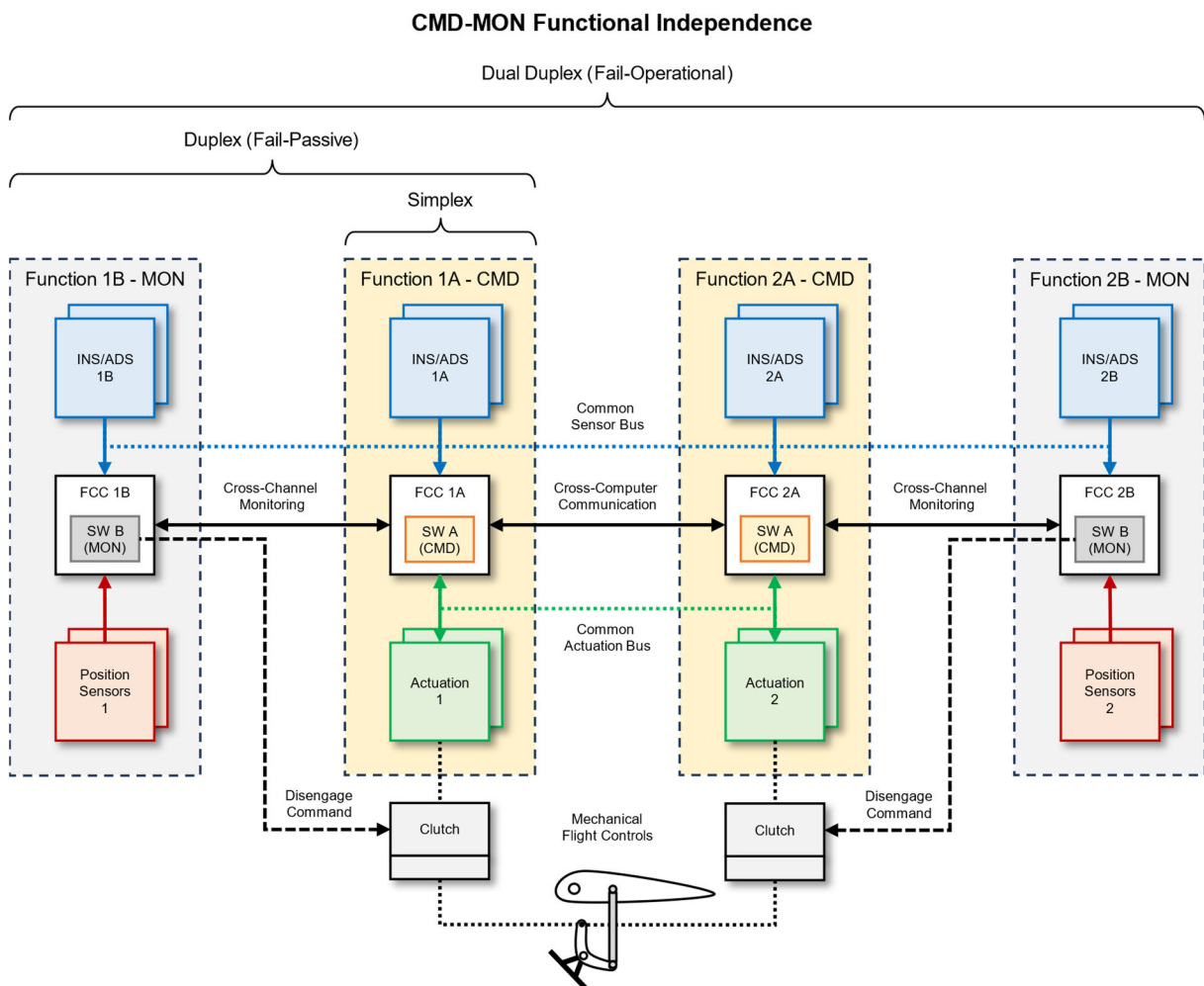


Figure 6.1: Principal simplex, duplex and dual duplex AFCS architectures.

The simplex architecture in Figure 6.1 is a single-channel system consisting of a single set of sensors, a single FCC for command calculations with input and output handling, and a single set of actuators moving the control surfaces. The architecture is susceptible to single failures; thus, this architecture is insufficient for a multi-axis AFCS with unlimited authority, in that a single failure may cause a hardover malfunction.

The duplex architecture in Figure 6.1 introduces a functional independence between the dual channels. The dual-channel system consists of two sets of sensors, dual FCC with separate

software and input/output handling, a set of actuators, and a set of control surface or actuator position pickup sensors, for independent measurement of control surface deflections. The command channel 1A provides the nominal automatic flight path control function and the monitoring channel 1B provides independent monitoring of the command channel. The two functions are implemented from different sets of functional requirements. The command channel calculates the flight control commands and outputs to the actuation system, whereas the monitor channel supervises the command channel outputs. If the monitor channel detects an error in the command channel, an automatic disengagement of the system is triggered. When the AFCS disengages, control is reverted to some primary flight control system (not shown in Figure 6.1). The duplex architecture is fail-passive, i.e., it may detect and passivate itself in the case of a single failure, preventing a catastrophic failure condition.

The dual duplex architecture in Figure 6.1 consists of two sets of duplex systems, i.e., dual command and monitoring channels as well as redundant actuator and position sensor sets. Here, the software in the command and monitoring channels, SW A and SW B respectively, are duplicated between the two duplex systems, i.e., the two command channels 1A and 2A are not considered functionally independent, likewise the two monitoring channels 1B and 2B.

Given INS/ADS, actuation, and position sensor independence, i.e., no common mode errors, the dual duplex architecture is fail-operational, i.e., one duplex channel is still operational in the case of a single failure in the other duplex channel.

In the following, the three principal architectures are analyzed with respect to their functional failure sets and valid FDAL/IDAL assignments, depending on the level of item independence. The clutch illustrated in Figure 6.1 is not included in this analysis. The FDAL/IDAL assignment is exemplified for a top-level FDAL A, the case for a multi-axis, unlimited authority AFCS in a Class IV airplane according to AC 23-1309-1E [104], see Table 6.4.

For lower authority AFCS and/or other aircraft classes, the top-level FDAL is assigned according to Table 2.3 and an FDAL/IDAL assignment follows analogously.

Simplex Architecture - Functional Failure Sets

The functional failure set of the simplex architecture is straight forward, as illustrated in Figure 6.2. In the absence of any functional or item independence or redundancy, a single functional development error or item error may lead to the catastrophic top-level failure condition.

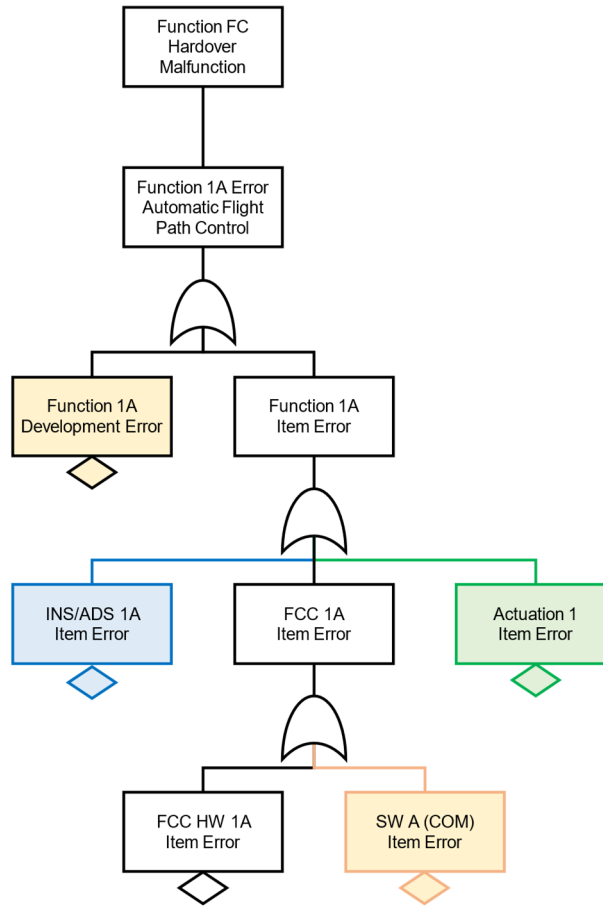


Figure 6.2: Functional fault tree analysis for the simplex architecture.

Simplex Architecture – FDAL/IDAL Assignment

The FDAL/IDAL assignment for the simplex architecture is equally straightforward. As there is no functional or item redundancy, the top-level FDAL for the system function is directly allocated to the items as IDALs, according to Table 6.5. For a top-level FDAL A, the IDAL for INS/ADS sensors, the FCC hardware, the software as well as the actuation system is A.

Table 6.5: Valid IDAL assignments for the simplex architecture for top-level FDAL A.

FDAL	IDAL			
F1A CMD	INS/ADS 1A	FCC HW 1A	SW A CMD	ACT 1
A	A	A	A	A

Duplex Architecture - Functional Failure Sets

The duplex architecture requires a double error for the top-level failure condition to occur, as illustrated in the functional fault tree in Figure 6.3. Any combination between the two channels of a function development error and/or an item error constitutes an FFS.

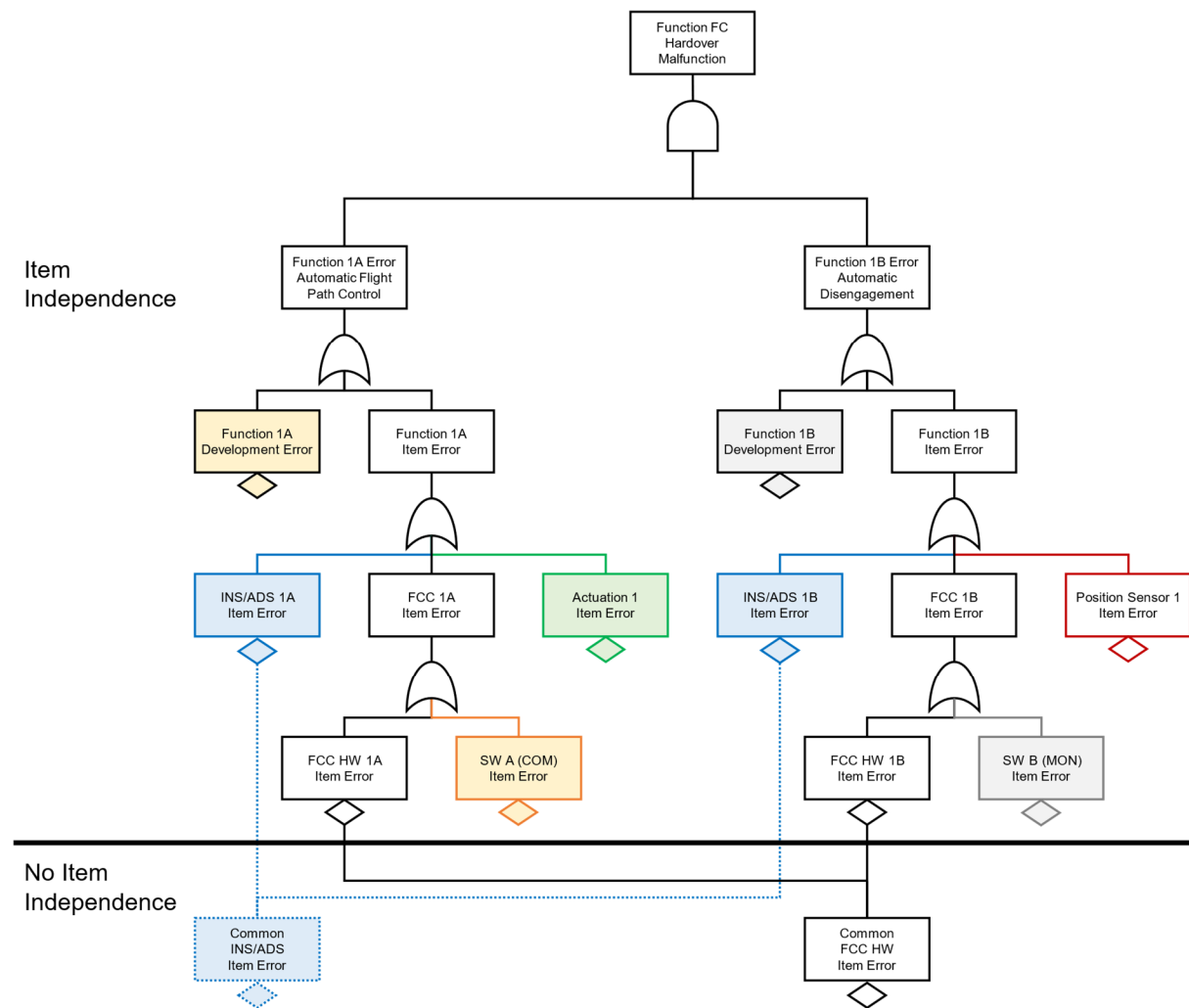


Figure 6.3: Functional fault tree analysis for the duplex fail-passive architecture.

In case of no INS/ADS independence or no FCC hardware independence, a single common mode error may cause the top-level failure condition, as seen in Figure 6.3. Thus, for the duplex architecture to be valid for a multi-axis, unlimited authority AFCS, sensor and FCC hardware independence is required to prevent a single common mode error from causing a catastrophic failure. For lower authority AFCS, where a hardover malfunction per Table 6.4 only constitutes a hazardous or major failure condition, INS/ADS and/or FCC hardware redundancy without independence may be considered.

Duplex Architecture – FDAL/IDAL Assignment

The duplex architecture with functional independence allows for the assignment of FDAL and IDAL with architecture considerations, in order to achieve a lower FDAL/IDAL for at least one of the independent channels. Table 6.6 illustrates the valid FDAL and IDAL assignment approaches for the case with INS/ADS as well as FCC hardware independence. The top-level FDAL A may be realized by either FDAL B for both functions supporting the top-level function (equals Option 2 in Table 3 in ARP4754B [37]), or FDAL A for one function and FDAL C for

the other (equals Option 1 in Table 3 in ARP4754B [37]). The items below each FDAL are assigned the identical IDAL, analog to the assignment for the simplex architecture.

Table 6.6: Valid FDAL/IDAL assignments for the duplex architecture with INS/ADS and FCC HW independence for top-level FDAL A.

FDAL		IDAL							
F1A CMD	F1B MON	INS/ADS 1A	FCC HW 1A	SW A CMD	ACT 1	INS/ADS 1B	FCC HW 1B	SW B MON	POS 1
B	B	B	B	B	B	B	B	B	B
A	C	A	A	A	A	C	C	C	C
C	A	C	C	C	C	A	A	A	A

Table 6.7 illustrates the valid FDAL and IDAL assignment approaches for the case where functional independence is claimed but no item independence, specifically no FCC hardware independence. A common FCC hardware error, illustrated in Figure 6.3 as a common FFS member for a Function 1A and Function 1B error, may cause a top-level failure condition and is thus allocated the IDAL corresponding to the top-level FDAL, in this case IDAL A, independent of the FDAL/IDAL approach.

Table 6.7: Valid FDAL/IDAL assignments for the duplex architecture with INS/ADS independence but not FCC HW independence for top-level FDAL A.

FDAL		IDAL (FCC HW 1A = 1B)							
F1A CMD	F1B MON	INS/ADS 1A	FCC HW 1A	SW A CMD	ACT 1	INS/ADS 1B	FCC HW 1B	SW B MON	POS 1
B	B	B	A	B	B	B	A	B	B
A	C	A	A	A	A	C	A	C	C
C	A	C	A	C	C	A	A	A	A

A similar assignment is valid for the case where the FCC hardware is independent, but no INS/ADS independence is claimed, as illustrated in Table 6.8. Here, the INS/ADS is allocated IDAL A as it is a common FFS member for a Function 1A and Function 1B error, as illustrated in Figure 6.3.

Table 6.8: Valid FDAL/IDAL assignments for the duplex architecture with FCC HW independence but not INS/ADS independence for top-level FDAL A.

FDAL		IDAL (INS/ADS 1A = 1B)							
F1A CMD	F1B MON	INS/ADS 1A	FCC HW 1A	SW 1A CMD	ACT 1	INS/ADS 1B	FCC HW 1B	SW 1B MON	POS 1
B	B	A	B	B	B	A	B	B	B
A	C	A	A	A	A	A	C	C	C
C	A	A	C	C	C	A	A	A	A

Dual Duplex Architecture - Functional Failure Sets

Figure 6.4 illustrates a functional failure analysis for the dual duplex architecture in Figure 6.1. Functional independence is only claimed between the command (1A/2A) and monitor (1B/2B) channels. A Function 1A/2A Development Error and Function 1B/2B Development Error are considered common events for the 1A and 2A channels, as well as for the 1B and 2B channels, respectively. The command and monitor channels exhibit the same software as well, with Software A and Software B Item Errors being common events for the FCC 1A and FCC 2A Item Errors, as well as FCC 1B and FCC 2B Item Errors, respectively.

In the case of dependent INS/ADS within channel 1 or 2 (INS/ADS 1A equals INS/ADS 1B, INS/ADS 2A equals INS/ADS 2B), a common mode error constitutes a common FFS member for Function 1A and 1B, or Function 2A and 2B, respectively, as illustrated in the partial item independence row in Figure 6.4. Similarly, in the case of dependent FCC hardware within channel 1 or 2 (FCC HW 1A equals FCC HW 1B, FCC HW 2A equals FCC HW 2B), a common mode error constitutes a common FFS member for Function 1A and 1B, or Function 2A and 2B. In the case of dependent INS/ADS between all channels (INS/ADS 1A and 1B equals INS/ADS 2A and 2B), a common mode error constitutes a common FFS member for all the Functions 1A, 1B, 2A, and 2B, is thus a single failure. The same holds for the case of dependent FCC hardware between all channels, as illustrated in Figure 6.4.

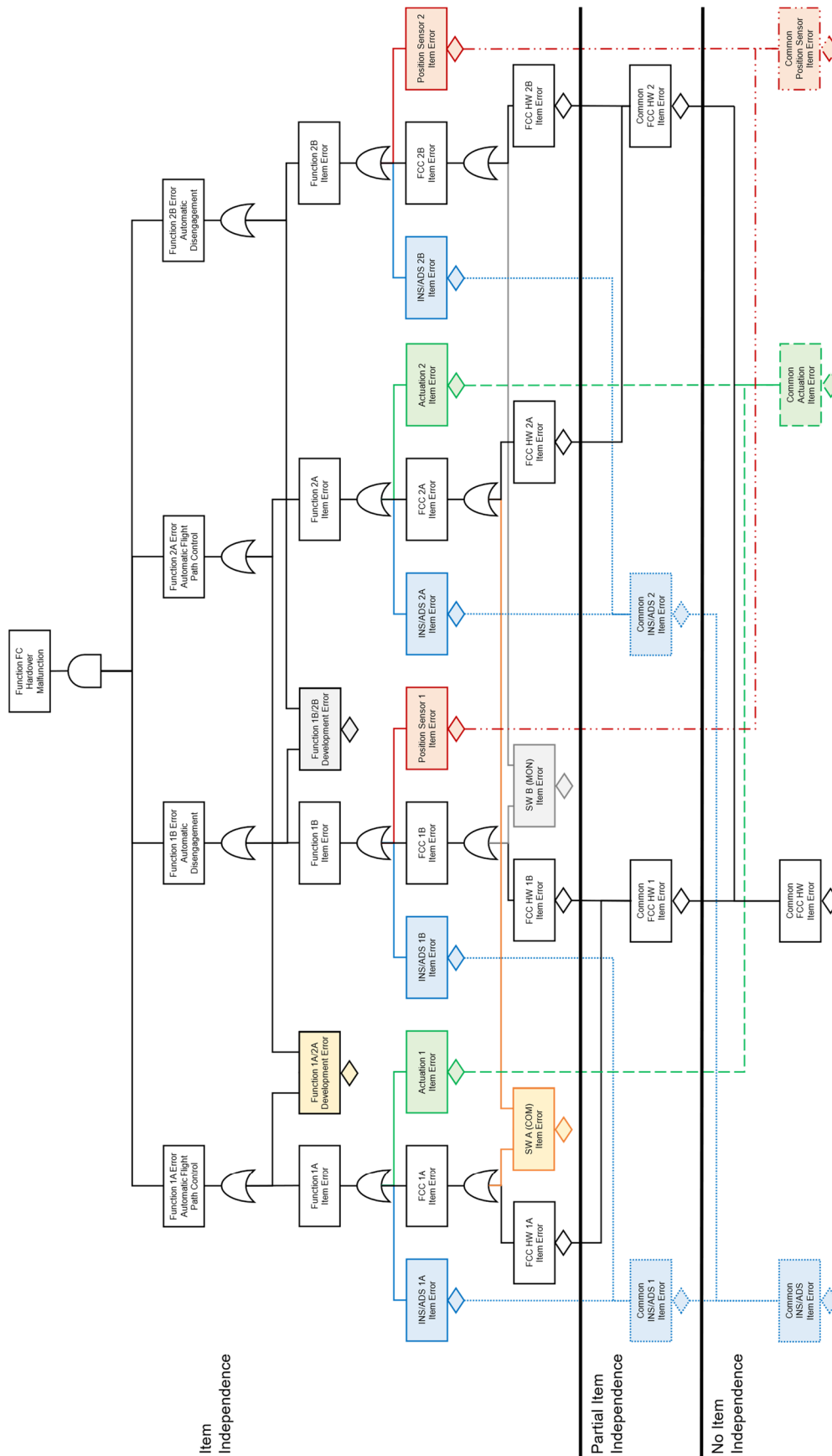


Figure 6.4: Functional fault tree analysis for the dual duplex fail-operational architecture.

Dual Duplex Architecture – FDAL/IDAL Assignment

The dual duplex architecture with functional independence allows for the assignment of FDAL and IDAL with architecture considerations, similar to the duplex architecture, in order to achieve a lower FDAL/IDAL for at least one of the independent channels.

Table 6.9 illustrates the valid FDAL and IDAL assignment approaches for the dual duplex architecture for the case with full INS/ADS as well as FCC hardware independence. The top-level FDAL A may be realized by either FDAL B for both functions supporting the top-level function (equals Option 2 in Table 3 in ARP4754B [37]), or FDAL A for one function and FDAL C for the other (equals Option 1 in Table 3 in ARP4754B [37]). The items below each FDAL are assigned the identical IDAL, analog to the assignment for the duplex and simplex architectures.

Table 6.9: Valid FDAL/IDAL assignments for the dual duplex architecture with full INS/ADS, FCC HW, actuation system and position sensor independence for top-level FDAL A.

FDAL		IDAL													
F1A/F2A CMD	F1B/F2B MON	INS/ADS				FCC HW				FCC SW		ACT		POS	
		1A	1B	2A	2B	1A	1B	2A	2B	A	B	1	2	1	2
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
A	C	A	C	A	C	A	C	A	C	A	C	A	A	C	C
C	A	C	A	C	A	C	A	C	A	C	A	C	C	A	A

Table 6.10 illustrates the valid FDAL and IDAL assignment approaches for the dual duplex architecture for the case with dependent INS/ADS within channel 1 or 2 (INS/ADS 1A equals INS/ADS 1B, INS/ADS 2A equals INS/ADS 2B), where a common mode error constitutes a common FFS member for Function 1A and 1B, or Function 2A and 2B, respectively. Thus, the INS/ADS is assigned an IDAL corresponding to the FDAL of the function at the highest-level failure condition that the common mode error may cause, which in this case is either Function 1A/2A or Function 1B/2B (and not the top-level function as for the duplex architecture). If the FDAL assignment follows Option 2 in Table 3 in ARP4754B [37], with FDAL B for the two independent functions F1A/F2A and F1B/F2B, the corresponding INS/ADS 1A/2A and INS/ADS 1B/2B IDAL equals B. If the FDAL assignment follows Option 1 in Table 3 in ARP4754B [37], with FDAL A for one function and FDAL C for the other, the corresponding INS/ADS 1A/2A as well as INS/ADS 1B/2B IDAL equals A, as this corresponds to the highest-level failure condition that a common mode error may cause.

Table 6.11 illustrates the valid FDAL and IDAL assignment approaches for the dual duplex architecture for the case with dependent FCC Hardware within channel 1 or 2 (FCC HW 1A equals FCC HW 1B, FCC HW 2A equals FCC HW 2B), where a common mode error constitutes a common FFS member for Function 1A and 1B, or Function 2A and 2B, respectively. The FDAL and IDAL assignment for the FCC hardware elements follows

analogously to the case with partially independent INS/ADS described in Table 6.10. The FCC hardware is assigned an IDAL corresponding to the FDAL of the function at the highest-level failure condition that the common mode error may cause, which is either Function 1A/2A or Function 1B/2B.

For the case where all INS/ADS or all FCC Hardware are dependent, a common mode error would cause a functional failure in all four channels and thus the top-level failure condition, as illustrated by the Common INS/ADS Item Error base event in the functional fault tree in Figure 6.4. The independence of the actuation systems or the position sensors does not affect on their respective IDAL assignment, as they are always assigned the IDAL of the command or monitor channel FDAL, respectively.

Table 6.10: Valid FDAL/IDAL assignments for the dual duplex architecture with partial INS/ADS independence for top-level FDAL A.

FDAL		IDAL (INS/ADS 1A = 1B and 2A = 2B)													
F1A/F2A CMD	F1B/F2B MON	INS/ADS				FCC HW				FCC SW		ACT		POS	
		1A	1B	2A	2B	1A	1B	2A	2B	A	B	1	2	1	2
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
A	C	A	A	A	A	A	C	A	C	A	C	A	A	C	C
C	A	A	A	A	A	C	A	C	A	C	A	C	C	A	A

Table 6.11: Valid FDAL/IDAL assignments for the dual duplex architecture with partial FCC HW independence for top-level FDAL A.

FDAL		IDAL (FCC HW 1A = 1B and 2A = 2B)													
F1A/F2A CMD	F1B/F2B MON	INS/ADS				FCC HW				FCC SW		ACT		POS	
		1A	1B	2A	2B	1A	1B	2A	2B	A	B	1	2	1	2
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
A	C	A	C	A	C	A	A	A	A	A	C	A	A	C	C
C	A	C	A	C	A	A	A	A	A	C	A	C	C	A	A

6.3.3 Primary and Secondary AFCS

Section 6.3.2 discusses the implications of different levels of redundancy and the effect of function and item independence on the FDAL/IDAL assignments. AC 23-1309-1E [104] provides guidance on the classification of redundant systems in the form of secondary systems, defined as redundancy systems that provides the same function as the primary

system [104, p. 13]. Secondary systems may be installed to meet probability requirements for failure conditions associated with loss of function. AC 23-1309-1E does not explicitly state that secondary systems shall be independent of the primary systems, although it can be inferred from their purpose of reducing failure condition probabilities, that independence is required in order for the secondary systems to constitute basic events in a corresponding fault tree analysis. Examples include a PFD as the primary system for aircraft state awareness and a backup instrumentation as a secondary system. The FDAL for a function of a secondary system, if installed, is assigned according to Table 2.3.

For automatic flight control systems, secondary systems may be realized as some simplified alternate or backup control law designed to be more reliable and robust than the primary system. For the AFCS for the SAGITTA Research Demonstrator, a backup law was designed and implemented with the purpose of increasing the survivability of the aircraft in the case of a loss of the primary flight control computer channel. The backup AFCS utilized a reduced sensor set and was designed around simpler attitude-based control laws not requiring the kinematic flight path information from a GNSS-aided INS, only relying on AHRS and IMU sensor data. The SAGITTA is a research platform, thus a formal FDAL/IDAL assignment process for the primary/backup AFCS was not performed. The detailed design and implementation of the AFCS backup law for the SAGITTA Research Demonstrator is outside the scope of this thesis and is not elaborated here.

AC 23.1309-1E Appendix 1 does not provide any classification guidance on the loss of primary vs. secondary AFCS, only for total loss of function and malfunction. Thus, the addition of a secondary AFCS only serves the objective of meeting the probability requirements associated with the loss of function and does not impact the FDAL assignment of the primary system or the classification of the failure condition for the loss of the primary system.

7 System Realization

This chapter describes the system realization activities of the AFCS development, according to the system realization track of the System Development Process defined in Section 2.3.4:

- Design Model Implementation and Integration
- Source Code Generation and Software Integration
- Hardware-Software Integration
- Aircraft Integration

The system realization track is where the system design is implemented and integrated, i.e., realized in terms of software and physical hardware components. The system implementation and integration follow through the system levels, from design model to aircraft integration, according to the structural view of the modular FGCS introduced in Figure 2.1, Section 2.1.1:

- *Application level*, i.e., the design model implementing the flight control algorithms and their interfaces
- *Software level*, i.e., the application software generated from the design model, the software framework, and the hardware-close drivers and interface handling
- *Item level (target computer or hardware-software integration level)*, i.e., software deployed on the embedded hardware
- *System level*, i.e., the target computer with sensors, mechanical interfaces, and actuation system
- *Aircraft level*, i.e., the AFCS with the airframe and the aircraft dynamics, propulsion system dynamics

7.1 Design Model Implementation and Integration

The development of the modular FGCS of which the AFCS is part is a team effort involving a large number of team members, developing and implementing the flight control algorithm modules in parallel, as described in the introductory chapter, Section 1.3.3. Thus, the need for a model-based development approach for safety-critical software that supports this distributed development was needed. The model-based development further takes the approach that the flight control algorithm model developed at the system level also serves as the design model for the automatic source code generation and software integration without any further modifications.

Hochstrasser [98] describes a modular development process and a modeling framework for safety-critical model-based software development in Simulink/Stateflow (SL/SF) that translates and concretizes the activities and objectives in the DO-178C/DO-331 process framework to such a modular design approach. The model-based development process and modeling framework were developed to a large extent in parallel to the development of the flight control algorithm modules of the modular FGCS and the AFCS described in this thesis. Thus, the AFCS design model implementation and integration activities largely adhere to the development process and modeling framework described in [98]; however, no claim of full conformance is made here, as the framework was a work in progress at the time of implementation.

The model-based design process breaks down the software architecture into *components* developed either according to DO-178C or model-based [98, p. 46]. Model-based components are further broken down into *modules*; modules consist of *units* and *module data*. Units contain operations that implement parts of the functional algorithms and logic, using module data such as constants or type definitions.

The design models for the flight control algorithm modules of the modular FGCS functional architecture, shown in Figure 1.1, are implemented directly from the functional design at the system level. The AFCS design model is integrated with interfacing flight control algorithm modules to a *component* design model for the FCC application (the FCC design model), that is integrated into different test harness models for model-level verification and validation.

The model-based development process describes the sub-process for modules, which are developed and implemented independently as separate configuration items. The process supports workflow example 5 in DO-331, Table MB.1-1 [49], which supports the special use case of flight control laws in software development. From the perspective of DO-331, the design model constitutes the software architecture and the software low-level requirements (LLR). In this workflow approach, the model is shared between the system and the software development process, with different application purposes. The system-level model supports the linearization and application of linear systems theory to verify control algorithms, parameter optimization, and model-in-the-loop simulations to verify and validate the system-level functionality and performance. The software design model reflects how the software works (i.e. the LLR), holds all information to generate standard-compliant code for a specific target, provides verifiability at the software level, and fulfills the DO-178C process [98, p. 45].

The modeling framework consists of *design rules*, *coding rules*, *module design rules*, *fundamental modeling rules*, and *traceability rules* [98, p. 78]. Design rules define functional and process concepts and requirements for the software design that are independent of the chosen modeling language. Examples include architectural aspects such as the scope and definition of software modules, module interface approaches, required encapsulation, unintended functionality, naming conventions, technical units, and mathematical algorithm accuracy. Coding rules define aspects related to the target computer and the compiler, and describe the language set and structure of the auto-generated code. Module design rules define a safe modeling subset in SL/SF and transform the design rules regarding for example

architectural design to the SL/SF-specific context. The fundamental modeling rules build upon existing modeling guidelines, like the MathWorks guidelines for high-integrity systems. Traceability rules describe the creation and granularity of necessary trace links, as well as the handling of derived requirements in SL/SF. For the implementation, the AFCS design model, early iterations of the design rules, module design rules and fundamental modeling rules were adhered to, whereas the traceability rules were only rudimentary explored. The coding rules primarily concern the code generator, compiler, and target hardware, and are thus less addressed by the model implementation but by the source code generation process.

7.1.1 Implementation and Integration Environment

The design model as well as the integration and test environment were implemented in MathWorks MATLAB/Simulink version 2013b.

Configuration management and version control were performed using the software *Git*. Each software module is a configuration item, including the AFCS module. Version control at the module level was managed within a module-specific repository. A main integration repository included the functional module repositories as submodules, together with repositories for common library blocks, bus objects, flight dynamics model, actuation system models, and sensor models.

The FCC model contains the application software; the framework software that handles the real-time operating system and the communication with the physical bus interfaces is programmed directly in C code and is not covered by the model-based development process referenced in this section. The application software consists of three main parts: the input handling, the functional algorithms (i.e., the flight control algorithm modules), and the output handling, as illustrated in Figure 1.1. The input handling monitors the input signal validity and quality criteria. Consolidated input signals are then passed to the flight control algorithm modules, omitting redundant basic input monitoring at the module level. The output handling formats the output data from the flight control algorithms and controls when and on which bus the data is written.

The functional algorithm part of the FCC model integrates the different flight control algorithm modules, such as the AFCS, the inner loop controller, system automation, and the automatic landing and takeoff modules. The principal architecture is shown as the application layer in Figure 1.1. As described above, each flight control algorithm module is implemented as an individual configuration item. The FCC model uses extensive *model referencing* to integrate the control modules. The FCC model for the DA42 OPV use case at the time of writing contains 129 model references, which in turn consist of over 100,000 individual blocks in total [87].

Several integration environments with test harness models were developed to support different aspects of integration testing and formal verification and validation. In the following, three different integration test environments are introduced, each detailed in subsequent sections of this chapter.

The first integration test environment is a set of library block test harnesses for unit-level library block testing, further described in Section 7.1.3. This integration test environment supports the

automated unit testing of the more complex units of the AFCS module, such as the distributed mode control logic.

The second integration test environment is the FCC design model-in-the-loop test harness, further described in Section 7.1.4. This is the main integration test harness for the overall modular FGCS development and contains the FCC design model together with the FDM, sensor models, actuation system models, mode control panel and monitoring display models, and visualization.

The third integration test environment is the AFCS design model-in-the-loop test harness for integration testing of the AFCS design model and nonlinear control law assessment, further described in Section 7.1.5.

7.1.2 AFCS Design Model Implementation

The AFCS design model is implemented as a separate configuration item according to the model-based development process. The layout and structuring of the design model are based on the functional architecture considerations in Section 5.2.1. Figure 7.1 shows the top level of the AFCS design model, named *af_nl*, that is referenced in the FCC model.

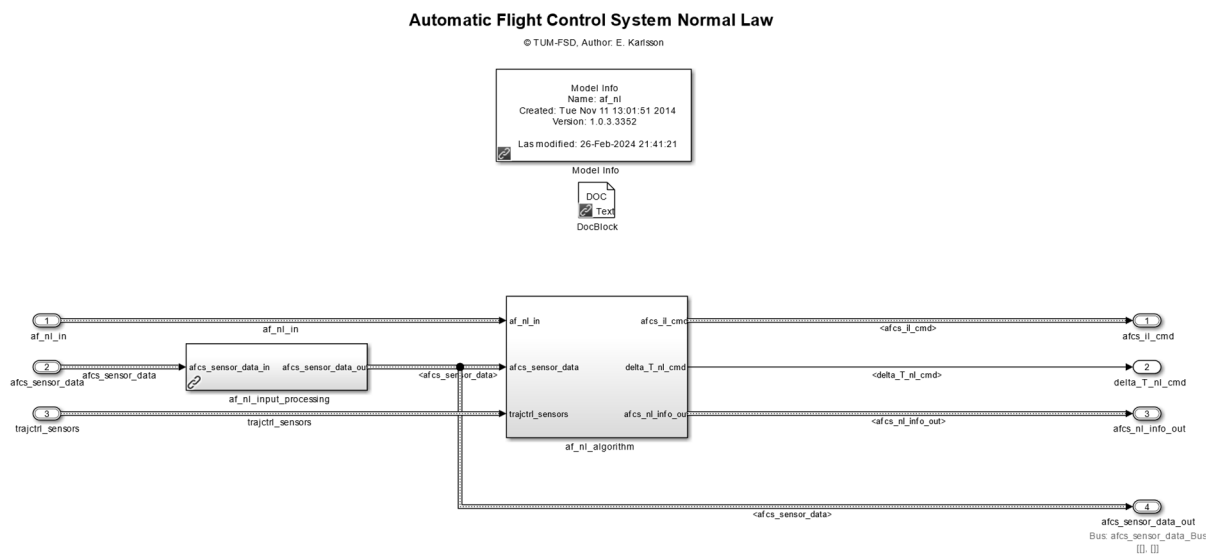


Figure 7.1: AFCS top-level model.

The top level consists of an input processing block and the main algorithm block. The input processing block contains the AFCS-specific sensor and input signal filtering and mapping of sensor signals onto an AFCS internal sensor bus utilized by the individual control modes. The AFCS-specific input signal filtering encompasses:

- Calculation of specific forces in the kinematic frame according to Eq. (5.16)
- Plant response estimations according to Eqs. (5.17)-(5.22)
- Filtering of the plant response estimations according to Eq. (5.23)
- Complementary filtering of indicated airspeed according to Eq. (5.65)

The top-level model includes metadata such as version number and modification date, as well as a short block description. The design model in- and outputs, defining the software module in- and outputs, are specified as non-virtual bus objects, with data types and structure of data fully specified for all module external interfaces.

The main algorithm block, shown in Figure 7.2, consists of six main blocks, corresponding to the elements of the principal functional architecture described in Section 5.2.1 and visualized in Figure 5.10:

- Limits and Protection Mode Control
 - Calculates the active limits based on desired energy rate and force prioritization and protection modes (described in Sections 5.2.8 and 5.2.9)
- Mode Control and Control Laws (Kinematic Frame)
 - Contains the nominal flight path and additional control mode control laws and distributed mode control logic (described in Sections 5.2.3-5.2.7) as well as the embedded trajectory controller (described in Section 5.2.6)
- Command Selection (Kinematic Frame)
 - Selects the output incremental specific forces in the kinematic frame, based on the active speed, lateral and vertical control modes, respectively
- Command Transformation and Limitation (Body Fixed Frame)
 - Transforms the kinematic frame commands to body fixed frame commands for the inner loop controller (described in Section 5.2.10)
- Autothrust Controller
 - Contains the autothrust control laws and moding logic (described in Section 5.2.11)
- Output Information Consolidation
 - Consolidates the status information from all individual control modes and onto an AFCS status output bus

The functional algorithms and logic are implemented as library blocks, in order to facilitate individual unit-level library block testing (see Section 7.1.3), and reuse of recurring algorithms such as filters and logics common among the control modules. Each library block is given a unique name, limited to 11 characters, according to the block naming convention of the modeling framework. Each signal representing a physical variable is annotated with technical units according to the modeling framework design rules. The AFCS design model consists of 274 library blocks, of which are 245 unique and AFCS specific, i.e., implementing functions and logic specific to the AFCS. The AFCS design model further contains 118 links to 7 generic blocks in control module common libraries, such as angle flip blocks, integrator blocks, protected division blocks, and saturation blocks. In total, the AFCS design model contains around 9600 individual block items.

Figure 7.3 shows an example of the implementation structure of the energy-based prioritization, with the encapsulated calculations of the acceleration and flight path angle limits according to Section 5.2.8, as well as output mode logic based on protection activation flags and external limits. Figure 7.4 shows an example of the implementation of the algorithm for the

calculation of the energy-based acceleration limits according to Eqs. (5.93)-(5.94). Each block implementing a part of the algorithm includes a documentation block with a textual description of the implemented part. Module data such as AFCS specific parameters and globally defined parameters are colored blue and easily identified.

Figure 7.5 shows an example of the distributed mode control logic, with the implementation of the altitude capture/hold and open climb/descent logic, with mode state transitions, transition criteria, and transition actions visualized in Figure 7.6 and Figure 7.7.

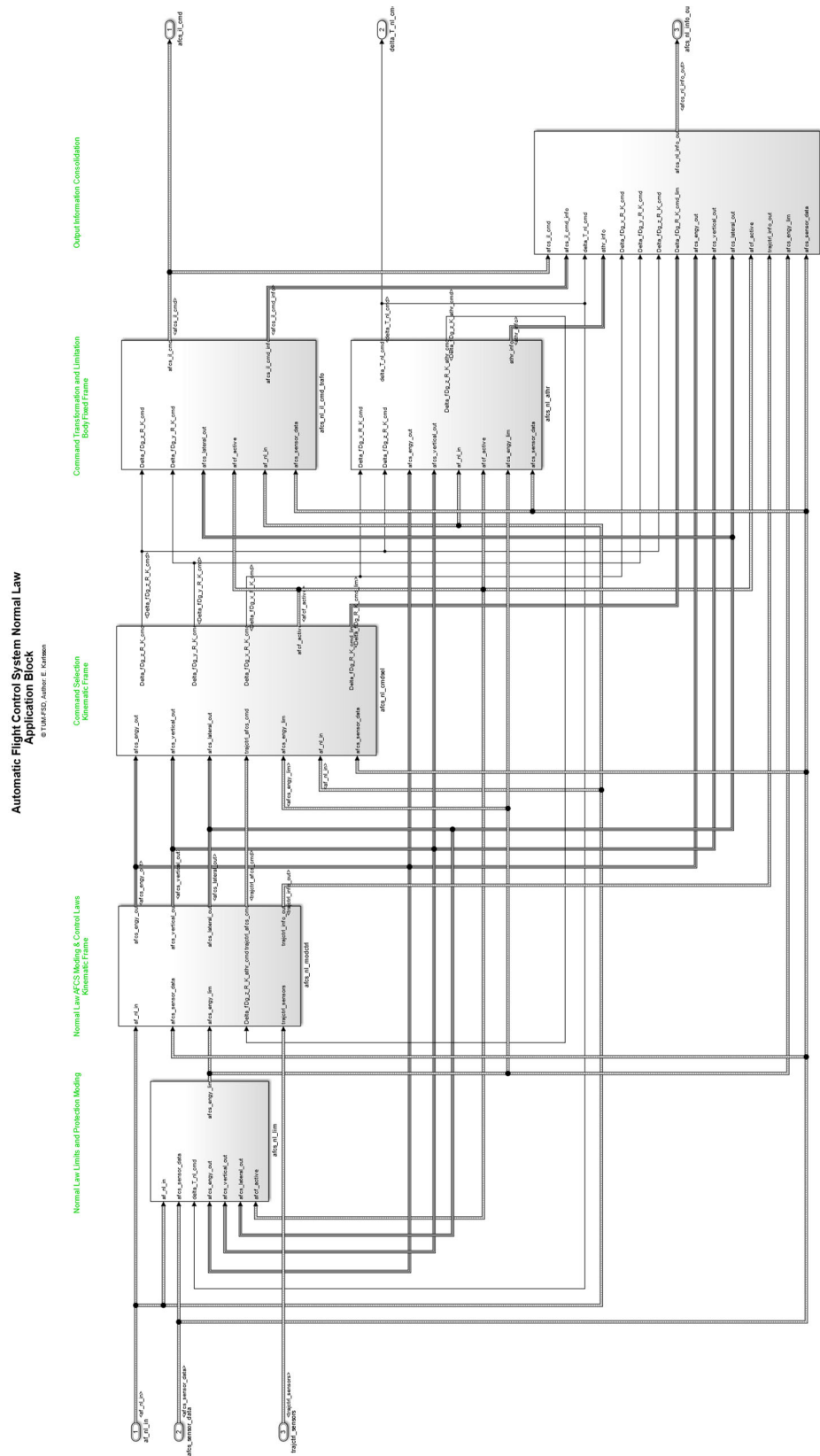


Figure 7.2: AFCS main algorithm block.

Energy Distribution Prioritization

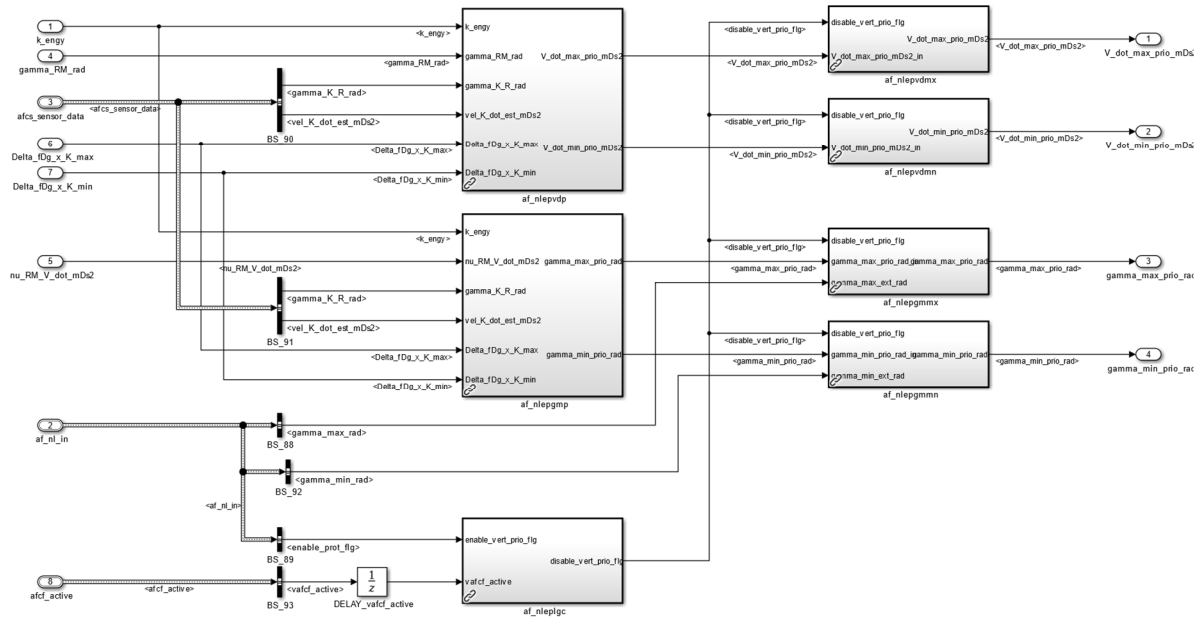


Figure 7.3: AFCS algorithms implemented as individual library blocks linked in the main model.

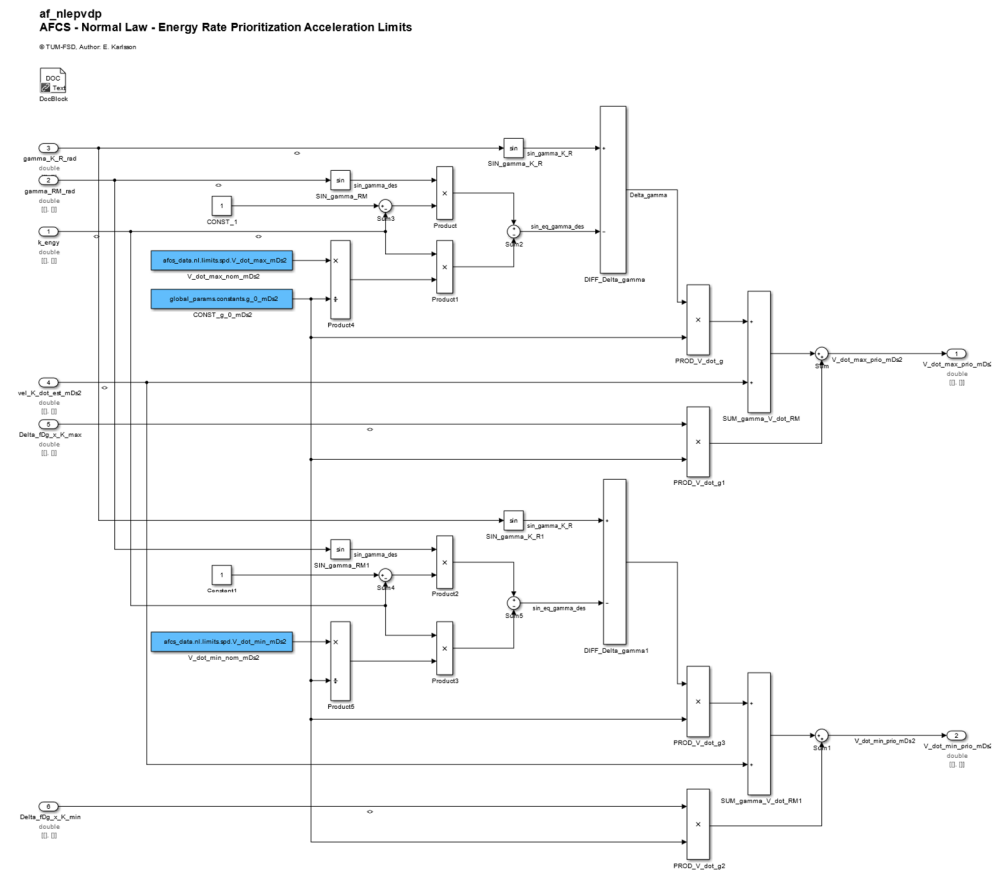


Figure 7.4: AFCS algorithm implementation example: calculation of maximum acceleration for speed or vertical path prioritization, according to Eqs. (5.93)-(5.94).

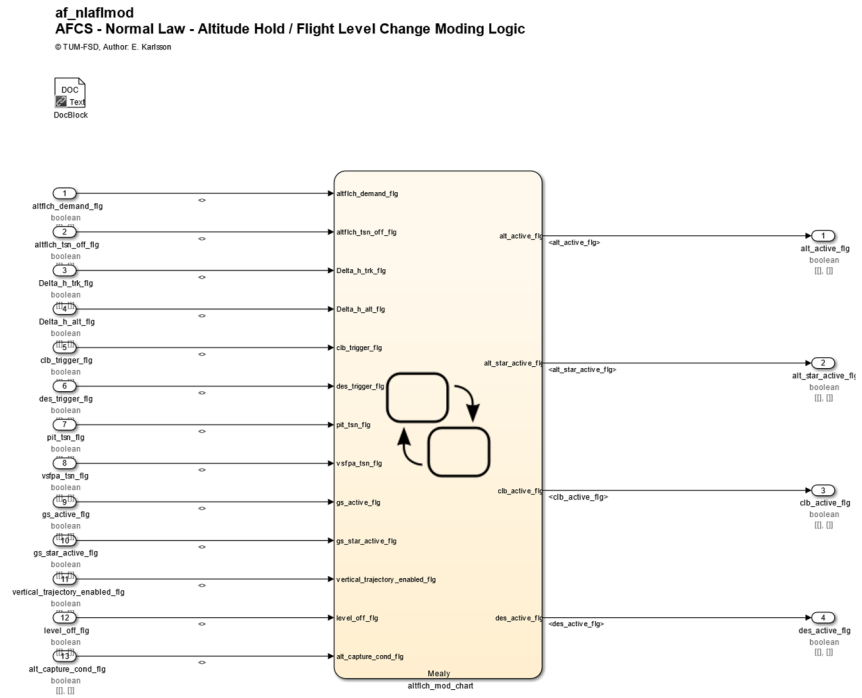


Figure 7.5: AFCS algorithm implementation example: library block containing the altitude capture/hold and flight level change mode control logic, as part of the distributed mode control logic of the AFCS.

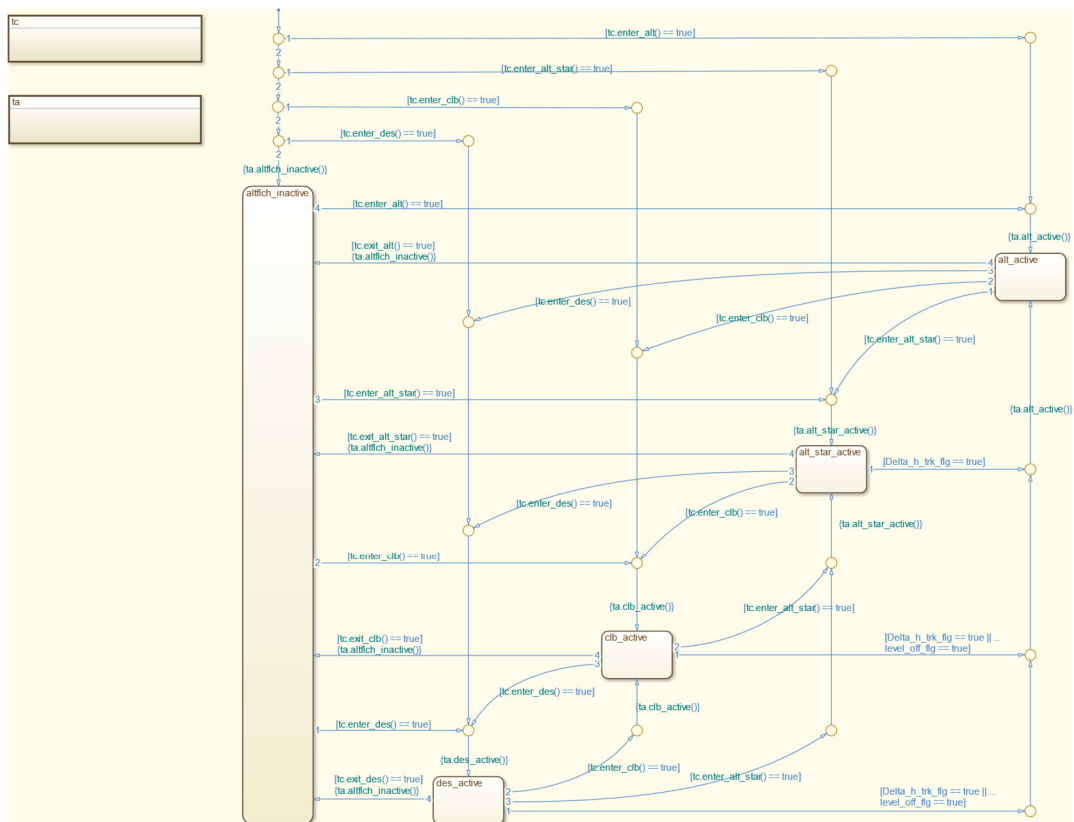


Figure 7.6: AFCS algorithm implementation example: local flight level change and altitude capture/hold logic, as part of the distributed mode control logic of the AFCS.



Figure 7.7: AFCS algorithm implementation example: transition criteria and transition action functions as part of the flight level change and altitude capture/hold logic.

7.1.3 Library Block Unit Testing

Integration tests at the library block level were performed using MATLAB *Unit Test Framework*. For each library block under test, a test suite is defined, that contains a *main function* and *local test functions*. The main test function is the top-level function that contains all functions associated with the test flow and is the one called to execute the tests. A set of local test functions defines the library block-specific test cases, with one function for each test case. The test suite also includes *setup* and *teardown functions*, in the framework referred to as *test fixture functions*, to define and set up the pretest state of the system and return it to the original state after running the test. A general setup, a so-called *file fixture*, is performed for all test cases. Test case-specific setups and teardowns, so-called *fresh fixtures*, set up test case-specific data and processes the results of each test case. The generic test flow is illustrated in Figure 7.8.

Unit testing of library blocks was concentrated on the blocks containing the more complex algorithm implementation. Many library blocks implement simple algorithms or parts of logic that were verified as part of model reviewing and integrated model tests.

For the library blocks subject to unit testing, block-unique test harness models were created, one for each block. The test harness models contain an inport bus with a bus object specifying the input signals, a link to the block under test, and outputs with signal logging for automated verification. The generic test harness model setup for library block unit testing, with the main

test function and local test functions executing the test harness model, is illustrated in Figure 7.9.

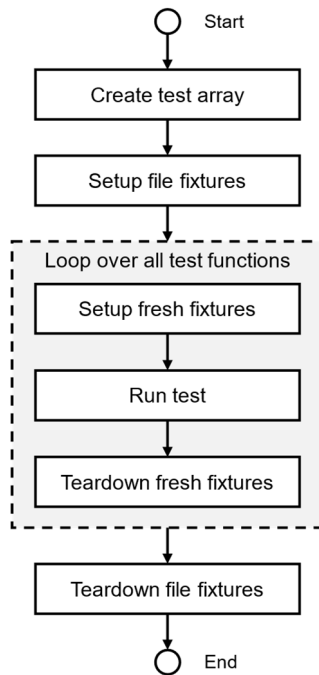


Figure 7.8: Generic MATLAB Unit Test Framework test flow, used to automate library block testing as well as model-in-the-loop and nonlinear assessment tests.

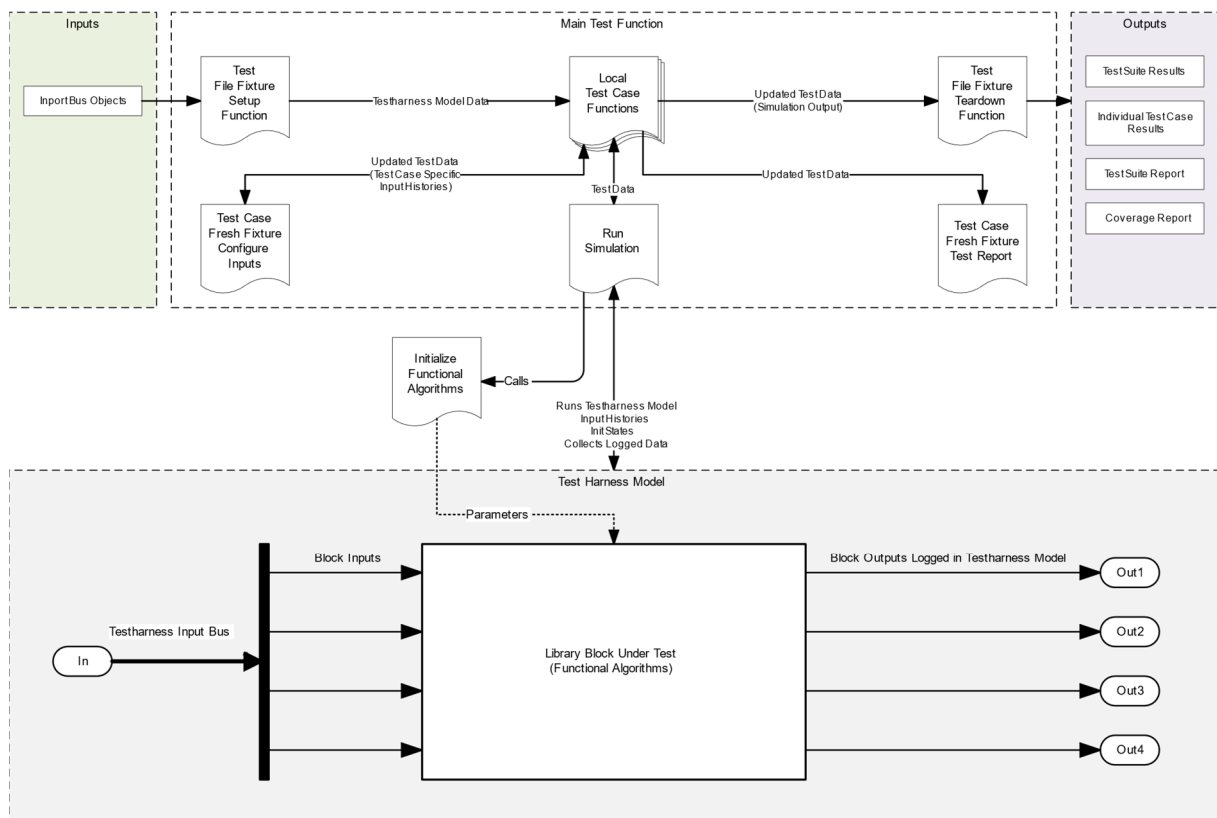


Figure 7.9: Simulink test harness model structure for library block unit testing, with main test function and local test functions executing the test harness model.

The main function, named `<block>_testscript`, defines the test harness model and the meta information for test documentation, local test functions define a set of tests for desired test coverage, and setup and teardown functions initialize the block and produce plots and test reports. An example model, the test harness for the altitude capture/hold and flight level change logic block, named `af_nlafimod`, described in Figure 7.5-Figure 7.7 is pictured in Figure 7.10.

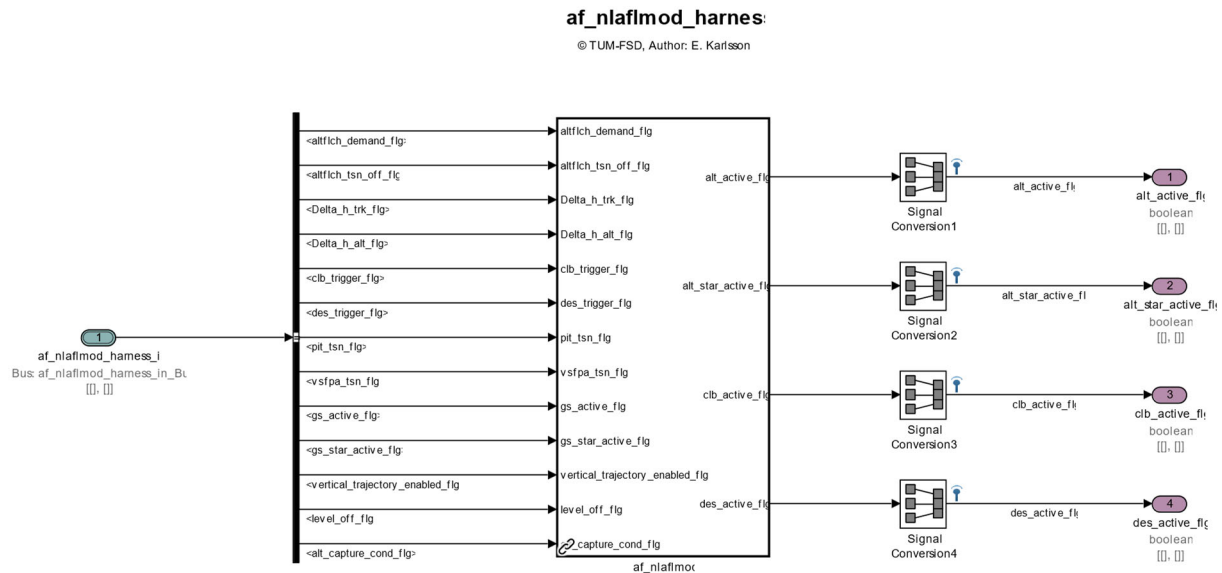


Figure 7.10: Simulink test harness model structure example for library block unit testing, here the altitude hold/flight level change logic block.

The test framework includes a model coverage analysis using *Simulink Coverage* at the unit level. *Simulink Coverage* produces a detailed coverage report for each block and its included child blocks, showing how much of the model has been exercised, indicating the model complexity, decision, condition, and modified condition/decision coverage (MCDC).

For the mode logic blocks, decision coverage means that each transition to and from every chart element has been evaluated as true and false by the test suite. Full condition coverage means that each condition at every decision has been evaluated as true and false. MCDC further tests the independence of each condition, that each condition independently affects the true and false output of every decision. For full decision coverage of the altitude capture/hold and flight level change logic block described in Figure 7.5-Figure 7.7, a total of 35 test cases was required. An excerpt of the coverage report is shown in Table 7.1.

The Unit Test Framework also produces a summary table of the test results for the block under test, which is included in the automatically generated test report. The result table for the `nlafimod` block is shown in Table 7.2.

Table 7.1: Excerpt of library block unit test coverage report using *Simulink Coverage*, with summary of decision, condition, and modified condition/decision coverage for the tested block and its child elements.





































Coverage Report for af_nlaflmod_harness (excerpt)							
Summary							
Model Hierarchy/Complexity		Test 1					
		Decision		Condition		MCDC	
1. af_nlaflmod	86	100%		76%		50%	
2.... altflch_mod_chart	85	100%		76%		50%	
3..... SF: af_nlaflmod/altflch_mod_chart	84	100%		76%		50%	
4..... SF: tc	54	100%		76%		50%	
5..... SF: enter_alt	7	100%		79%		43%	
6..... SF: enter_alt_star	4	100%		100%		100%	
7..... SF: enter_alt_clb	5	100%		100%		100%	
8..... SF: enter_alt_des	5	100%		100%		100%	
9..... SF: exit_alt	8	100%		63%		25%	
10..... SF: exit_alt_star	7	100%		79%		57%	
11..... SF: exit_clb	9	100%		61%		22%	
12..... SF: exit_des	9	100%		61%		22%	

Table 7.2: Test case execution summary table for the *af_nlaflmod* block, with test cases included in the test script *af_nlaflmod_testscript*.

Name	Passed	Failed	Incomplete
af_nlaflmod_testscript/testActivationAlt	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testActivationAltStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerClimb	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitClimbGSStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitClimbGS	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitClimbVNAV	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerDescent	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitDescentGSStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitDescentGS	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitDescentVNAV	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerAltStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitAltStarGSStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitAltStarGS	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testInhibitAltStarVNAV	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerLevelOff	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerClimb2LevelOff	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTriggerDescent2LevelOff	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransClimb2AltStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransDescent2AltStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAltStar2Alt	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAlt2Pit	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAlt2VSFPA	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAlt2Descent	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAlt2Climb	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAlt2AltStar	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAltStar2Pit	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAltStar2VSFPA	TRUE	FALSE	FALSE

Name	Passed	Failed	Incomplete
af_nlaflmod_testscript/testTransAltStar2Climb	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransAltStar2Descent	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransClimb2Pit	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransClimb2VSFPA	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransDescent2Pit	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransDescent2VSFPA	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransClimb2Descent	TRUE	FALSE	FALSE
af_nlaflmod_testscript/testTransDescent2Climb	TRUE	FALSE	FALSE

7.1.4 FCC Design Model-in-the-Loop

For each of the demonstration platforms, an integration environment and model-in-the-loop simulation for system-level verification and validation were developed. The integration environment and model-in-the-loop simulation for the DA42 OPV use case are described in Zollitsch et al [87]. Similar integration environments and model-in-the-loop simulations were set up for the Do-288 D-CODE experimental autopilot system and the SAGITTA Research Demonstrator flight control system. The author contributed to the model-in-the-loop simulation with the application-specific AFCS model and corresponding initialization scripts, and the desktop implementations of the mode control panel and mode control and monitoring displays described in Section 5.4.

The integration environment consists of a main integration repository, with the flight control algorithm module repositories as submodules, and a test harness model for model-in-the-loop simulation. Version control and change management of the model integration and the specific control modules are performed within each repository. The test harness model contains the FCC design model in the loop with a flight dynamics model, sensor models, and actuation system models, as well as models of external interfaces and visualizations.

The system integration test harness, as well as the AFCS module test harness described in Section 7.1.5, are hybrid multi-rate models that combine model elements in continuous-time as well as discrete-time. The models utilize different sample rates to represent the characteristics of the specific model elements. The FCC model and actuator models are integrated into the test harness model as referenced models. Reference models have their own configuration settings that specify the equation solver and sample rate. The FCC model runs at a sample rate of 100 Hz (as the real-time target computer executes code at discrete time steps), whereas the flight dynamics model that simulates the airframe and subsystem physics requires a much higher sampling rate. The topmost integration model is required to utilize a sample rate that is an integer multiple of the sample rate of the reference models. For the flight dynamics model and the overall integration test harness model, a sample rate of 2000 Hz is used. Rate transition blocks are introduced at the interfaces between models of different sample times, to up-sample the outputs from the FCC before entering the FDM and actuator models, and to down-sample the inputs before entering the FCC model.

The test harness model is initialized in two steps. A *base initialization* loads configuration settings, bus objects, and default variables for all flight control algorithm modules and FDM, sensors, and actuation system models. A *mission initialization* loads a mission flight plan with

waypoints, a mission profile like general in-air initialization, in-air initialization for activation of the automatic landing system, or an on-ground initialization for activation of the automatic takeoff system [87], and corresponding trim data before initializing the FDM and actuation system at the associated trimmed flight condition.

In the SL/SF model-in-the-loop environment, any variable of interest may be logged, providing a rich data set for analysis and verification or validation purposes. When deployed on the target hardware, besides the logging of the bus communication, the logging of internal FCC software variables is limited to 20 variables of data type *single* and four variables of data type *uint8* per time step via a serial interface. Thus, the model-in-the-loop simulation environment is essential to provide test data and verify the internal behavior of the flight control system, building confidence before hardware-in-the-loop simulation and flight testing.

When run in *Accelerator mode*, the test harness model supports soft real-time execution using the *Simulation Pace* block of the *Aerospace Toolbox* for “desktop flight testing.” Desktop versions of the MCP and MCMD implemented as external executables interface with the test harness model via UDP, greatly supporting control law prototyping and early validation of system operation. The MCP and MCMD designs were iteratively improved based on test pilot feedback, ensuring design maturity before hardware manufacturing and HIL integration. The MCMD together with external visualization software including the open-source software *FlightGear*, the commercial software *XPlane*, and a custom-developed 2D moving map provide situational awareness and intuitive monitoring of the control system and aircraft behavior during testing. The 2D map can display the active waypoints and provides intuitive monitoring of lateral trajectory tracking performance. The external visualizations are similar to the MCP and MCMD connected via UDP ports fed from the test harness model.

7.1.5 AFCS Design Model-in-the-Loop

A separate integration test harness model was developed for the automation of AFCS nonlinear assessment and model-in-the-loop verification, based on the same flight dynamics, sensor, and actuation system models and architecture as the system-level test harness model described in Section 7.1.3.

The integration test harness was developed concurrently with the AFCS controller design and design model implementation, enabling early control law validation using a simple point mass kinematics model for the flight dynamics and transfer functions for assumed closed inner loop performance. The integration test harness evolved to include the full AFCS design model and the full six-degree-of-freedom flight dynamics model with sensor and actuation system models. The last iteration of the AFCS test harness uses a modified FCC model with only the required subset of flight control algorithms necessary to verify the AFCS functionalities, i.e., the AFCS module, the embedded trajectory controller, and the inner loop controller, as well as associated input conditioning and sensor filtering. The AFCS test harness model inputs and outputs allow for direct commands to the AFCS interfaces, representing a bypass of the omitted system automation and higher-level logic.

The AFCS model-in-the-loop test harness utilizes the MATLAB Unit Test Framework similar to the library block unit testing described in Section 7.1.3. Where the library block unit tests required one test harness model and test suite for each library block, the AFCS design model-in-loop test suite uses only one test harness model. The implemented test flow for the AFCS, illustrated in Figure 7.11, is summarized as follows:

1. The main function creates an array of tests specified by the local test functions
2. The setup file fixture function creates a test data structure:
 - a. Defines the test suite name, date, and folder paths
 - b. Defines the name and sample time of the test harness model
 - c. Loads trim data set
 - d. Defines the test condition and configuration index based on the loaded trim data set, i.e., the altitude/speed envelope and the aircraft configuration subject to test
 - e. Loads test harness model-specific bus objects
3. Each local test function defines the test case-specific initial states and input histories for each test condition and configuration, calls the simulation of the test harness model for each test condition and configuration, and stores the simulation output to the test data structure:
 - a. Setting initial states
 - i. For each test condition and configuration, the corresponding trim data for the FDM and actuation systems is gathered from the trim data set, and stored in an initial state structure part of
 - ii. The initial state structure is written to the test data structure
 - b. Definition of input histories
 - i. A default test input vector structure is loaded
 - ii. The test case-specific inputs, such as command doublets for a control mode under test, are defined
 - iii. The test input vector structure is updated with test case specific inputs and written to the test data structure
 - c. Execution of simulation model
 - i. The test harness model is run, taking initial states and input vectors from the current test data structure
 - ii. The simulation output is written to the test data structure
 - d. If desired, check of verification criteria
 - i. The local test case defines verification pass criteria such as specific signal ranges, flags for a subset of the test case outputs or internally logged signals
 - ii. The verification criteria are evaluated against the extracted signals from the simulation output structure, resulting in a pass or fail status of the test case
4. The teardown fresh fixture function loops over all test conditions and configurations stored in the test data structure:
 - a. Calculates and plots the simulation output summarized over the envelope

- b. If desired, plots the outputs for each test condition and configuration
- c. If desired, writes the test case metadata, envelope, and individual test points, and test pass/fail according to verification criteria to a LaTeX test report

The functions-based MIL verification setup allowed a generic MIL verification setup for all demonstration platforms. The file fixture setup is platform-specific, loading the specific trim data, and pointing to the platform-specific test harness model. The range of local test case functions was also adjusted for the demonstration platform under test, as the included control modes were platform-specific.

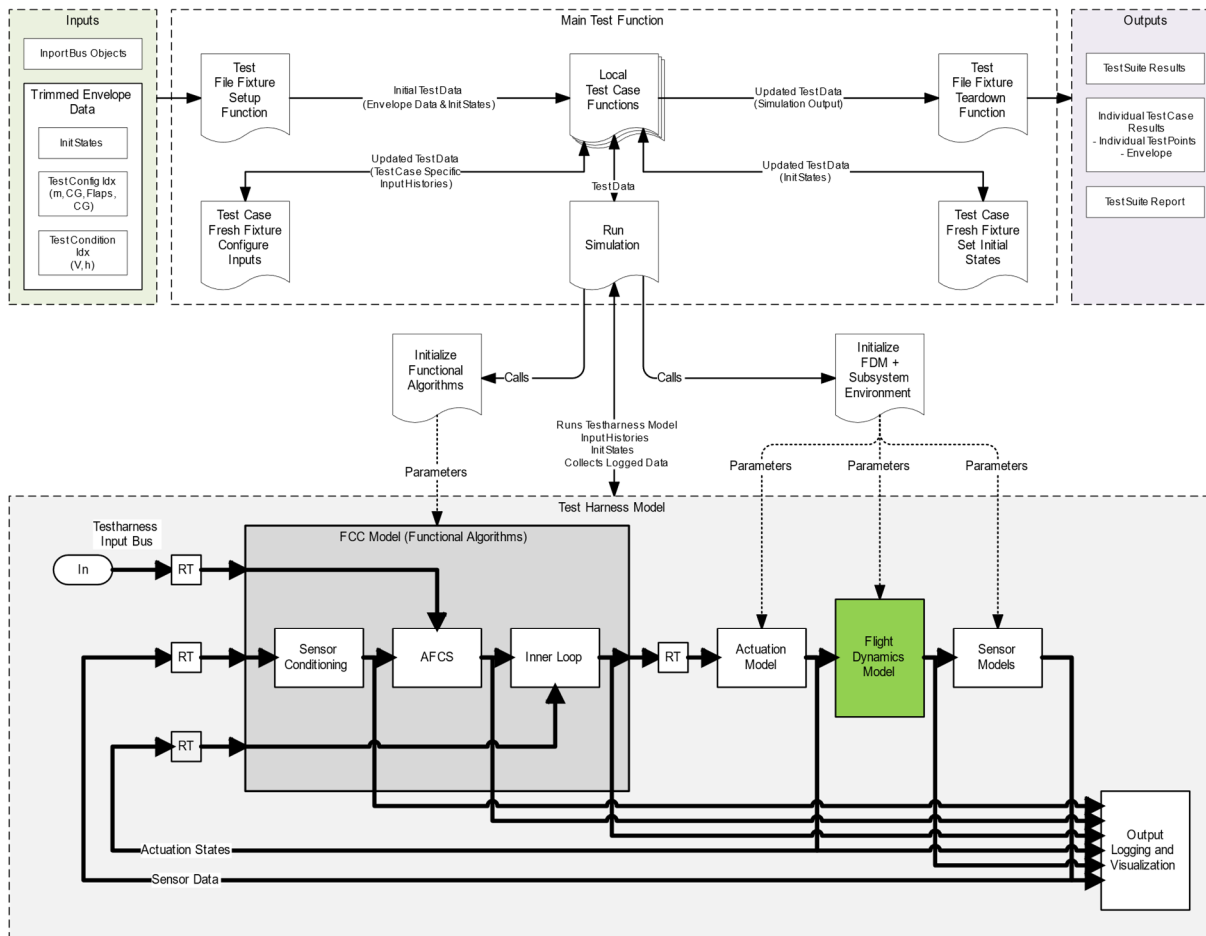


Figure 7.11: Simulink test harness model structure for AFCS model-in-the-loop verification.

7.2 Source Code Generation and Software Integration

This section provides a brief discussion on the automatic source code generation from the application software design model, and the integration of the automatically generated and the manually written software components. The purpose is to give an overview and understanding of the AFCS realization steps toward flight testing; however, the code generation and software integration activities have not been a primary focus of the author.

A detailed description of the hardware architecture, software components, and software integration activities is provided by Nürnberger and Hochstrasser [89], and the detailed workflow for generation of source code by Hochstrasser [98, p. 259].

The flight control algorithm design models are transformed into source code using MathWorks *Embedded Coder*. The intended workflow of Embedded Coder is to generate source code from a single, fully integrated design model, so-called *integral code generation*. An alternative approach, as presented in [98], is a modular code generation approach, where code is generated for each flight control algorithm module and integrated after the coding process. This allows for example for reuse of verified code, scalability due to shorter code generation times, and limited impact of design changes of separate modules on the code. The modular code generation approach, however, requires adaptation to the Embedded Coder workflow. At the time of implementation of the AFCS design model, the integral code generation approach was used, as discussed in [89].

DO-178C DAL A to C requires a code review [39]. With a model-based development process where the design model captures the software architecture and low-level design, the code review step may be accomplished by the use of a qualified code generation tool, or by a qualified automated checker tool. In SL/SF, the second approach is possible through the *Simulink Code Inspector*, which translates both the design model and automatically generated code to a special internal representation and checks these generated representations for equality. The safe modeling subset of the modeling framework module design rules discussed in Section 7.1.1 ensures that the design model is compliant with the Simulink Code Inspector.

The application software is embedded in a handwritten software application framework, developed conventionally according to DO-178C. The design objective of the application framework was a simple and efficient base software, with a simple scheduling strategy to achieve a robust timing behavior of the system, of great importance to flight control systems. Assumed worst-case execution times in the control loops are accounted for during the algorithm design and the stability and robustness analysis of the system. The design of the integrated software must comply with the timing constraints of the closed-loop system. Worst-case execution time verification is performed as part of the hardware-software integration testing [89].

A static scheduler using a periodic interval timer ensures a cyclic execution of the system. Each cycle is split into a foreground task and a background task. The foreground task requests input data from the FCC I/O processors (see overview of the FCC hardware architecture in Section 2.1.1), then extracts the data, executes the control algorithms, packs and transmits the data to the I/O processors, which in turn transmits the data on the external interfaces. The background task performs hardware monitoring as the I/O processors transmit the data, and then waits for the next periodic interval timer reset [89].

7.3 Hardware-Software Integration

The integrated software is compiled to a single binary using the compiler *CompCert* [89], before being deployed on the target hardware, described in Section 2.1.1, in the hardware-software

integration process. The hardware-software integration process further integrates the actuation system components such as ACE, the physical sensor interfaces, and the physical mode control and monitoring system presented in Section 5.4.

At TUM-FSD, a HIL laboratory setup for each demonstration platform described in Section 1.4 was developed by the respective project team. The purpose of the HIL laboratory setup is to enable testing of integrated hardware/software system behavior, communication, timing, bus loads, and wiring with the intended physical components and interfaces. The FDM and sensor models are executed in a test environment, receiving the actuator positions from the physical actuation system, and feeding the sensor outputs to the FCC on the real, physical interfaces via the laboratory test environment setup. The HIL setup enables real-time simulations or “virtual flight tests” for increased confidence before aircraft integration and flight testing, with verification of flight control algorithm performance and validation of the mode control and monitoring concept and design.

The architecture of the HIL laboratory setup for the Do-228 D-CODE is illustrated in Figure 7.12. The AFCS components integrated in the HIL setup are identical to the components installed in the aircraft: the FCC, the MCP and MCMD, the Safety Relay Box (SRB), the DCU reading the control surface position and control force sensors, the ACE as well as the Flight Control Actuation Units (FCAU) for all three axes. A load machine, exerting the simulated aerodynamic loads on the actuator, is integrated into one of the control axes. In the setup, the load machine is connected to the elevator actuator, with the purpose of testing the automatic pitch trim system of the Do-228 D-CODE AFCS, which triggers a movement of the Trimmable Horizontal Stabilizer (THS) based on the elevator control force. The cable harness connecting the AFCS components is not the cable harness for aircraft integration, but a specific HIL cable harness produced by the aircraft manufacturer according to the same regulatory standards as the one for aircraft integration.

To formally verify the HIL setup and the integrated actuation system performance, a set of acceptance tests was developed and performed. The acceptance tests included verification of the actuation system performance during tracking of sinusoidal, step, and pulse shaped input signals at various frequencies and amplitudes, as well as during the operation of the different AFCS control modes.

The HIL setup was further extensively used to validate the MCP/MCMD visual concept and operational logic. The MCP and MCMD layout and functionality were initially developed as executable desktop applications for the model-in-the-loop simulation environment. This greatly supported control law prototyping, early validation of system operation, and test pilot feedback before hardware manufacturing and HIL integration. In the HIL setup, the MCMD command logic and display application software (see description of the HMI architecture in Section 5.4.2), were fine-tuned before aircraft integration and flight tests.

The physical Do-228 D-CODE hardware laboratory setup is pictured in Figure 7.13.

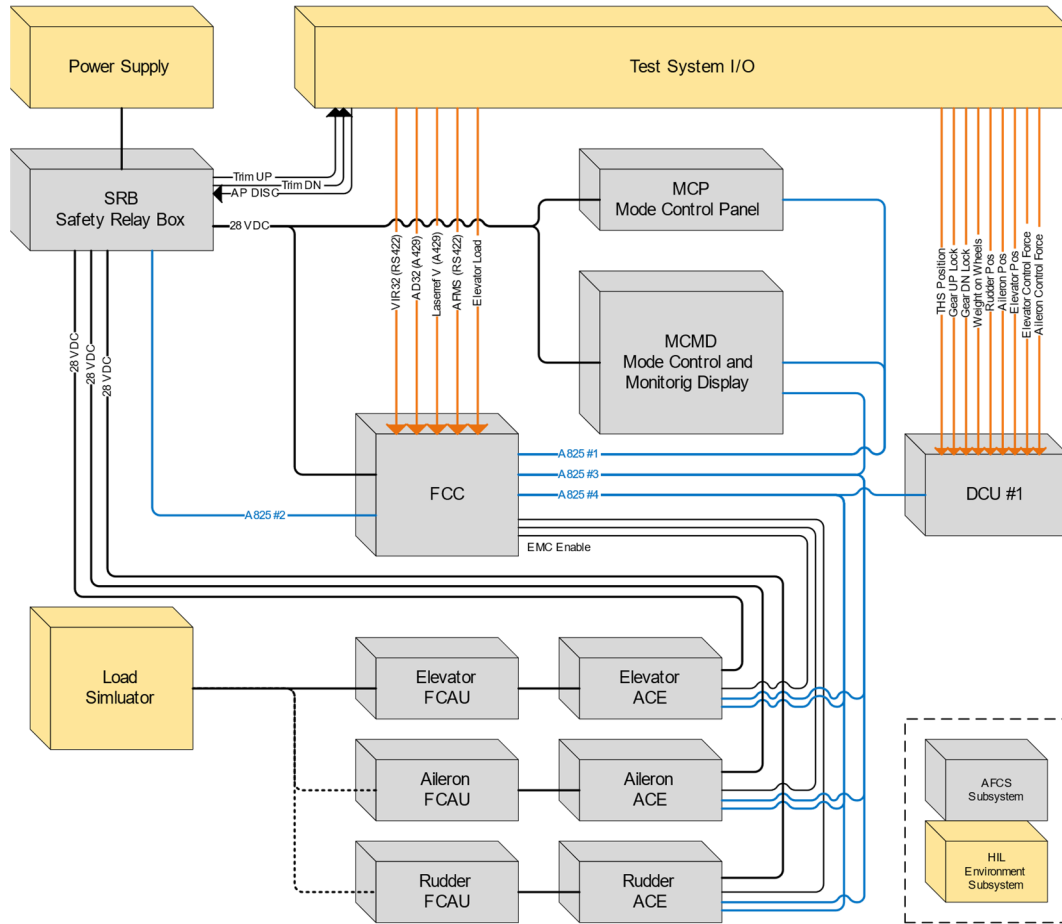


Figure 7.12: System architecture overview of the hardware-in-the-loop (HIL) laboratory setup for the Do-228 D-CODE experimental AFCS, with AFCS subsystems and HIL subsystems identified.

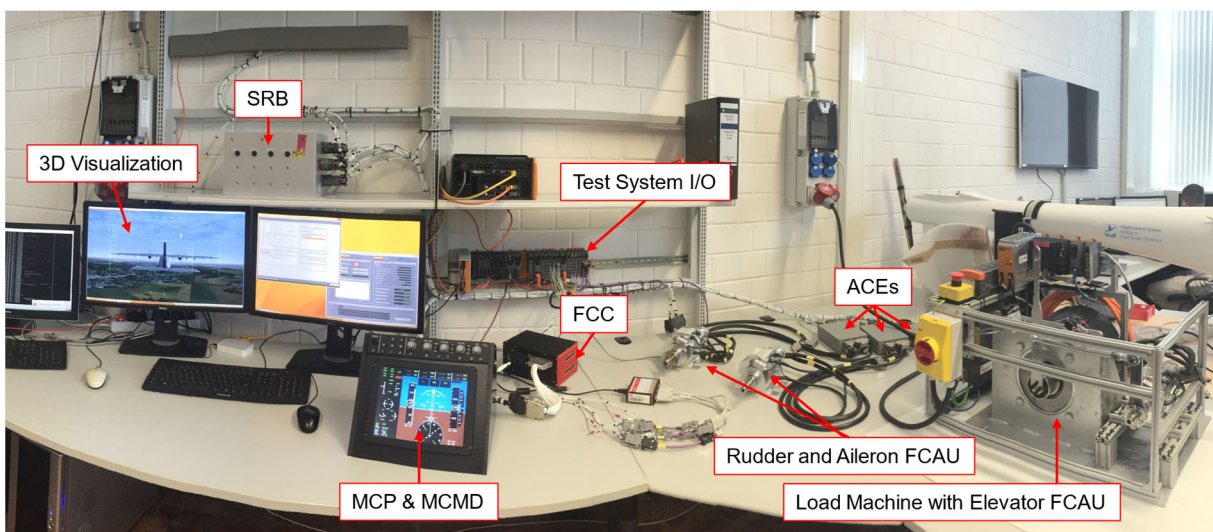


Figure 7.13: The hardware-in-the-loop (HIL) laboratory setup for the Do-228 D-CODE experimental AFCS at TUM-FSD (photo by the author).

7.4 Aircraft Integration

Following hardware-software integration and system verification and validation in the HIL, the flight control system components are integrated into the aircraft. After integration into the aircraft, functions and control system elements were tested in aircraft-in-the-loop simulations. The purpose of the aircraft-in-the-loop simulations is to provide final verification of the correct installation of the control system components and their correct functionality as installed in the aircraft before flight testing.

8 Verification and Validation

This chapter presents results from the verification and validation activities of the AFCS development, according to the verification and validation track of the System Development Process defined in Section 2.3.5:

- Control Law Assessment, Linear and Nonlinear
- Model-in-the-Loop (MIL) Verification and Validation
- Hardware-in-the-Loop (HIL) Verification and Validation
- Aircraft-in-the-Loop (AIL) Verification and Validation
- Flight Test Verification and Validation

The chapter focuses on results from the linear and nonlinear control law assessment, as well as flight test verification and validation.

The linear control law assessment utilizes the controller design and assessment framework described in Section 5.3.2 which is based on the linearized control system and flight dynamics models implemented as MATLAB generalized state space models. The nonlinear control law assessment utilizes the AFCS design model-in-the-loop setup described in Section 7.1.5, with the AFCS and inner loop controller in the loop with the full flight dynamics model as well as sensor and actuation system models. The nonlinear assessment utilizes the MATLAB Unit Test Framework for test case specification and test automation.

The purpose and scope of the system-level model- and software-in-the-loop verification and validation activities with respect to the integrated FCC control application software are summarized in Section 7.1.4 and are not further elaborated in this chapter. The purpose and scope of the hardware- and aircraft-in-the-loop verification and validation activities are summarized in Sections 7.3 and 7.4, respectively, and are likewise not further elaborated in this chapter.

The chapter is structured as follows. First, Section 8.1 discusses the reference configuration for the presented control law assessment and flight test results. The reference configuration is the DA42 OE-FSD introduced in Section 1.4.1. The linear assessment with time and frequency domain performance analysis is discussed in Section 8.2. Section 8.3 presents the nonlinear assessment including nominal path tracking verification as well as verification of the control objective conflict resolution strategy and energy protections, based on the AFCS MIL integration environment discussed in Section 7.1.5. The nonlinear assessment also covers the

verification of mode transition logic and mode control interfaces. Section 8.4 presents results from the modular FGCS deployment in January 2016 and initial flight tests from the perspective of the AFCS. The flight test history of the AFCS is discussed and additional flight test results covering flight path tracking and validation of energy protections concludes the chapter.

8.1 Reference Configuration

This section discusses the reference configuration for the presented results, the DA42 OE-FSD flying testbed used for the deployment and initial flight testing of the modular FGCS including the AFCS. An introductory description and basic aircraft parameters are given in Section 1.4.1 and Table 1.3.

8.1.1 Test Condition and Configuration Index

The test condition index is the airspeed-altitude envelope and flight conditions for which the different simulation tests and controller assessments are performed. The configuration index is the set of corresponding aircraft configurations, i.e., flaps and gear settings as well as weight and balance configurations.

The test condition and configuration index for the AFCS were chosen as the envelope and aircraft configuration grid for the inner loop command response dynamics that constituted the grid of input from the inner loop development and parameters design process to the AFCS parameter design. The assessment framework is scalable in the sense that performance assessment and stability analysis may be performed for any set of test conditions and configurations of interest.

For the linear assessment of the AFCS for the DA42 OE-FSD, four sets of closed inner loop command response dynamics for linearized level flight conditions over four different envelopes were provided. For the nonlinear assessment, trim data for the corresponding envelopes were the basis for the test case setup and initialization of the test harness model. The envelope grids for the four configurations of flaps and gear up and down are illustrated in Figure 8.1.

Figure 8.1 illustrates the test condition grids over altitude and indicated airspeed, with secondary axes indicating the corresponding static and dynamic pressures. For the longitudinal dynamics, the inner loop specific force response was scheduled according to static and dynamic pressure; for the lateral dynamics, the inner loop bank angle response was scheduled according to static pressure and indicated airspeed.

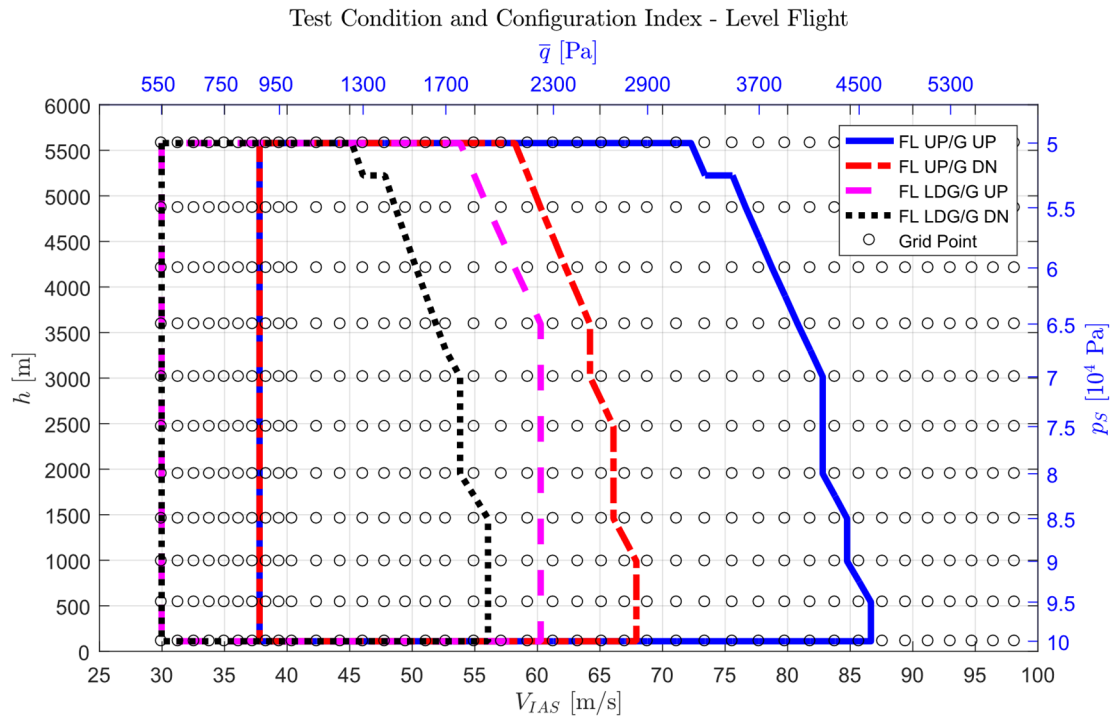


Figure 8.1: Test condition and configuration index for the assessment of the AFCS for the reference configuration DA42 OE-FSD. Test conditions are defined over indicated airspeed and altitude envelope, with trimmed flight conditions scheduled over static and dynamic pressure. Double x- and y-scales illustrates the corresponding values. Test configurations include flaps up and down, as well as gear up and down configurations.

8.1.2 Closed Inner Loop and Autothrust Dynamics

Figure 8.2 shows an analysis of the closed inner loop performance over the envelope, with bank angle and incremental body fixed normal specific force step responses, as well as the closed autothrust performance.

As described in Section 5.2.11, the development and validation of the AFCS concept lacked an underlying model of the engine dynamics. A simplified but realistic first-order model according to Eq. (3.88) was thus the basis for the design of a basic autothrust controller. The time constant for the throttle dynamics was set at $T_{\delta_T} = 0.5$ s. The autothrust proportional and integral gains were designed using the MATLAB *pidtune* function, by specifying a desired tuned open-loop crossover frequency of $\omega_{C,ATHR} = 0.5$ rad and a desired phase margin of 60 degrees. The desired design parameters resulted in a proportional gain $k_{p,\delta_T} = 0.157$ and integral gain $k_{I,\delta_T} = 0.803$, providing a smooth acceleration response according to the top left plot in Figure 8.2. The closed loop response time is approximately the inverse of the desired crossover frequency.

The linearized closed inner loop dynamics includes the second order models of the actuation system dynamics: for the normal specific force, the elevator dynamics; for the bank angle the aileron and rudder dynamics. The linearized models are implemented as generalized state

space models in MATLAB with analysis points at the actuator commands (and not the corresponding control surface commands), enabling loop openings for the analysis of required gain and phase margins. For the purpose of the linear analysis of the closed AFCS loops, the actuator commands are denoted ϕ_{η_C} for the elevator actuator command, ϕ_{ξ_C} for the aileron actuator command, and ϕ_{ζ_C} for the rudder actuator command.

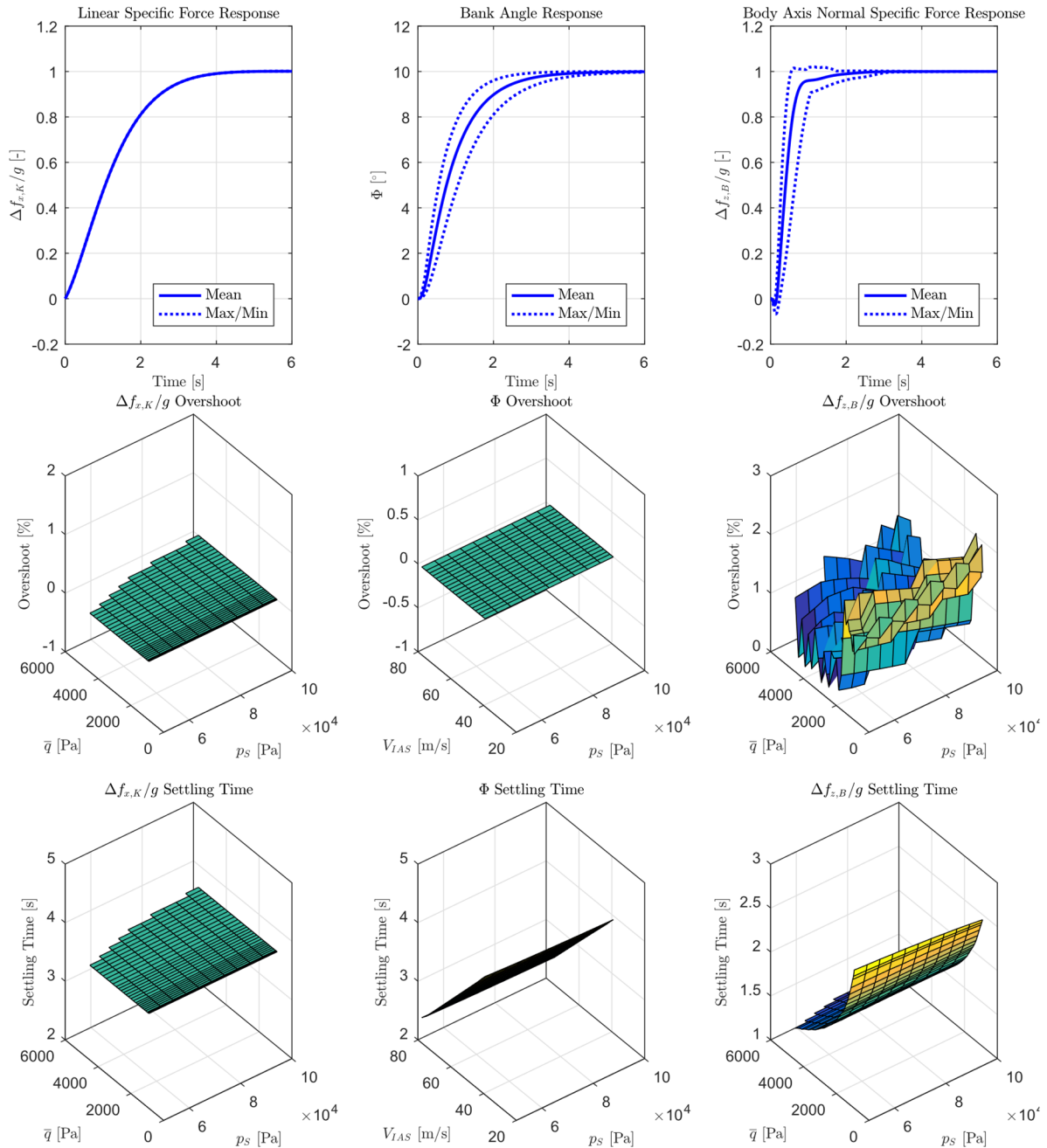


Figure 8.2: Inner loop and autothrust performance and over the envelope: step response curves, overshoots, and settling times for linear specific force (left column), bank angle (middle column), and normal specific force (right column). Gear up, flaps up configuration.

The inner loop tracks the bank angle command without overshoot over the envelope, with settling time significantly decreasing with increasing indicated airspeed (i.e., with increasing dynamic pressure and aileron effectiveness). The normal specific force response shows a typical non-minimal phase behavior with an initial small undershoot as well as a small overshoot over the envelope, with settling time decreasing with increasing dynamic pressure, analog to the bank angle response.

8.1.3 Controller Gains and Parameters

The parameter scheduling grid for the AFCS gains is based on the grid of the corresponding linear inner loop dynamics that is the input to the path loop parameter design. The reference model time constants for the flight path angle loop $T_{1,\gamma}$ and $T_{2,\gamma}$ as well as the error controller gains $k_{P,\gamma}$ and $k_{I,\gamma}$, are thus scheduled over the dynamic pressure \bar{q} and static pressure p_S . The reference model time constants for the track angle loop, $T_{1,\chi}$ and $T_{2,\chi}$ as well as the error controller gains $k_{P,\chi}$ and $k_{I,\chi}$, are scheduled over the indicated airspeed V_{IAS} and static pressure p_S .

Figure 8.3 illustrates the reference model time constants and error controller gains for the flight path angle loop for the DA42 OE-FSD in the gear up, flaps up configuration. As discussed in the path loop parameter design strategy in Section 5.3.3, the reference model time constants are designed such that the fixed faster pole is selected to smoothen the pseudo control, whereas the slower pole is designed to achieve a desired step response reference dynamics. The faster pole is selected to $T_{1,\gamma} = 0.5$ s. As seen in Figure 8.3, the slower pole $T_{2,\gamma}$ is faster with increasing dynamic pressure, as the maneuvering bandwidth of the inner loop controller increases.

As also discussed in Section 5.3.3, the error controller integral gain $k_{I,\gamma}$ is set to zero, as only the proportional gain $k_{P,\gamma}$ is required to provide static accuracy for path disturbances. The proportional gain is increasing with increasing dynamic pressure, as the maneuvering bandwidth of the inner loop allows for an increased disturbance rejection bandwidth.

Figure 8.4 illustrates the parameters for the Speed by Thrust loop for the DA42 OE-FSD in the gear up, flaps up configuration, i.e. the reference model time constants $T_{1,V,SBT}$ and $T_{2,V,SBT}$, as well as the error controller gains $k_{P,V,SBT}$ and $k_{I,V,SBT}$. The speed loop parameters are analog to the flight path angle loop scheduled over the dynamic pressure \bar{q} and static pressure p_S . The nominal acceleration limits in the speed reference model are set to $\dot{V}_{min/max,nom} = \pm 1$ m/s². The time constant for the complimentary filtering of the indicated airspeed was set quite low to $T_{CF} = 1$ s.

For the DA42 OE-FSD, where the simplified thrust dynamics that is the basis for the speed loop parameter design, the scheduling approach could be omitted and replaced by a fixed gain design due to the uniform parameter distribution over the envelope. However, in order to maintain a generic parameter design process and model implementation, the autothrust dynamics was integrated into the set of longitudinal plant models for AFCS loop design (see Section 5.3.2, Figure 5.25) for every envelope point defined by a linear inner loop model.

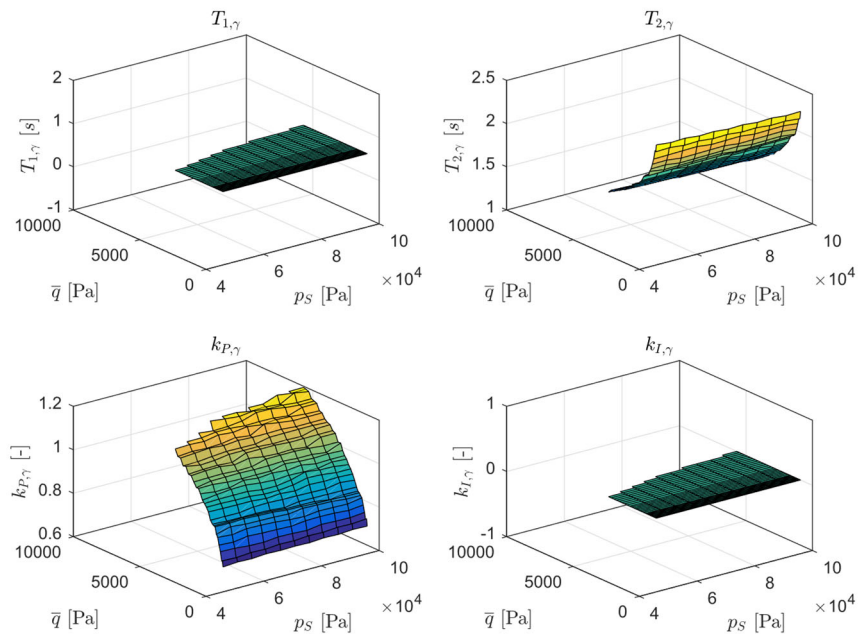


Figure 8.3: AFCS reference model time constants and error controller gains over the envelope for the flight path angle loop. Gear up, flaps up configuration.

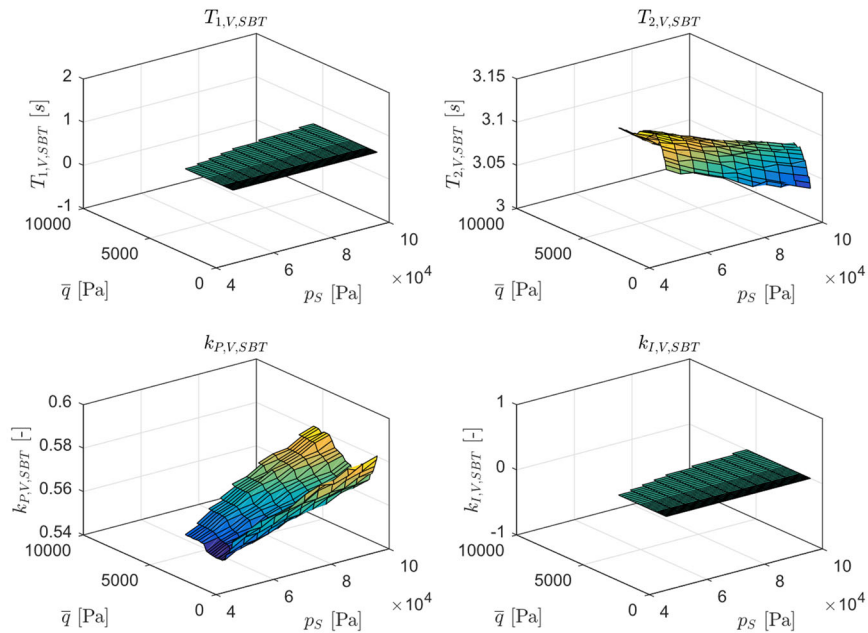


Figure 8.4: AFCS reference model time constants and error controller gains over the envelope for the speed by thrust loop. Gear up, flaps up configuration.

Figure 8.5 shows the parameters for the Speed by Pitch loop for the DA42 OE-FSD in the gear up, flaps up configuration, i.e. the reference model time constants $T_{1,V,SBP}$ and $T_{2,V,SBP}$, as well as the error controller gains $k_{P,V,SBP}$ and $k_{I,V,SBP}$. Where the speed by thrust gains are uniform over the envelope due to the assumed uniform thrust dynamics, the speed by pitch gains follow the pattern of the flight path angle gains, determined by the available normal specific force dynamics from the closed longitudinal inner loop controller.

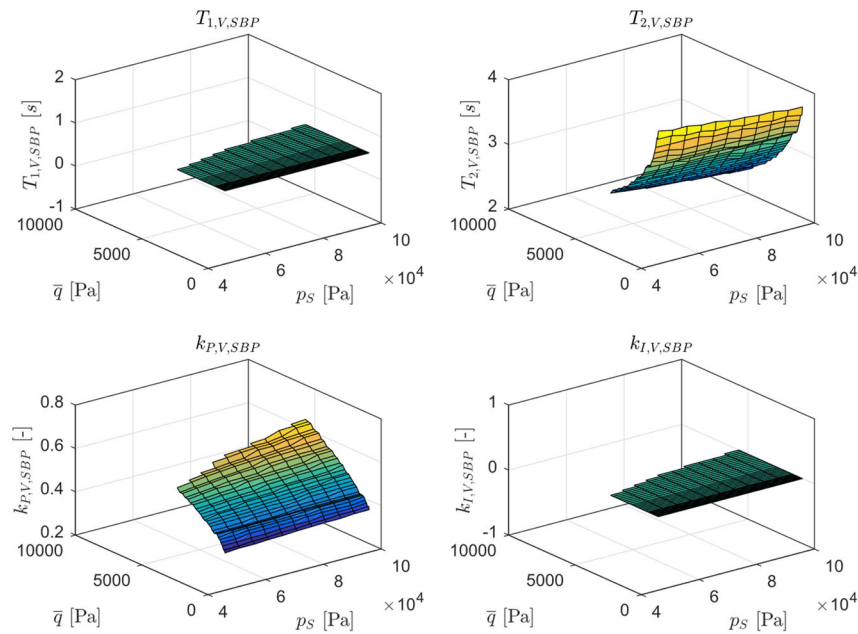


Figure 8.5: AFCS reference model time constants and error controller gains over the envelope for the speed by pitch loop. Gear up, flaps up configuration.

Figure 8.6 illustrates the parameters for the Track Angle loop for the DA42 OE-FSD in the gear up, flaps up configuration. Analog to the reference models for flight path angle and speed, the faster pole is selected to $T_{1,\chi} = 0.5$ s. The slower pole $T_{2,\chi}$ is faster ($T_{2,\chi}$ decreasing) with increasing indicated airspeed, as the bank angle control bandwidth of the inner loop controller increases. Analog to the error controllers for the flight path angle and speed, the error controller integral gain $k_{I,\chi}$ is set to zero. The proportional gain $k_{P,\chi}$ is increasing with increasing dynamic pressure, as the bank angle control bandwidth of the inner loop allows for an increased track disturbance rejection bandwidth.

Track angle rate limitations in the reference model are nominally set to provide standard rate turns at 3 degrees per second. For the verification and validation of the control objective conflict resolution, the track angle rate limits were set to 6 degrees per second.

The transformed body fixed commands to the inner loop are further limited at $\Phi_{C,min/max} = \pm 30$ degrees and $\Delta f_{B,C,min/max} = \pm 0.5$.

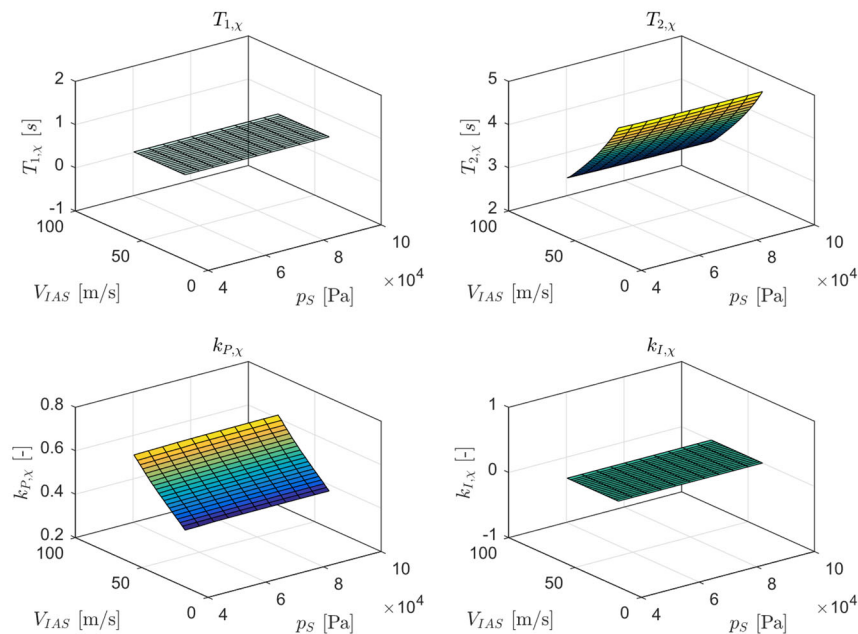


Figure 8.6: AFCS reference model time constants and error controller gains over the envelope for the track angle loop. Gear up, flaps up configuration.

8.1.4 Internal Command Processing and Sensor Delays

The physical control system elements introduce delays into the control loops in the form of processing time for the control algorithms on the FCC and ACEs, internal actuator, and command transport delays, as well as sensor processing and sensor data transport delays.

To capture these internal command and sensor processing effects, delays are introduced in each actuator command input and sensor output path. In the system-level integration environment described in Section 7.1.4, the actuation and sensor delays are set to 0.05 s. Identical delays are accounted for in the linear and nonlinear assessment of the AFCS. For the linear assessment, the delays are introduced in the generalized state-space models at the corresponding sensor output and actuation input analysis points. The nonlinear assessment is performed using the AFCS design model-in-the-loop environment described in Section 7.1.5 which utilizes the identical flight dynamics model, sensor, and actuation system models as the system-level integration environment in Section 7.1.4, and is thus initialized with the same actuation command and sensor delays. The effect of transmission delays due to remote-control operation of the AFCS via an external data link has not been analyzed.

8.2 Linear Assessment

The control law assessment uses a combination of test cases on the linear and nonlinear closed-loop model to verify time domain requirements, and a mathematical frequency domain analysis to verify stability and robustness requirements. The time domain performance requirements are defined as desired and adequate performance for target acquisition and disturbance rejection in terms of desired step response characteristics, such as target overshoot and settling time, see Figure 5.7 in Section 5.1.3. The time domain performance requirements also include desired and adequate performance for tracking accuracy in smooth air (static accuracy) as well as in turbulent air, with the turbulence profiles defined in Section 3.3.2. Table 5.12 in Section 5.1.3 provides an overview of the application-generic function-specific functional requirements. The frequency domain performance requirements include desired and adequate gain and phase margins at the bottleneck cuts (actuator commands).

The flight path control structure contains both linear and nonlinear elements, where the nonlinear elements, i.e. the command transformation, plant response estimation, and inversion, are analytically linearized and included in an extended linear plant model, see Section 5.3.2. The plant model includes the linearized flight dynamics model and the linear inner loop controller. They yield a linear plant model for the linear parts of the flight path controller, i.e., the reference models and the error controllers. Thus, for the different configurations of the controller, it is possible to create a linear form for a given operating point and utilize standard assessment techniques for linear systems.

The linear assessment is based on the linearized models of the closed inner loop and plant dynamics, i.e. models with the flight path and thrust controller outputs, $f_{z,B,C}/g$ and $\delta_{T,C}$, respectively, as inputs, and the aircraft states as outputs. The linear flight path control structure, i.e. the reference models and error controllers, together with linearized models of the plant response estimation, inversion, and command transformation for different flight conditions, are connected to the linear versions of the closed inner loop longitudinal (see Figure 5.25, Section 5.3.2) and lateral control systems (see Figure 5.26, Section 5.3.2), respectively.

The linear systems are implemented as generalized state-space models in MATLAB, with parametrized gains and switches, in order to configure the models for different control modes, as well as analysis points to simplify the analysis of loop stability margins. The model contains internal loop openings at all sensors, actuators, and intermediate commands for stability and robustness analysis. For each loop, step responses and settling times are analyzed with respect to the time domain requirements, as well as gain and phase margins and Nichols charts for the frequency domain requirements.

The time domain analysis encompasses vertical and lateral flight path and speed responses to corresponding command inputs, and vertical and lateral flight path and speed responses to path disturbance inputs, with automatic check of overshoot and settling time requirements. The frequency domain analysis encompasses system stability analysis of actuator and sensor loop cut stability margins (gain and phase margins, and Nichols plots), with automated requirements verification over the envelope.

8.2.1 Path Tracking Performance

Figure 8.7 shows linear step responses for the speed by thrust (first column), track angle (middle column), and flight path angle (right column) loops, respectively, as produced by the design and assessment routines described in Section 5.3.2. The first row shows the maximum, minimum, and mean speed, vertical, and lateral flight path responses over the envelope, together with the response boundaries for desired and adequate performance. Figure 8.8 shows the speed and flight path response overshoots and settling times plotted over the envelope for the assessed configuration, here the clean configuration with flaps and gear up, with corresponding assessment grid of desired and/or adequate performance achieved, as well as statistics of requirement violations. In Figure 8.8, it is concluded that the system satisfies the desired command tracking performance requirements for the speed and flight path angle loops over the entire envelope, as well as for the track angle loop except for the low-speed range of the envelope, where the settling times are slightly larger than desired, but well within the stated adequate performance.

In Figure 8.7, the second row shows the control errors, i.e., e_V , e_χ and e_γ . As discussed in Section 5.2.2, the path control architecture decouples the reference dynamics and error dynamics through the pseudo control hedging “hiding” the inner loop dynamics not accounted for in the reference model dynamics. Without the pseudo-control hedging, the inner loop dynamics would cause the flight path response to lag the reference model, resulting in a control error. Here, in the absence of external disturbances, the path loop output perfectly follows the reference dynamics as defined by the reference model parameters, and the resulting control error is equal to zero, as can be seen in Figure 8.7.

The third row shows the pseudo-controls from the reference models, $v_{RM,\dot{V}}$, $v_{RM,\dot{\chi}}$ and $v_{RM,\dot{\gamma}}$. The second-order reference models produce smooth pseudo controls entering the command transformation to the inner loop interface. The fourth row shows the pseudo controls added by the respective error controller, $v_{PI,\dot{V}}$, $v_{PI,\dot{\chi}}$ and $v_{PI,\dot{\gamma}}$. As discussed above, the control error is equal to zero in the absence of external disturbances, and the resulting error controller pseudo-controls are likewise equal to zero.

The fifth row shows the commands to the autothrust loop, $\Delta f_{x,K,C}/g$, and to the inner loop, Φ_C and $\Delta f_{z,B,C}/g$. The sixth and seventh rows show resulting the incremental throttle command from the autothrust loop and the aileron and rudder actuator commands, ϕ_{ξ_C} and ϕ_{ζ_C} , as well as the elevator actuator command ϕ_{η_C} . It can be noted that the control surface actuator commands for the elevator and aileron are of opposite sign as the corresponding control surface deflections. A positive flight path angle command results in a negative normal specific force command (i.e., a positive load factor), achieved by a negative elevator deflection (i.e., upwards), here a positive elevator actuator command. Analog to the positive track angle command, the positive bank angle is achieved by a negative aileron deflection (right hand aileron upwards), here a positive aileron actuator command.

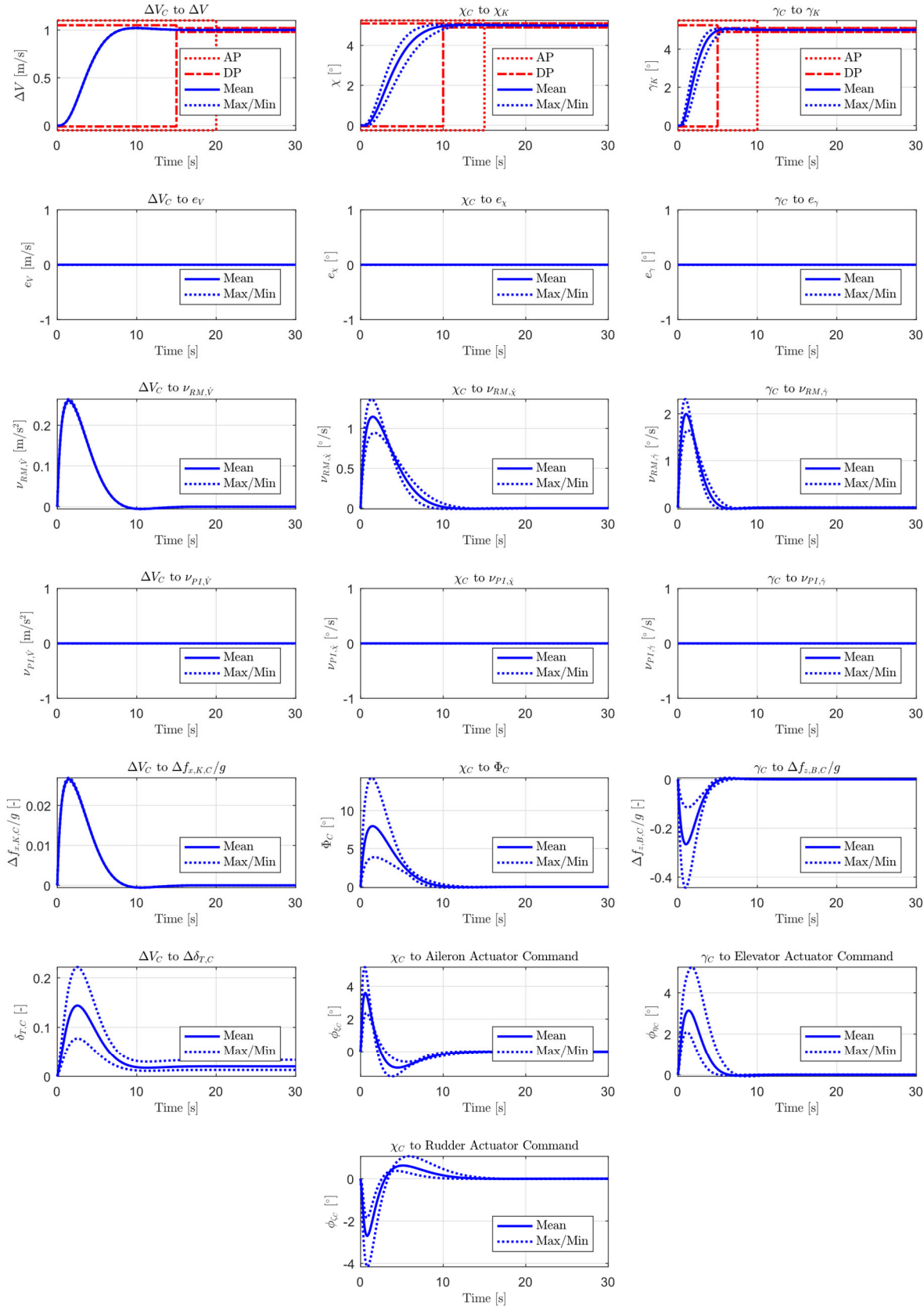


Figure 8.7: Linear step responses over the envelope for speed by thrust (left column), track angle (middle column), and flight path angle (right column). Gear up, flaps up configuration.

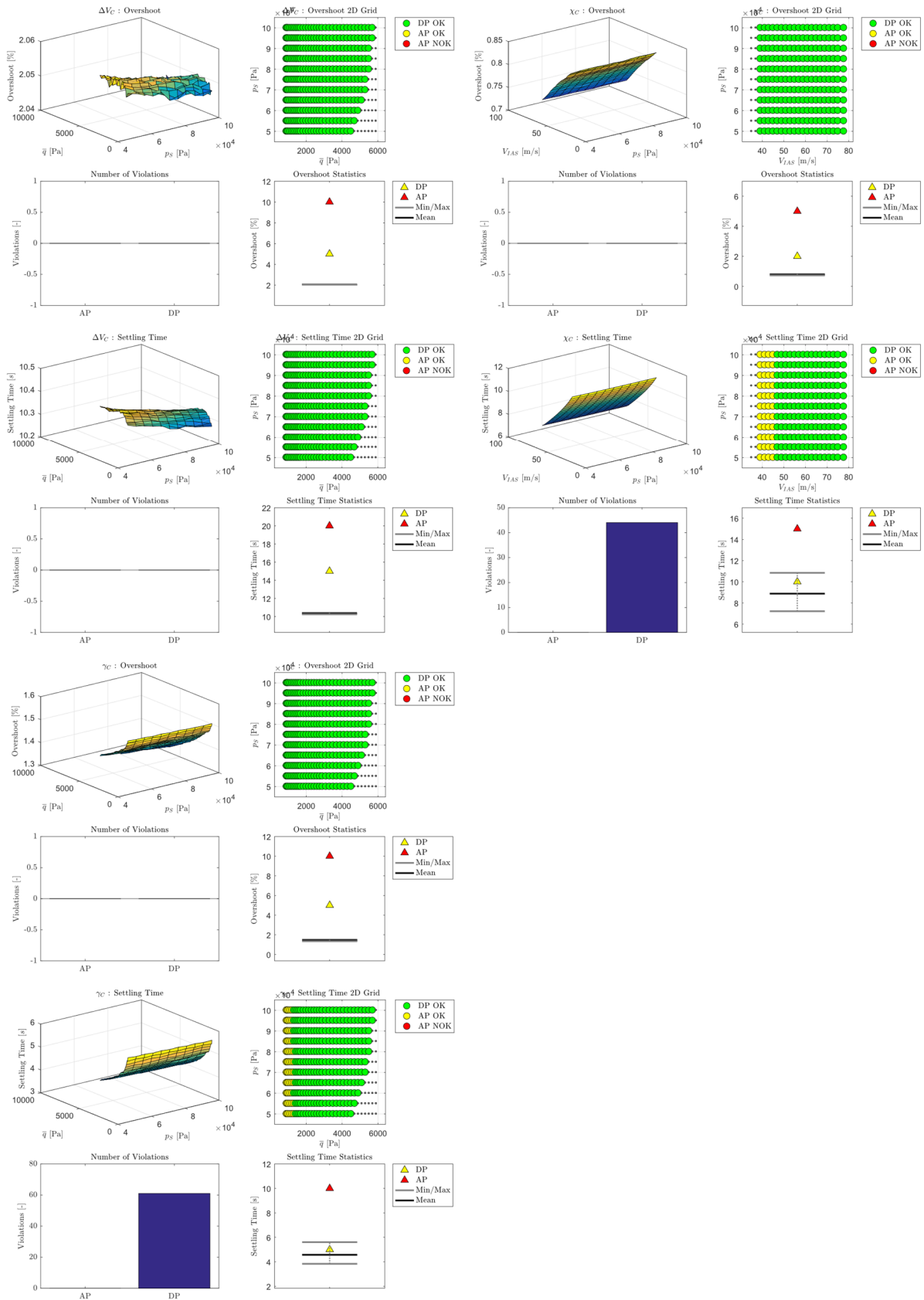


Figure 8.8: Step response performance requirements for the speed by thrust, track angle, and flight path angle loops. Overshoot and settling time analysis and performance requirement evaluation over the envelope. Gear up, flaps up configuration.

8.2.2 Disturbance Rejection

Step disturbances are injected in the linear path loops at the output signal according to Figure 5.29 in Section 5.3.3 (disturbance input denoted Δ_2), resulting in a step in the flight path angle, track angle, or aerodynamic speed driven to zero by the error controller. Figure 8.9 shows linear step responses for a speed (first column), track angle (middle column), and flight path angle (right column) disturbance, respectively. The order of the step response plots is identical to Figure 8.7. The first row shows the maximum, minimum, and mean speed, lateral and vertical flight path responses over the envelope. The second row shows the control errors; in contrast to the command response dynamics in Figure 8.7, where the pseudo-control hedging hides the inner loop dynamics from the error dynamics, resulting in a control error equal to zero, the external disturbance must be handled by the error controller.

The third row again shows the pseudo-controls from the reference models, $v_{RM,\dot{v}}$, $v_{RM,\dot{\chi}}$ and $v_{RM,\dot{\gamma}}$. The feedback structure to the reference models due to the pseudo-control hedging affects the reference flight path and the pseudo-controls from the reference models. Their contributions to the total pseudo-controls are however minor, as the pseudo-controls from the error controllers are about an order of magnitude greater, as can be seen in the fourth row.

The fifth row shows the commands to the autothrust loop, $\Delta f_{x,K,C}/g$, and to the inner loop, Φ_C and $\Delta f_{z,B,C}/g$. The sixth and seventh rows show resulting the incremental throttle command from the autothrust loop and the aileron and rudder actuator commands, ϕ_{ξ_C} and ϕ_{ζ_C} , as well as the elevator actuator command ϕ_{η_C} .

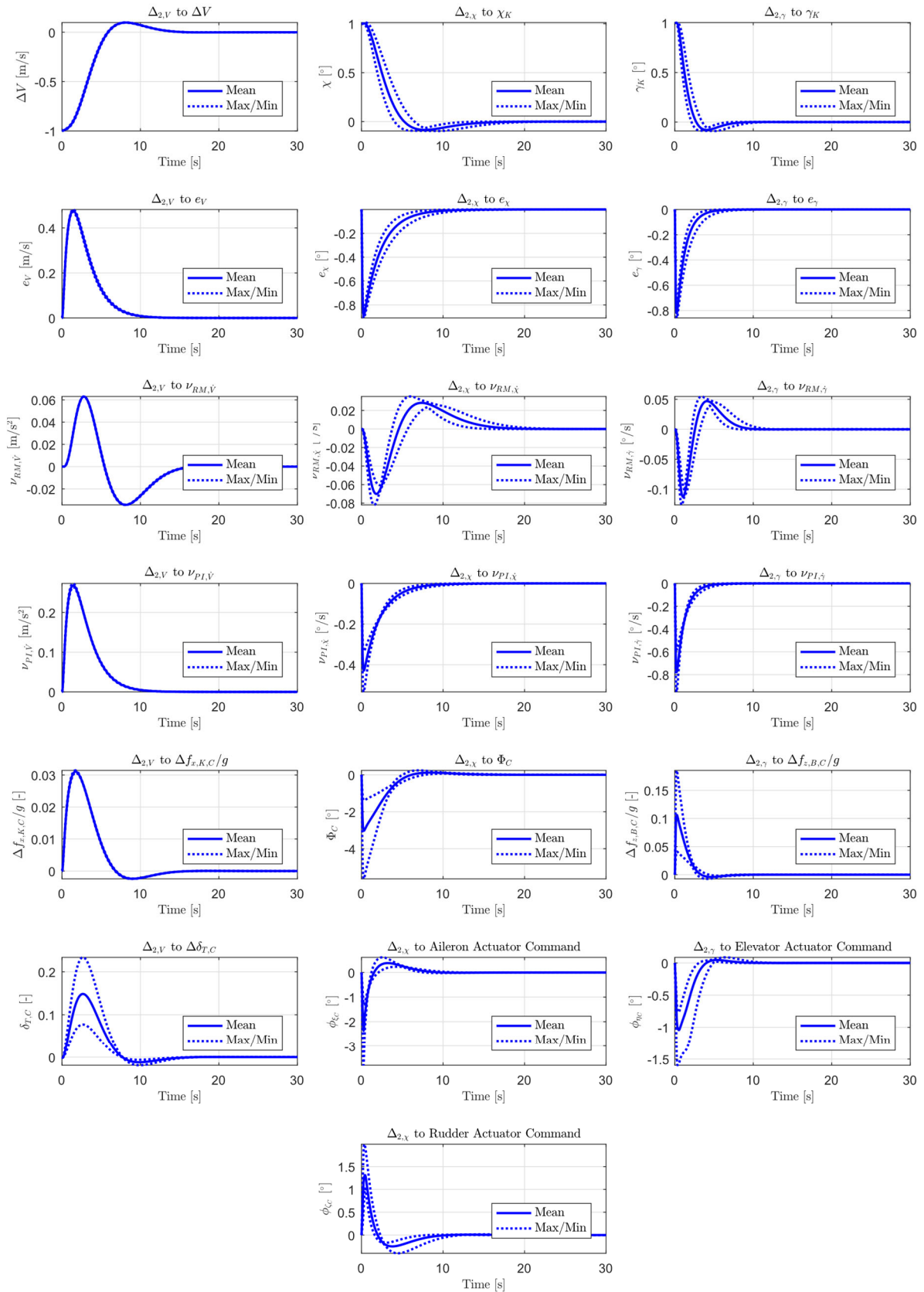


Figure 8.9: Linear step responses over the envelope for a speed/path disturbance input. Speed by thrust (left column), track angle (middle column) and flight path angle (right column) responses. Gear up, flaps up configuration.

8.2.3 Stability and Robustness

The stability and robustness of the control loops are determined according to the required gain and phase margins as per AS94900A [103], see discussion on system requirements in Section 5.1.3 and requirement example in Table 5.10 regarding desired and adequate general stability margins. The linear control system model implementation as generalized state space models in MATLAB with analysis points allowing for the calculation of gain and phase margins at all actuator commands as well as sensor outputs. For the assessment of the closed-loop system stability, the closed loops are cut one at a time at a location subject to analysis. The remaining loops are kept closed during the analysis. Common practice, and as required by AS94900A [103], is to cut each loop at the “bottleneck”, i.e. at the actuator inputs, which has also been the focus for the assessment of the AFCS. Loop cuts at the sensor outputs may provide additional information on the acceptable gain and time delay reserves for a particular feedback signal, as well as the impact of sensor degradation on the system stability, providing input to the specification of required sensor characteristics.

Figure 8.10 shows the flight path angle and speed by thrust loop stability and robustness performance requirements assessment over the envelope, including the nominal actuation command and sensor delays described in Section 8.1.3. Gain and phase margins are calculated for the flight path angle loop cut at the elevator actuator command ϕ_{η_C} and for the speed by thrust loop cut at throttle command $\delta_{T,C}$. Results are shown for the gear up, flaps up configuration. Figure 8.11 shows the corresponding Nichols charts for the elevator actuator command and throttle command cuts, together with Nichols diamonds representing the desired and adequate stability performance according to the requirement example in Table 5.10, i.e. an outer diamond illustrated a desired gain margin of 8 dB and a desired phase margin of 50 degrees, as well as an inner diamond illustrating an adequate gain margin of 6 dB and adequate phase margin of 45 degrees. Figure 8.10 shows large gain and phase margins for the flight path angle loop, whereas the speed loop phase margin is slightly above the desired level. The Nichols diamonds in Figure 8.11 are not violated. Thus, it is concluded that the longitudinal system with nominal sensor and command transport delays satisfies the desired stability requirements. The large stability margins in the flight path angle loop indicate a possibility for increasing the gain of the error controller. With satisfactory controller performance in the linear and nonlinear simulation, the large gain and phase margins were kept as an uncertainty buffer for controller validation in flight tests. The satisfactory initial flight testing led to no major retuning of the gains.

Figure 8.12 shows the track angle loop stability and robustness performance requirements assessment over the envelope, including the nominal command and sensor transport delays. Gain and phase margins are calculated for the track angle loop cut at the aileron actuator command ϕ_{ξ_C} and the rudder actuator command ϕ_{ζ_C} , showing sufficient gain and phase margins over the envelope. Results are shown for the gear up, flaps up configuration. Figure 8.13 shows the corresponding Nichols charts for the aileron and rudder actuator command cuts, with the Nichols diamond not violated. Thus, it is concluded that the lateral system with nominal sensor and command transport delays satisfies the desired stability requirements.

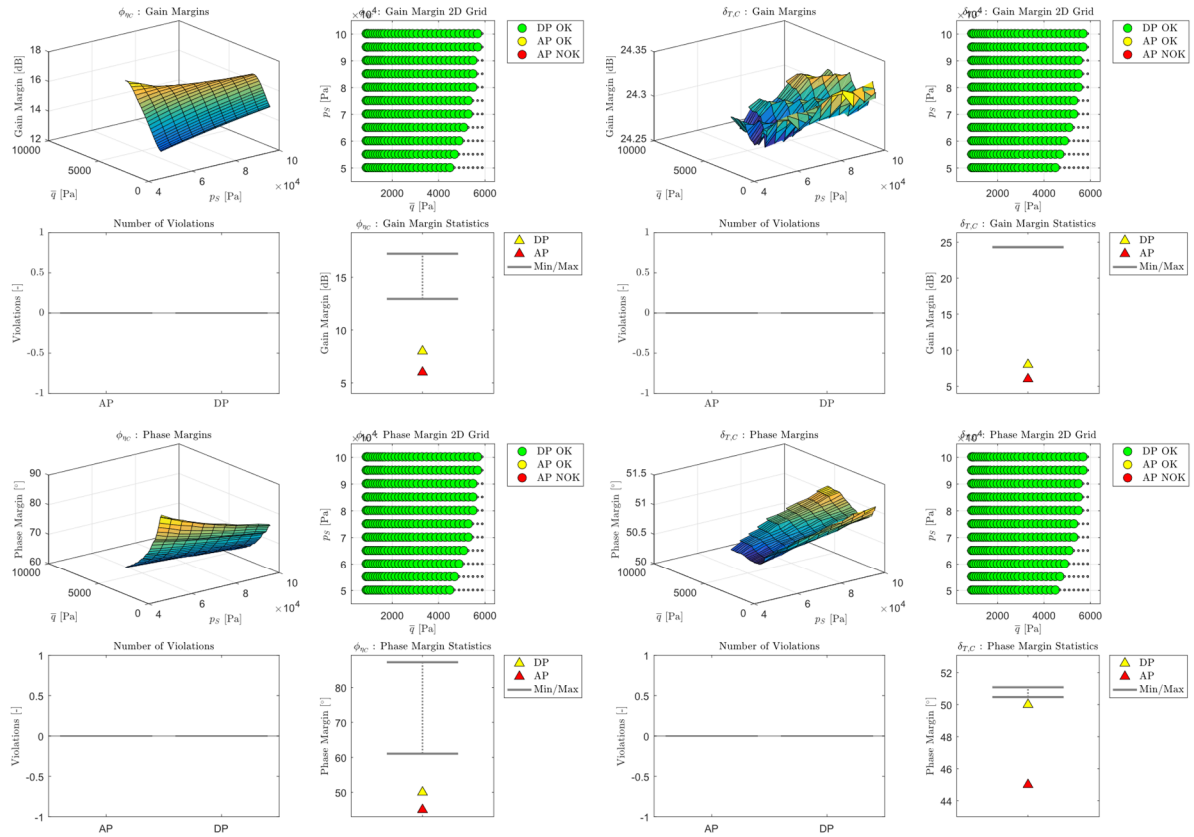


Figure 8.10: Flight path angle and speed by thrust loop stability and robustness performance requirements assessment over the envelope. Gain and phase margins for the flight path angle loop cut at the elevator actuator command ϕ_{η_C} (left) and for the speed by thrust loop cut at throttle command $\delta_{T,C}$. Gear up, flaps up configuration.

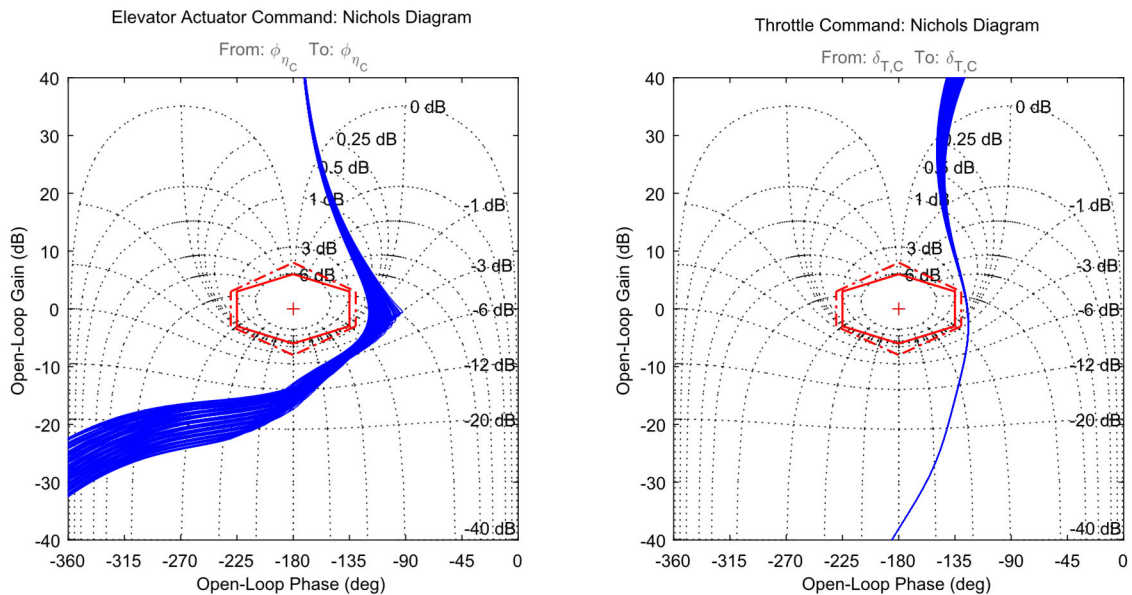


Figure 8.11: Nichols diagrams for the flight path angle loop cut at the elevator actuator command ϕ_{η_C} (left) and for the speed by thrust loop cut at throttle command $\delta_{T,C}$. Gear up, flaps up configuration.

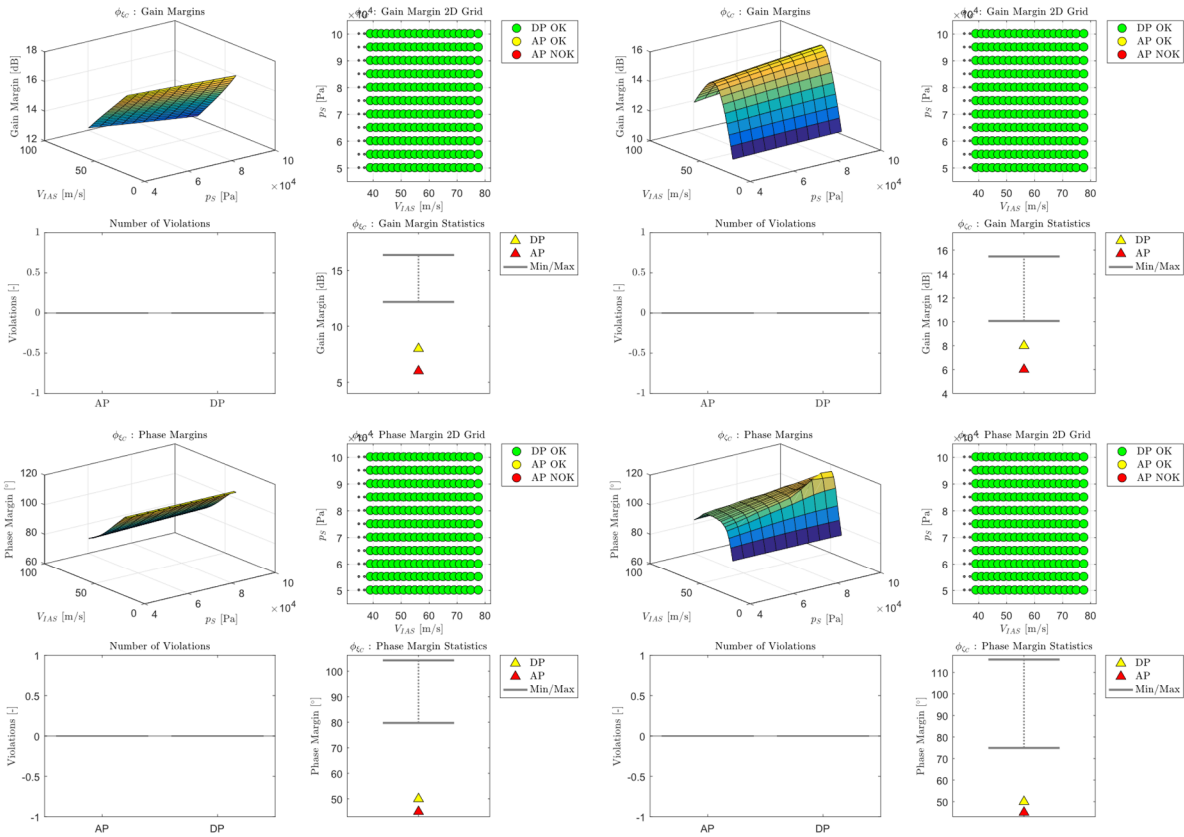


Figure 8.12: Track angle loop stability and robustness performance requirements assessment over the envelope. Gain and phase margins for the for the track angle loop cut at the aileron actuator command ϕ_{ξ_C} (left) and rudder actuator command ϕ_{ζ_C} (right). Gear up, flaps up configuration.

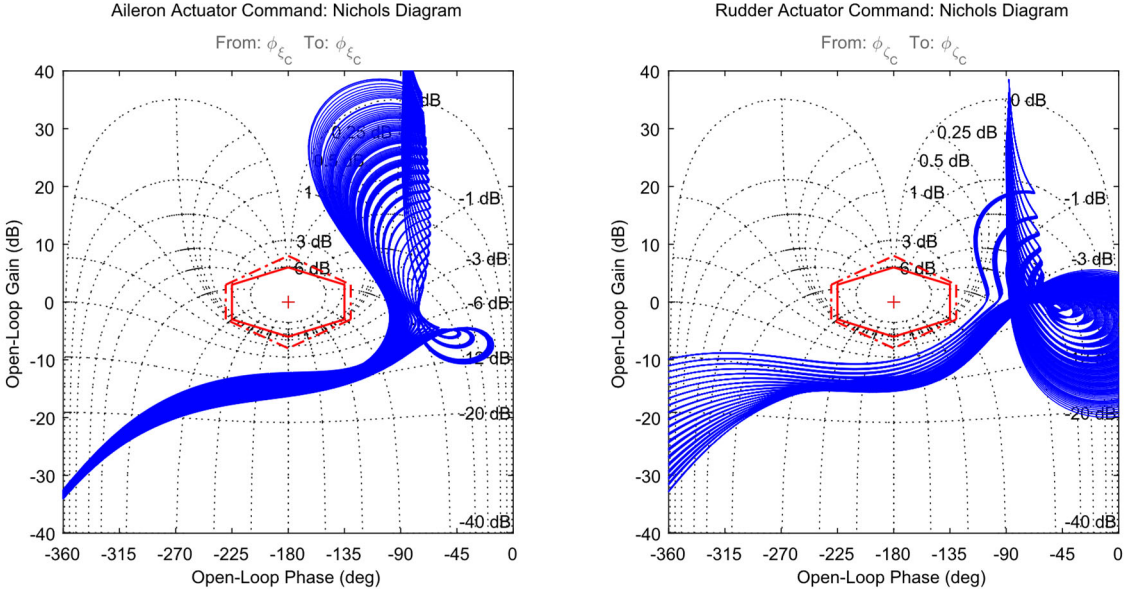


Figure 8.13: Nichols diagrams for the track angle loop cut at the aileron actuator command ϕ_{ξ_C} (left) and rudder actuator command ϕ_{ζ_C} (right). Gear up, flaps up configuration.

8.3 Nonlinear Assessment

The nonlinear assessment includes verification of command response dynamics, accuracy, and disturbance rejections in smooth and turbulent air. A dedicated test harness is used for automated verification over the envelope, the AFCS Design model-in-the-loop environment described in Section 7.1.5, including the AFCS, the inner loop controller, flight dynamics model, and sensor and actuation models including nominal transfer delays.

The nonlinear simulation allows for the evaluation of system performance during combined vertical and lateral plane maneuvers, compared to the linear analysis where the longitudinal and lateral dynamics are modeled decoupled.

For analysis of nominal path tracking, a series of command doublets representing standard autopilot maneuvers well within the maneuvering envelope of the aircraft were defined, illustrated as *Maneuver 1* in Figure 8.14.

For analysis of the control objective conflict resolution strategy and the energy and force-based prioritizations, a second series of higher amplitude commands were defined, representing saturated energy-rate and path curvature control, triggering control objective conflicts, illustrated as *Maneuver 2* in Figure 8.14.

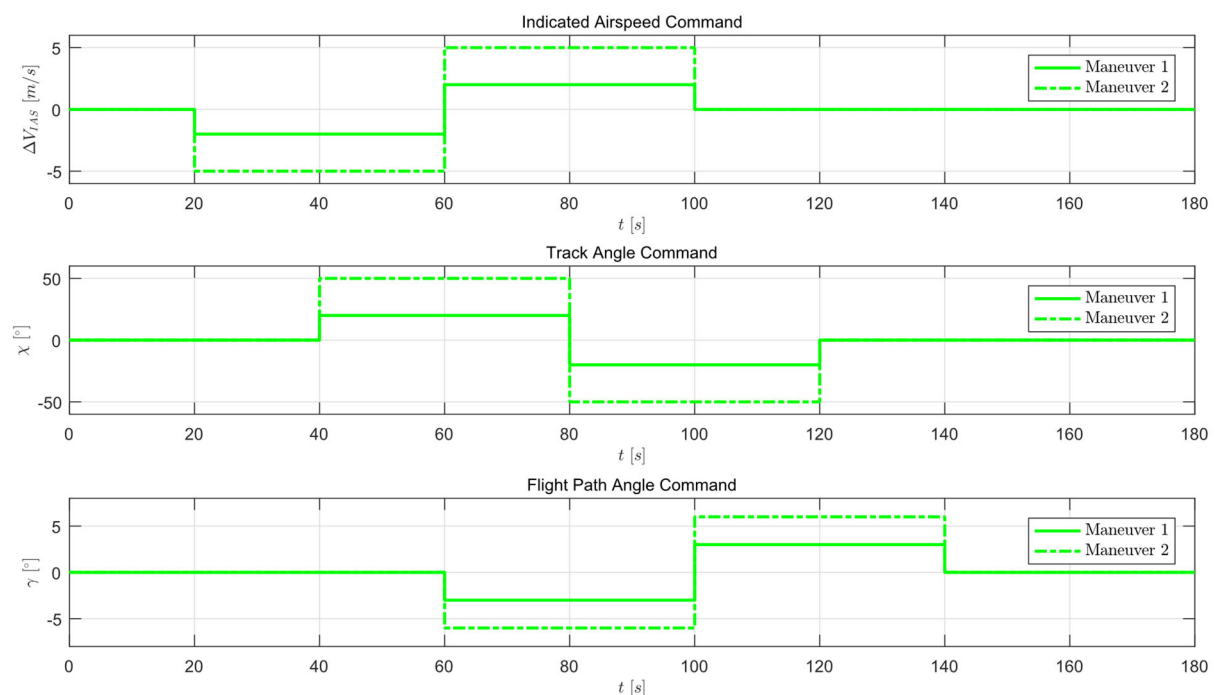


Figure 8.14: Command histories for maneuvers with simultaneous speed, vertical path, and lateral path tracking for nonlinear simulation testing. Maneuver 1 consists of nominal commands to test tracking performance in smooth and turbulent air. Maneuver 2 combines greater command inputs to trigger control objective conflicts and test resolution functionalities.

Maneuver 1 includes a plus/minus 2 meters per second incremental speed command doublet at 40-second intervals, a plus/minus 20-degree track angle command doublet, also at 40-

second intervals, offset the speed command, and a plus/minus 3-degree flight path angle command doublet. Maneuver 2 includes a plus/minus 5 meters per second incremental speed command doublet, a plus/minus 50-degree track angle command doublet, and a plus/minus 6-degree flight path angle command doublet, at the same intervals as Maneuver 1. The maneuvers are evaluated both in smooth and moderately turbulent air, at individual test points in the envelope, and automated over the entire test condition and configuration index.

8.3.1 Path Tracking in Smooth Air

Figure 8.15 shows system responses for simultaneous speed and path tracking in smooth air according to Maneuver 1 defined in Figure 8.14. Results are shown for an example test point representing an indicated airspeed of about 61 meters per second (ca 119 knots) and an altitude of about 1500 meters, in a gear up and flaps up configuration. The first row shows the command tracking for indicated airspeed (left), track angle (middle), and flight path angle (right). The second row shows the desired acceleration and flight path curvature tracking. The third row shows the linear specific force tracked by the autothrust controller and the bank angle and normal specific force commands tracked by the inner loop controller. The corresponding throttle, aileron, rudder, and elevator commands are seen in the fourth row, and the throttle, aileron, rudder, and elevator rates in the fifth row.

The tracking of speed and vertical and lateral flight path commands show smooth command acquiring and capture, and only minor disturbances resulting from maneuvering in the other control planes. The speed tracking shows small overshoots of about 2 percent of the commanded value, corresponding to the linear analysis. Track angle commands are captured with small overshoots of less than 0.5 degrees or about 1 percent, also corresponding well to the linear analysis. Flight path angle commands are captured with overshoots of about 0.1 degrees for a 3-degree step or about 3 percent, which is a somewhat larger overshoot compared to the linear analysis.

For the simultaneous speed and flight path angle command change at $t = 100$ s, where the speed command is reduced by 2 meters per second and the flight path angle command is increased by 6 degrees, the effect of the throttle feedforward command from the flight path angle loop to the autothrust controller according to Eq. (5.155) can be observed. The speed command reduction results in a reduced throttle setting, whereas the flight path angle command results in a feedforward command that dominates the throttle setting, increasing from about $\delta_{T,C} = 0.3$ to about $\delta_{T,C} = 0.8$ during the flight path change, where at the same time, the speed is smoothly reduced to the new speed target.

For this test, the energy-based acceleration versus vertical flight path prioritization is set to prioritize the flight path, which can be seen by the acceleration limits adjusting to the desired flight path angle. The desired flight path changes are consciously selected here not to trigger a control objective conflict, as they do with Maneuver 2 (more on the control objective conflict resolution results in Section 8.3.3), but the calculation of the limits is constantly running in the background unless switched off. The lateral flight path curvature is limited to standard rate turns at 3 degrees per second.

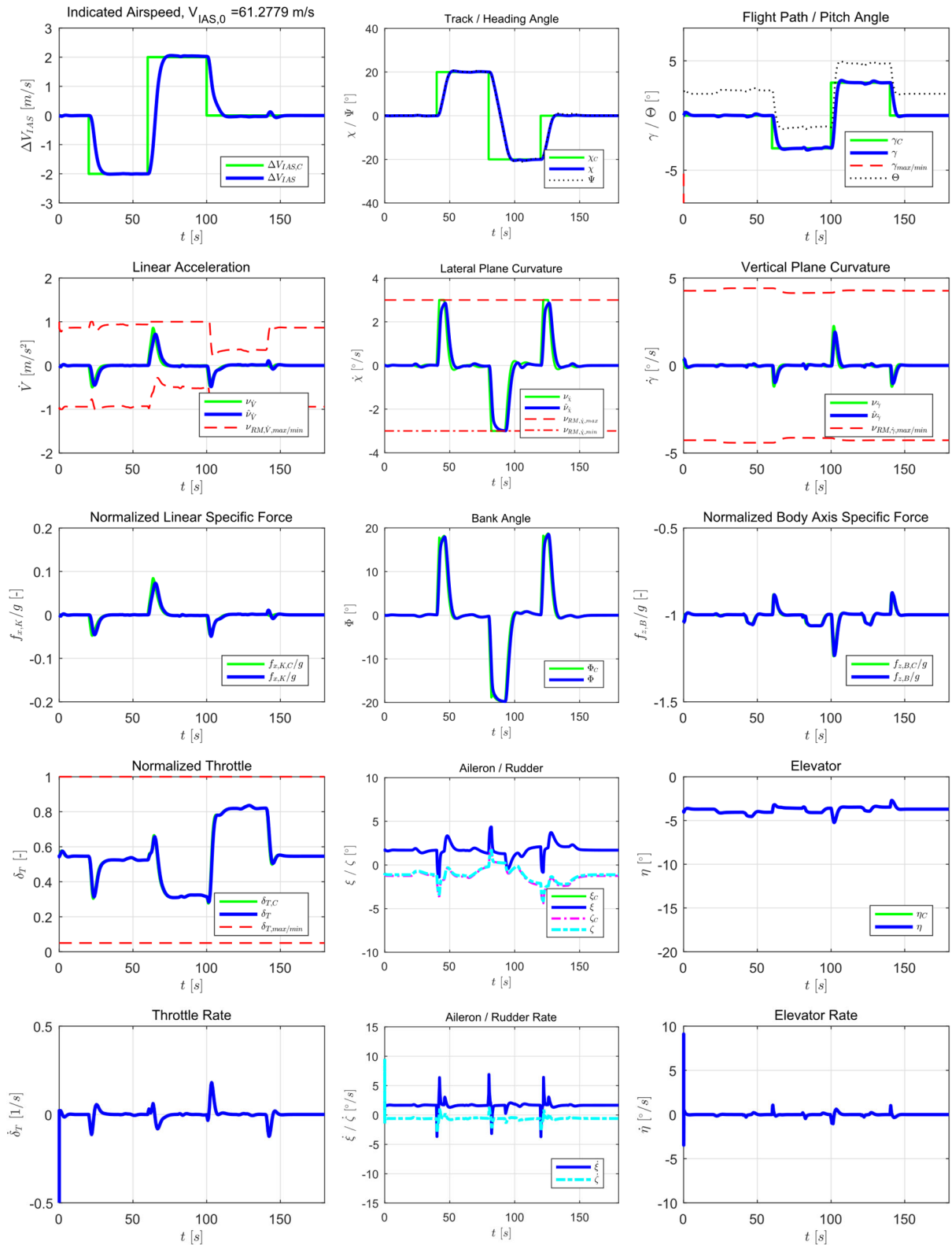


Figure 8.15: Speed and flight path command tracking during combined nominal vertical and lateral flight path and speed commands, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.

8.3.2 Path Tracking in Turbulent Air

Figure 8.16 shows system responses for simultaneous speed and path tracking according to Maneuver 1 defined in Figure 8.14 in moderately turbulent air and a 20 knots crosswind relative to the initial track angle. Results are shown for a test point representing an indicated airspeed of about 61 meters per second (ca 119 knots) and an altitude of about 1500 meters, in a gear up and flaps up configuration. A 5-second excerpt of the system responses in Figure 8.16 is shown in Figure 8.17.

Similar to the path tracking results in smooth air in Figure 8.15, the first row shows the command tracking for indicated airspeed (left), track angle (middle), and flight path angle (right). The second row shows the desired acceleration and flight path curvature tracking. The third row shows the linear specific force tracked by the autothrust controller and the bank angle and normal specific force commands tracked by the inner loop controller. The corresponding throttle, aileron, rudder, and elevator commands are seen in the fourth row, and the throttle, aileron, rudder, and elevator rates in the fifth row.

The indicated airspeed varies heavily with the turbulent wind conditions. The objective of the complimentary filtering of the indicated airspeed and additional low-pass filtering of the estimated acceleration along the flight path described in Section 5.2.5 is to reduce the impact of the noise resulting in turbulent conditions on the throttle activity. Although the noise impact is reduced on the throttle command, compared to for example the normal specific force command, the autothrust controller operates in the entire permissible throttle command range in order to counteract the airspeed deviations. Thus, the indicated airspeed is maintained at about plus/minus 2 meters per second from the desired target.

Track angle commands are acquired and maintained within 1 degree at angles of sideslip varying around $\beta_A = 10$ degrees. Bank angle commands vary at about plus/minus 4 degrees to maintain the track angle.

Flight path angle commands are acquired and maintained within 1 degree, with large pitch attitude variations of up to 8 degrees.

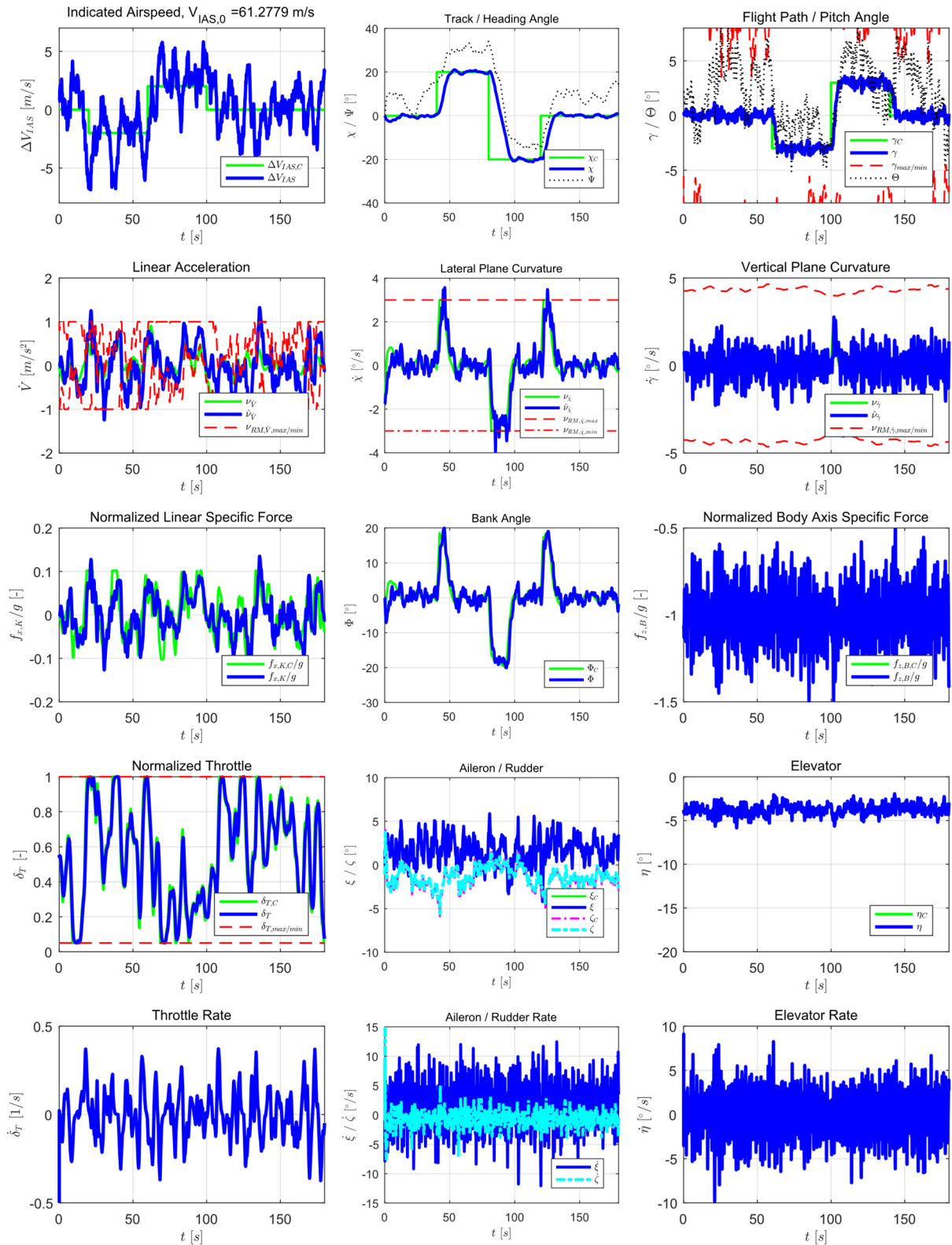


Figure 8.16: Speed and flight path command tracking during combined nominal vertical and lateral flight path and speed commands, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.

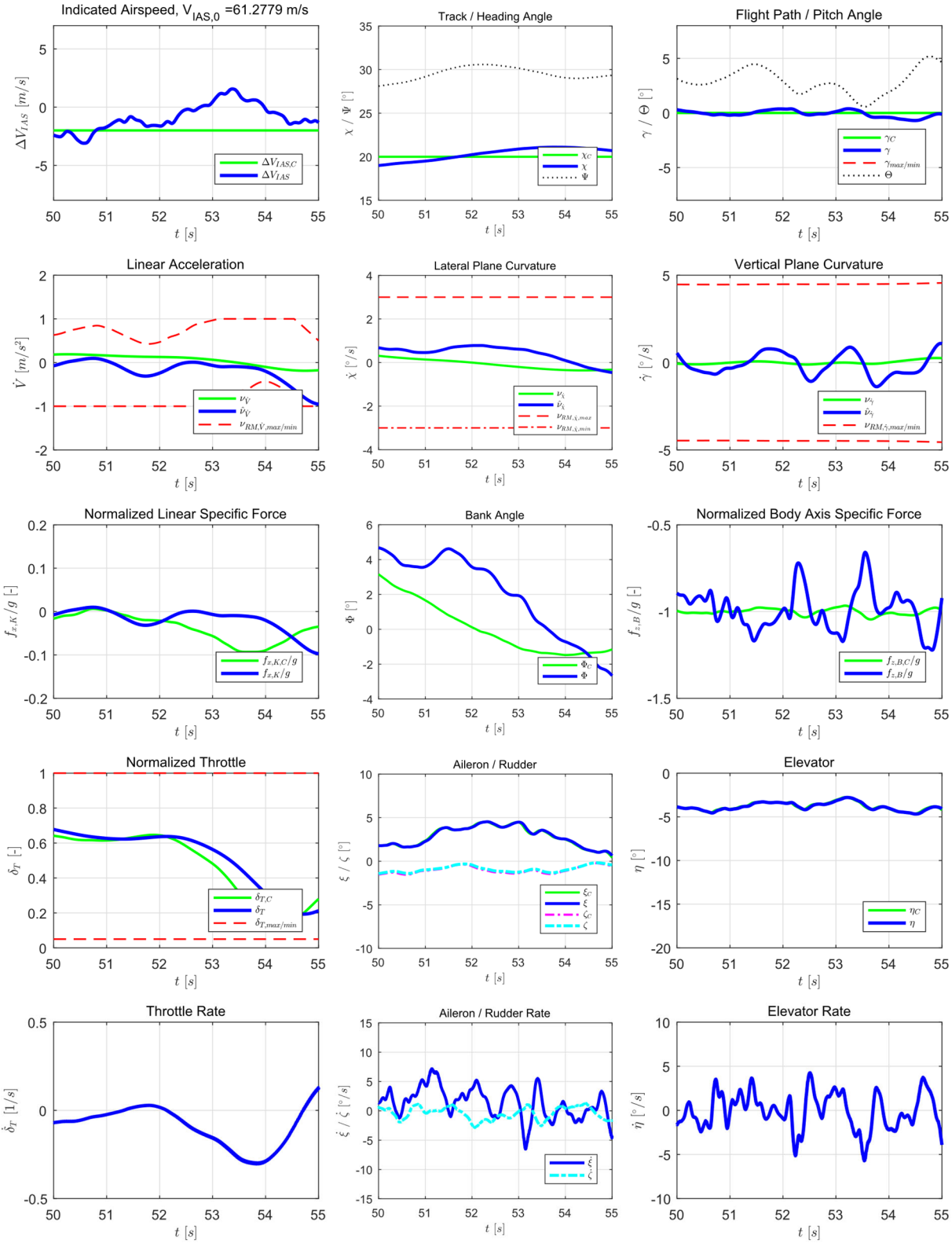


Figure 8.17: Excerpt of speed and flight path command tracking in Figure 8.16, $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.

8.3.3 Control Objective Conflict Resolution

Figure 8.18 to Figure 8.21 shows system responses for simultaneous speed and path tracking in smooth air according to Maneuver 2 defined in Figure 8.14, for speed and lateral plane curvature as well as flight path and vertical plane curvature priority.

Figure 8.18 and Figure 8.20 show the responses with the AFCS in speed and lateral plane curvature priority, i.e. $\kappa_E = 1$ according to Eq. (5.91)-(5.92) and $\kappa_F = 1$ according to Eqs. (5.126)-(5.127). Figure 8.18 shows the responses for a test point representing an indicated airspeed of about 61 meters per second (ca 119 knots) and an altitude of about 1500 meters, in a gear up and flaps up configuration, whereas Figure 8.20 shows the maximum and minimum response curves over the entire envelope.

Figure 8.19 and Figure 8.21 show the responses with the AFCS in flight path and vertical plane curvature priority, i.e. $\kappa_E = 0$ according to Eq. (5.91)-(5.92) and $\kappa_F = 0$ according to Eqs. (5.126)-(5.127). Figure 8.19 shows the responses for a test point representing an indicated airspeed of about 61 meters per second (ca 119 knots) and an altitude of about 1500 meters, in a gear up and flaps up configuration, whereas Figure 8.21 shows the maximum and minimum response curves over the entire envelope.

Speed vs. Path Prioritization

In Figure 8.18, the active sacrifice of the flight path angle tracking in order to prioritize indicated airspeed tracking can be seen. As the throttle setting becomes saturated, the flight path limits are adjusted based on the desired acceleration, and the flight path is sacrificed. In Figure 8.19 the opposite occurs. The acceleration limits are actively adjusted in order to sacrifice the airspeed tracking to maintain the desired flight path angle. Figure 8.20 shows that the airspeed tracking is contained over the envelope. The minimum and maximum response curves represent the reduced airspeed tracking performance at the edges of the envelope, where the throttle is saturated. In Figure 8.21, it can be seen that the airspeed tracking is sacrificed to a great extent of the envelope to maintain desired flight path tracking, which is prioritized.

Vertical vs. Lateral Plane Curvature Prioritization

To test the vertical versus lateral plane curvature prioritization, Maneuver 2 includes a large positive flight path angle command in the middle of a long turning maneuver at $t = 100$ s, where the flight path angle command changes from $\gamma_C = -6$ degrees to $\gamma_C = 6$ degrees, resulting in a pull-up maneuver. The inner loop command limit is set to $f_{z,B,C,min} = -1.5$, which triggers an active prioritization of the control plane curvature during the pull-up maneuver. When in lateral plane priority, Figure 8.18, the desired vertical plane curvature is limited so that the desired lateral curvature can be maintained. When in vertical plane priority, Figure 8.19, the bank angle is temporarily reduced from $\Phi = 30$ degrees to about $\Phi = 10$ degrees in order to utilize the full pull-up authority to achieve the desired flight path angle, before returning to completing the turn.

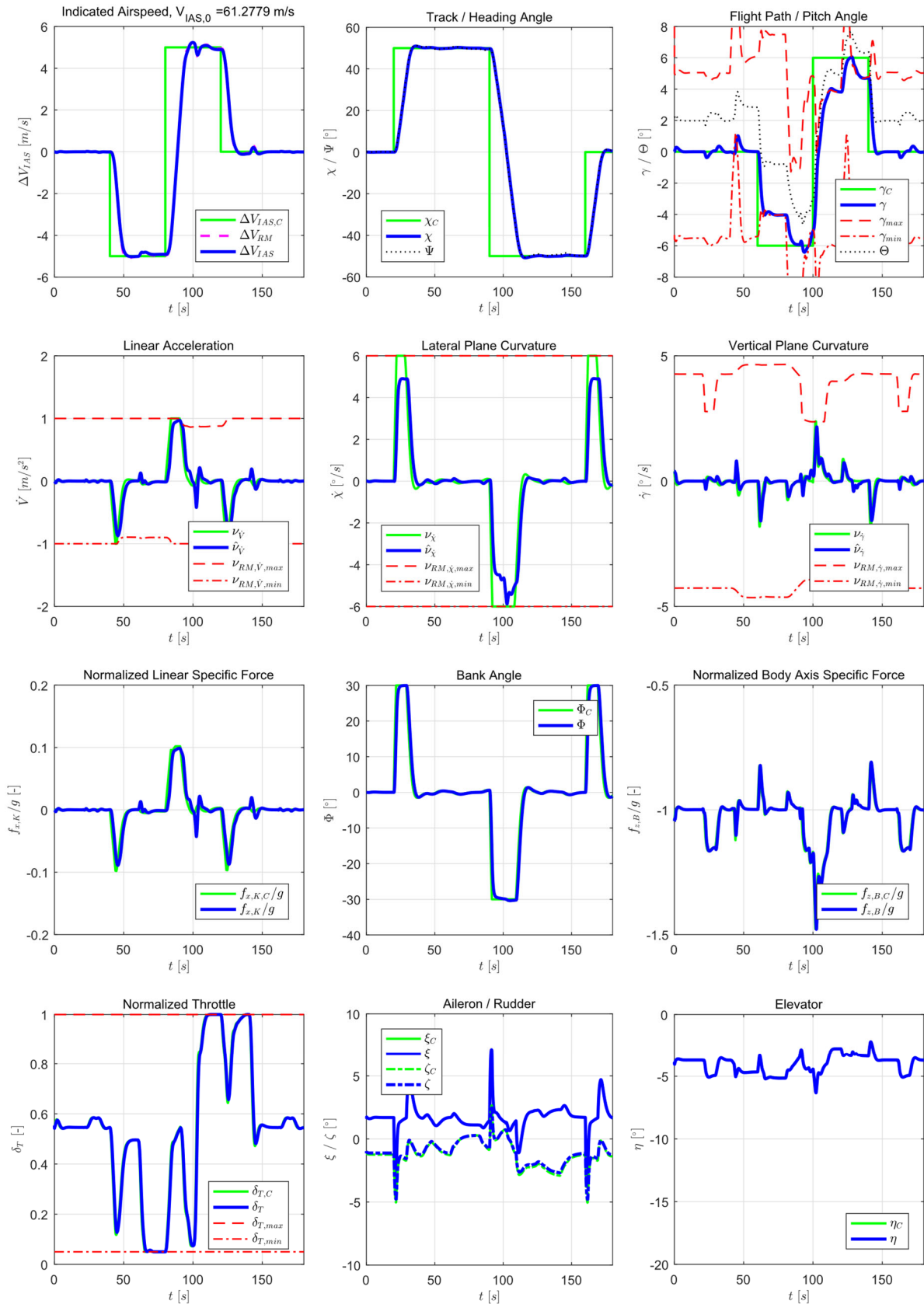


Figure 8.18: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Speed and lateral plane curvature priority. $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.

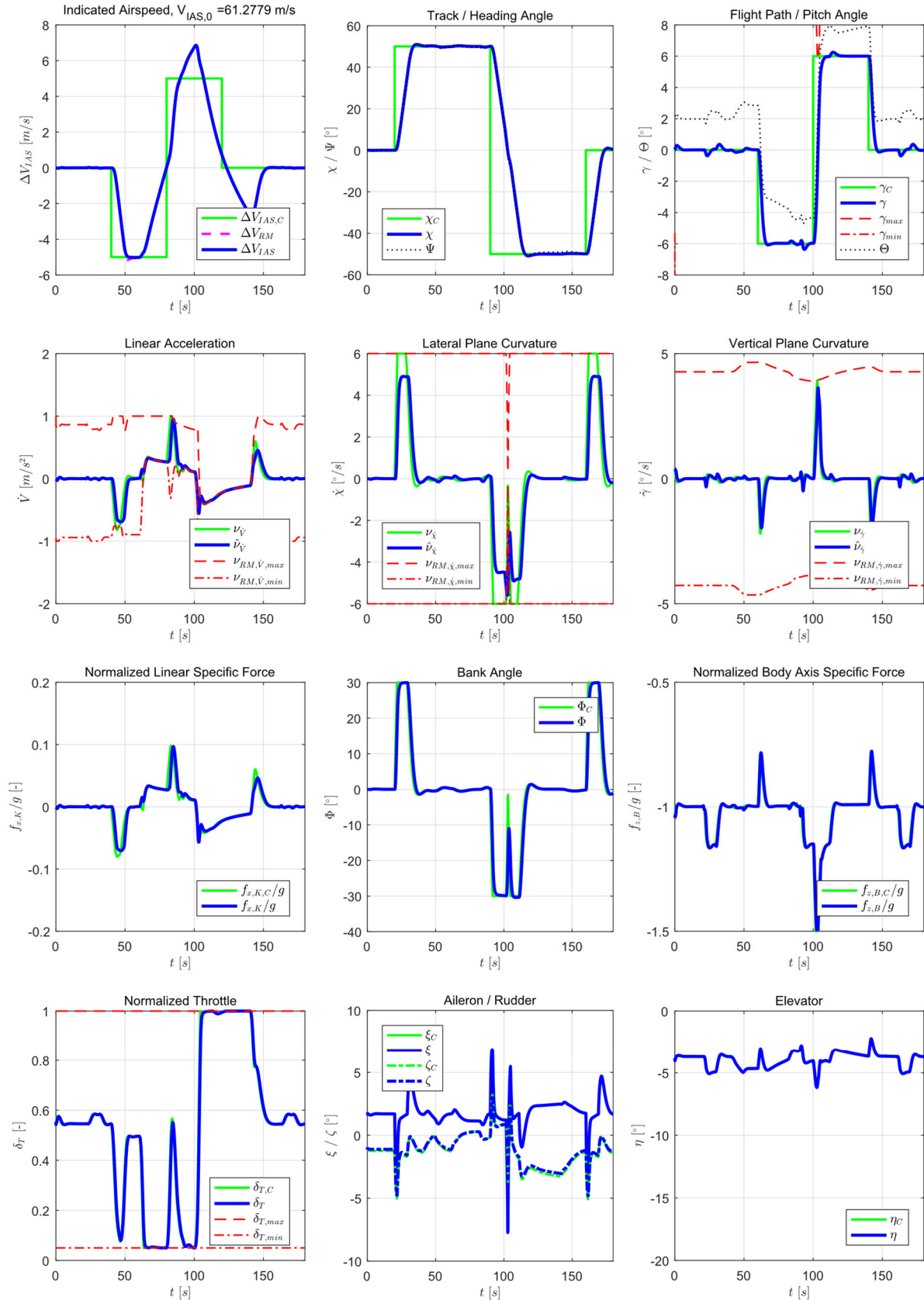


Figure 8.19: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Vertical path and vertical plane curvature priority. $V = 61$ m/s, $h = 1500$ m, gear up/flaps up configuration.

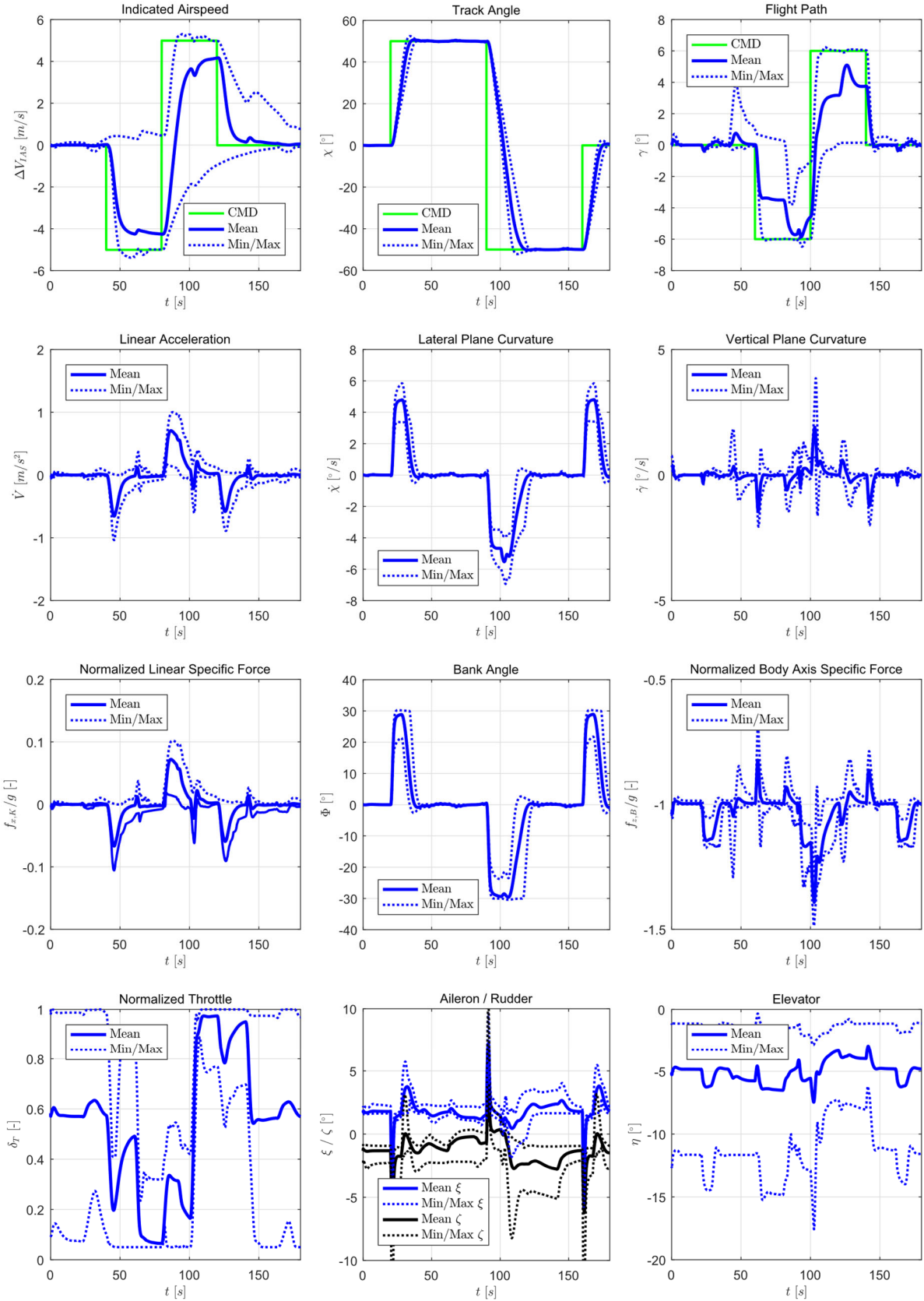


Figure 8.20: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Speed and lateral plane curvature priority. Evaluated over envelope for gear up/flaps up configuration.

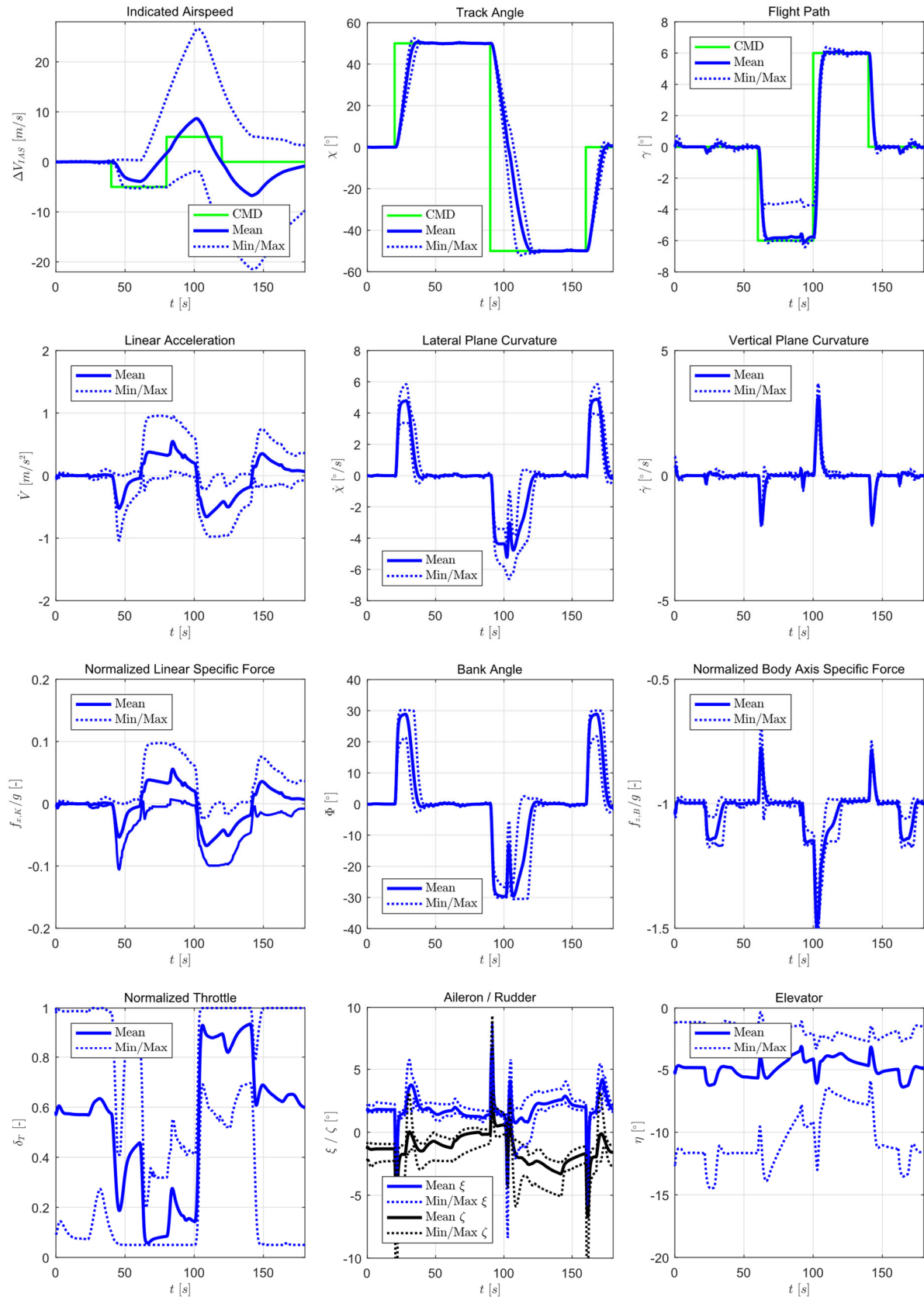


Figure 8.21: Maneuver 2, airspeed and path command tracking during combined vertical and lateral flight path and speed commands. Vertical path and vertical plane curvature priority. Evaluated over envelope for gear up/flaps up configuration.

8.3.4 Energy Protections

The energy protection functionality introduced in Section 5.2.8 actively adjusts the flight path limits at the edges of the airspeed envelope to automatically prioritize the airspeed, and smoothly acquire the limit airspeed without over- or undershoot and the corresponding steady-state flight path angle.

Figure 8.22 shows a demonstration of a low-speed protection (left column) and a high-speed protection (right column). The indicated airspeed limits are for this test set at $V_{IAS,min} = 46.3$ m/s, corresponding to 90 kts, and $V_{IAS,max} = 64.3$ m/s, corresponding to 125 kts. These were the limits that were set during the flight testing presented in Section 8.4.3. The limits are well within the envelope shown in Figure 8.1, but this was desired for initial flight test validation of the energy protections.

The low-speed protection is triggered by a large flight path angle change ($\gamma_C = 10$ degrees), where the throttle command is saturated, and the speed quickly bleeds off. The lower acceleration limits are smoothly adjusted as the speed bleeds off, and the acceleration enters the transition region defined in Section 5.2.8. The upper flight path limit is calculated based on the lower acceleration bound and is simultaneously reduced until the desired acceleration is equal to zero and the speed has reached the minimum airspeed limit.

The high-speed protection is analogously triggered by a large negative flight path angle change ($\gamma_C = -10$ degrees), where the speed quickly builds up. The upper acceleration limit and lower flight path angle limit forces the nose up until the airspeed stabilizes at the maximum permissible airspeed limit.

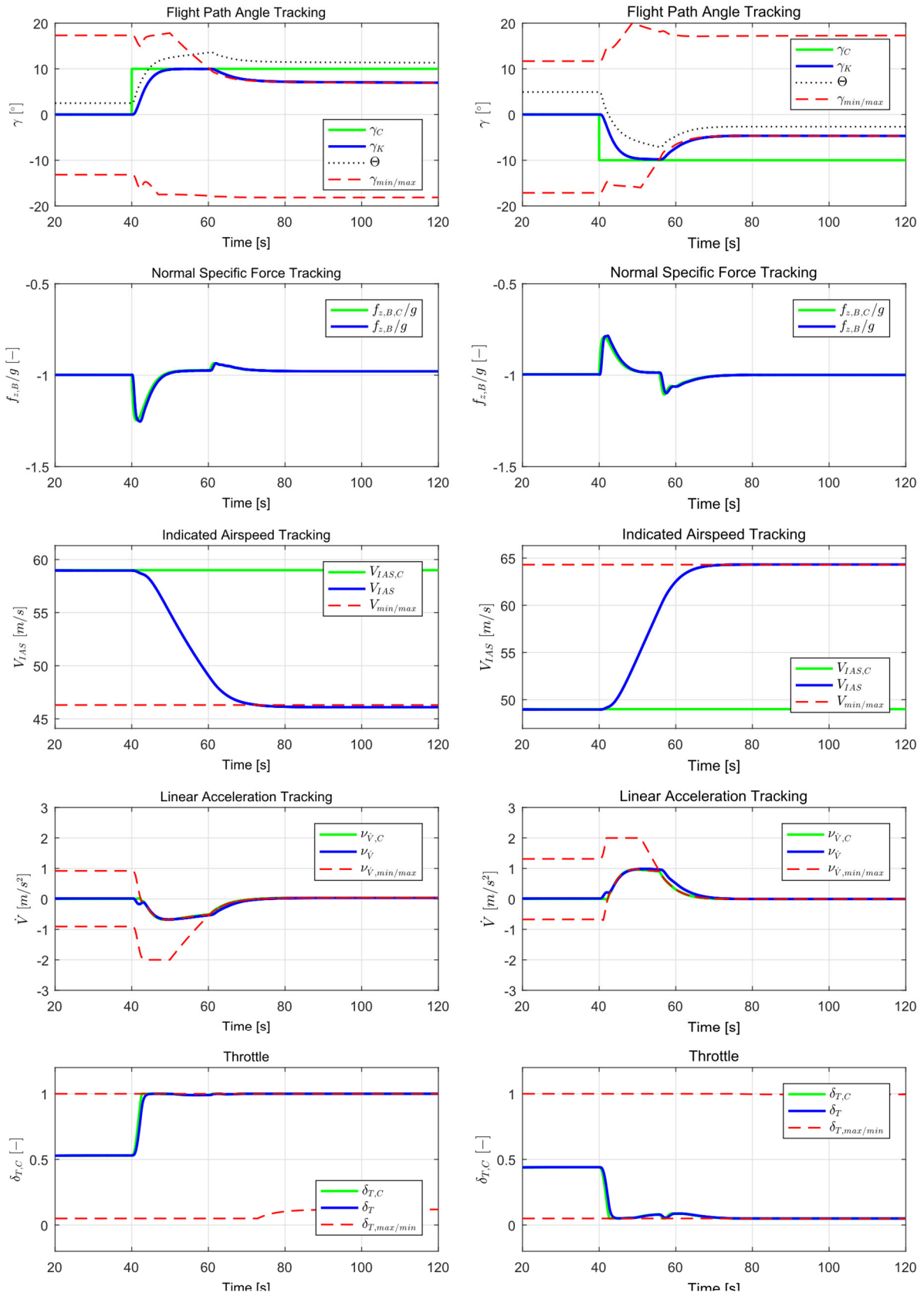


Figure 8.22: Demonstration of low-speed protection (left column) and high-speed protection (right column) responses. Gear up, flaps up configuration.

8.3.5 Mode Transition Logic

For the verification of mode transitions, a large set of tests was defined, representing different climbing, descending, and turning maneuvers, simulating manual and automatic mode changes.

Figure 8.23 shows a maneuver triggering an automatic open climb as well as a descent through an altitude command change, with automatic transitions to altitude capture and altitude hold modes. The first row shows the active vertical plane and associated speed control mode, as well as whether the altitude capture mode is armed or not. A mode change is highlighted throughout the subplots of the figure as a vertical solid, black line.

The second row shows the incremental altitude change. The open climb is triggered at $t = 20$ seconds by a step change in the altitude when in the altitude hold mode. The new target altitude is smoothly captured and maintained. A subsequent open descent is triggered at $t = 140$ seconds, resulting in a return to the initial altitude.

The third row shows the flight path angle tracking, with the maximum and minimum flight path angle limits from the speed by pitch logic presented in Section 5.2.5 and illustrated in Figure 5.14151. The flight path angle limits are initialized at the current flight path angle upon activation of the open climb/descent and slowly expanded to the nominal flight path limits from the prioritization mode control through a rate-limited PT1 filter. When in an open climb, the minimum flight path angle limit is set to $\gamma_{min,SBP} = 0$ degrees, in order to prevent the aircraft from transitioning to a temporary descent when accelerating to acquire a new speed target. This can be observed at $t = 40$ seconds when a positive speed command results in the flight path being reduced to zero to redistribute the energy to accelerate to acquire the new speed target but preventing a temporary descent. Analogously, at $t = 160$ seconds, a reduced speed command when in the open descent results in the flight path angle increasing to level flight in order to slow the aircraft down, but not result in climbing flight. The fourth row shows the normal specific force tracking by the inner loop controller.

The fifth row shows the indicated airspeed tracking, and the sixth row the corresponding acceleration command from the speed loop. A speed command doublet is introduced as discussed in the previous paragraph to verify the flight path angle limitation in the speed by pitch mode.

The seventh row shows the throttle command from the autothrust controller, operating either to track the desired acceleration or provide a fixed climb or descent thrust setting. The transition to the climb thrust setting or the descent thrust setting, respectively, is commanded at a fixed throttle rate as presented in Section 5.2.11.

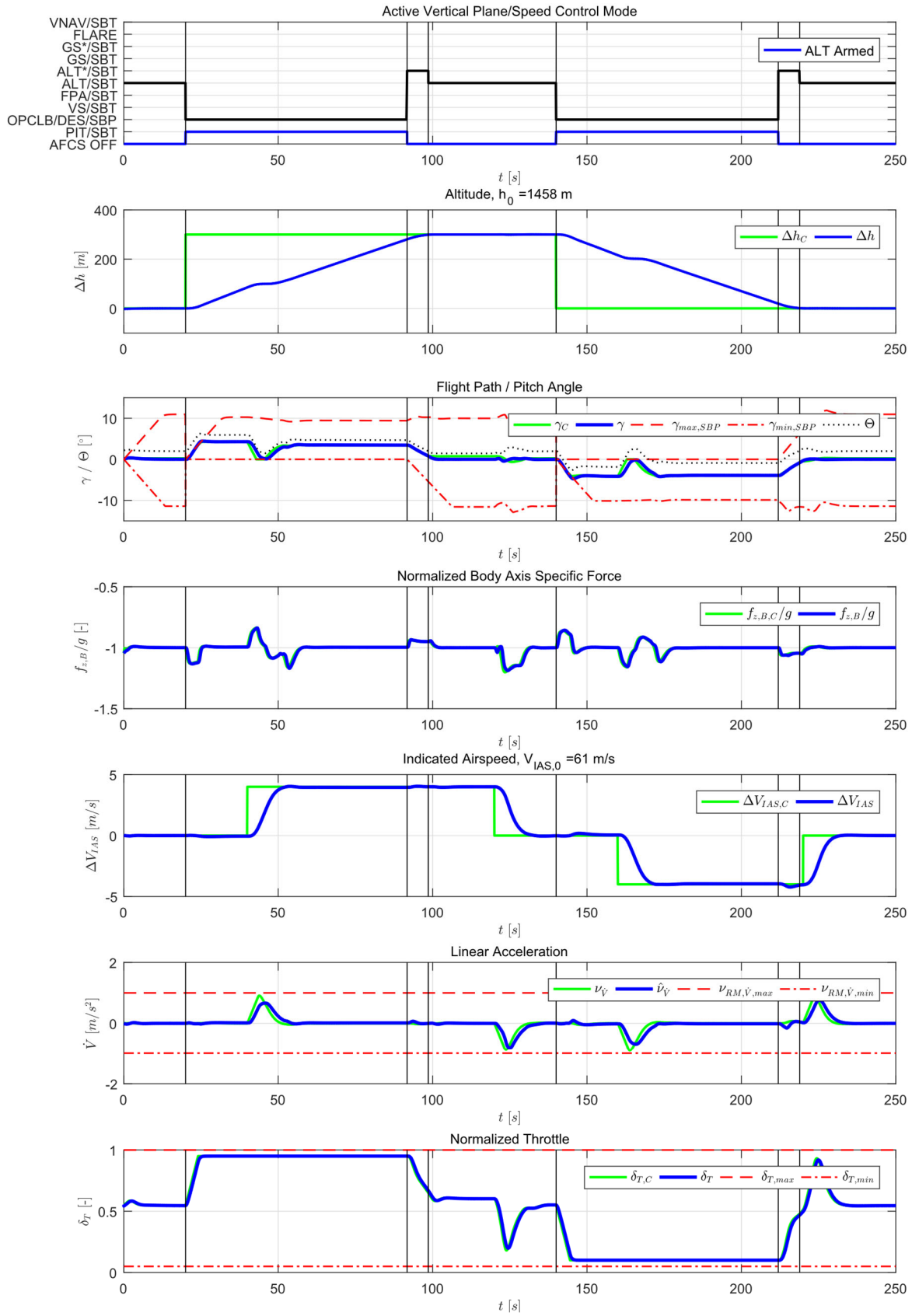


Figure 8.23: Demonstration of an open climb/descent maneuver with altitude capture and hold transitions. Mode changes indicated by a solid, black vertical line.

8.4 Flight Testing

Extensive AIL simulations, ground taxi tests, and direct law flight tests built confidence in the system stability and disconnect mechanisms before deployment of the automatic flight control functionalities in flight tests. Initial deployment and flight testing of the modular FGCS on the DA42 OE-FSD was performed in early 2016. The deployment of sensors and control modules followed a sequential order, with incremental testing of sensor characteristics and performance, validation of controller behavior open loop (with actuator clutches open), closed-loop performance of inner loop controller, then AFCS, and finally waypoint-based trajectory navigation, as well as full remote control from an HMI-equipped ground station.

Table 8.1 summarizes the history of the modular FGCS and particularly the AFCS flight test activities performed to which the author contributed with planning and execution, and from which results are included in this thesis. Section 8.4.1 further details the modular FGCS deployment flight tests in January 2016, verifying the initial performance of the AFCS and overall flight control system. Sections 8.4.2 and 8.4.3 summarize flight testing performed in August 2016, where additional AFCS control modes and the energy protection functions were engaged. Flight test cards for the AFCS deployment and additional flight test logs are included in Appendix D.1.

Flight testing occurred primarily at Diamond Aircraft facilities at Wiener Neustadt East Airport (ICAO airport code LOAN), Austria, where the DA42 OE-FSD was stationed. Additional flight tests were performed at Mindelheim-Mattsies Airport (ICAO airport code EDMN). Flight testing activities were planned and performed together with and under the supervision of Diamond Aircraft test pilots, acting as safety pilots during flight tests.

Several demonstration flights were performed for various stakeholders to showcase the functionalities and performance of the flight control system. These flights typically followed a preprogrammed trajectory with waypoint navigation, with an intermittent demonstration of AFCS control modes, but no formal verification according to prepared flight test cards. However, the demonstration flights provided valuable validation of the control system characteristics and performance under various atmospheric and aircraft configuration conditions.

Table 8.1: Flight test history of the modular FGCS to which the author contributed with planning and execution, and from which results are included in this thesis.

Test Date	Test Location	Test Scope
2015	LOAN	Various flight and ground/taxi tests to validate sensor characteristics and performance, and the actuation system performance in direct law
2016-01-14	LOAN	Carry flight tests monitoring FCS open loop Inner loop deployment

Test Date	Test Location	Test Scope
2016-01-15	LOAN	AFCS deployment, initial testing of a subset of AFCS modes
2016-01-20	LOAN	Continued AFCS flight testing
2016-03-03	LOAN	Remote controlled flight, operation of AFCS from MCP/MCMD ground station
2016-03-17	LOAN	ATOL preparatory testing
2016-04-07	EDMN	Demonstration flight, demonstration of trajectory navigation and various AFCS modes
2016-04-08	EDMN	Demonstration flight, demonstration of trajectory navigation and various AFCS modes
2016-08-17	LOAN	Preparatory demonstration flight, demonstration of trajectory navigation and various AFCS modes AFCS flight test, additional control modes and energy protections, test of flare mode in preparation for later ATOL flight testing
2016-08-18	LOAN	Demonstration flight, demonstration of trajectory navigation and various AFCS modes
2016-08-18	LOAN	AFCS flight test, additional control modes and speed protections, test of flare mode in preparation for later ATOL flight testing

8.4.1 Modular FGCS Deployment

Following various flight and ground/taxi tests to validate sensor characteristics and performance, as well as the actuation system performance in direct law configuration (i.e., a direct stick deflection to actuator position mapping), deployment flight tests of the modular FGCS were performed in January 2016.

Deployment of the control loops was performed sequentially. First, the flight control algorithm outputs were monitored open loop in carry flight tests, with the actuation system disengaged from the physical control surface mechanics. Thus, the dynamics and direction of the commands sent to the actuators were validated. Second, the correct functionality and performance of the inner loop controller were verified with preprogrammed command signals

sent to the inner loop from the system automation logic. With successful basic verification of the inner loop, the AFCS control loops were stepwise engaged and verified.

For the deployment of the AFCS control loops, a set of flight test cards was developed, defining a test sequence where the basic attitude hold modes implemented for the DA42 OE-FSD were first engaged, followed by activation of the flight path modes, altitude hold mode, altitude acquire and capture, combined maneuvers with climbing and descending during lateral path changes, followed by the activation of the trajectory controller for waypoint navigation tests. The flight test cards for the AFCS deployment are included in Appendix D.1. For the inner loop deployment, the flight test input commands were preprogrammed and triggered by the test pilot onboard. For the AFCS deployment, the function activation and command selection according to the flight test cards were performed manually by the test pilot onboard using the MCP/MCMD. The test and safety pilot quickly gained confidence in the behavior and performance of the flight control system, and deviated somewhat from the flight test cards, leading to only partial results compared to the flight test plan.

Figure 8.24 and Figure 8.25 show the AFCS deployment flight test, test flight 1 on January 15, 2016. Figure 8.24 shows the active vertical plane/speed control mode, altitude, and indicated airspeed tracking, vertical speed tracking, pitch angle tracking as well as inner loop incremental normal specific force tracking. Figure 8.25 shows the active lateral control mode, heading tracking as well as inner loop bank angle tracking. The time axis of the test flight starts during system start-up on the ground and does not represent flight time. The AFCS deployment test starts at around 34 minutes from the start of the recording and ends a little after 60 minutes.

Controller activation occurred at around 34.8 minutes, at level and wings-level flight at around 5500 feet altitude and an indicated airspeed of 100 knots, in a clean configuration (see flight test card #1, Figure D.2 in Appendix D.1). Atmospheric conditions were moderately turbulent. First, the pitch and roll hold modes were activated, as they are default modes for the controller implementation for the DA42 OE-FSD.⁵ The activation of the pitch and roll hold modes is shown in Figure 8.26. At activation, the active vertical and lateral mode switches from AFCS OFF to PIT and ROL, respectively, and the pitch and bank angle commands are initialized at the current aircraft attitude. The incremental normal specific force and bank angle commands are equal to zero when the AFCS is OFF and are likewise initialized at the current aircraft state upon controller activation. Pitch angle tracking is maintained within one degree and bank angle tracking within two degrees, despite the moderately turbulent flight conditions.

⁵ The control law design of the pitch hold mode is not elaborated in this thesis, as the focus is put on the development and verification of the coupled flight path control structure. The roll hold mode is simply a forward of the bank angle command to the inner loop controller and involves no additional control law on the AFCS part.

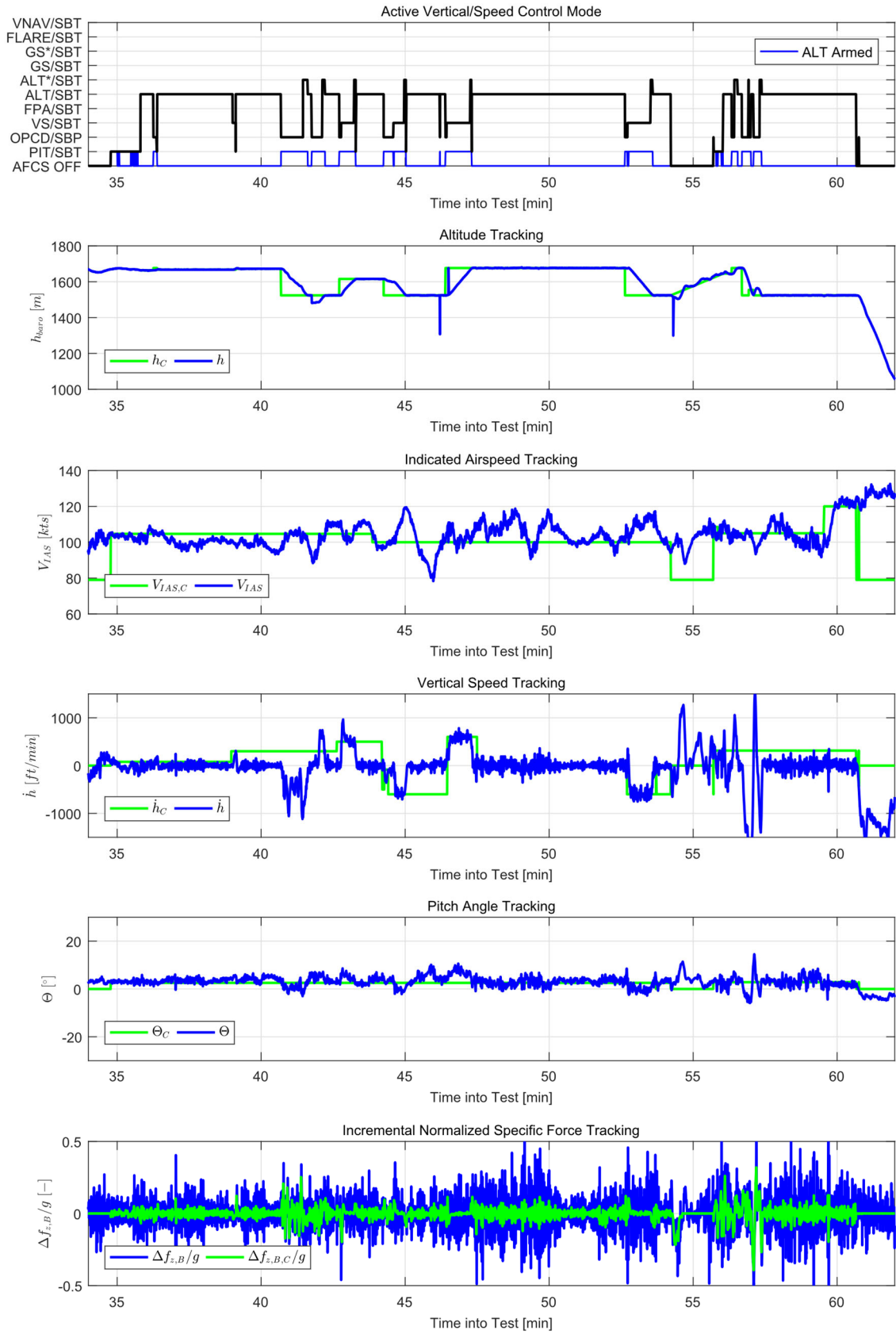


Figure 8.24: AFCS deployment flight test, flight test #1, 2016-01-15. Vertical plane/speed active control modes and command tracking.

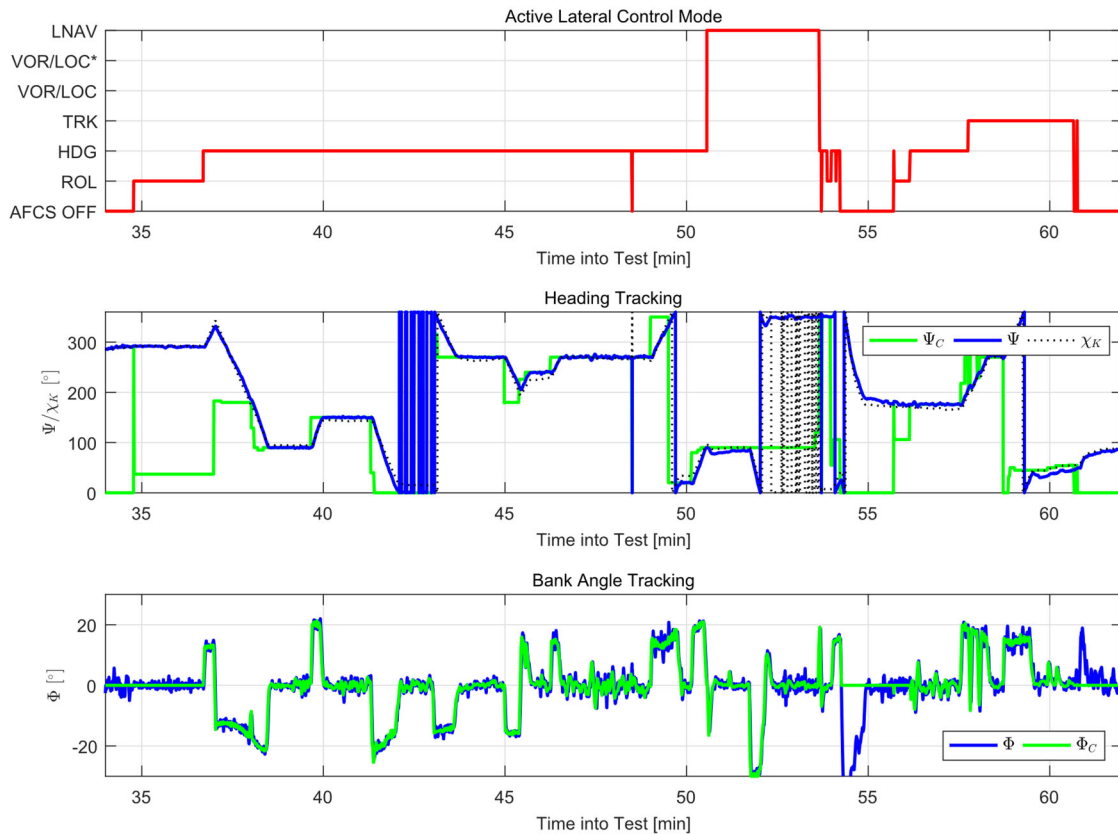


Figure 8.25: AFCS deployment flight test, flight test #1, 2016-01-15. Lateral plane active control modes and command tracking.

Following controller activation in pitch and roll hold modes, first, the altitude hold mode was activated, holding the current altitude, and then the heading mode. The heading mode activation, shown in Figure 8.27, occurred at around 36.5 minutes. The trajectory controller maintains the altitude within one meter. The flight test cards prescribed a heading target selection of the current heading and subsequent heading mode activation, followed by a series of small and then larger turns. However, during the flight test, an initial heading target was selected by the test pilot that initiated a right-hand turn upon heading mode activation, followed by a new heading target that initiated a left-hand turn, and a subsequent series of heading target changes before a steady heading target was captured. The heading mode initiated the turns and acquired the selected headings with a standard rate turn as expected. The heading target was captured without overshoot and maintained within 0.5 degrees.

The vertical speed mode activation followed at around 44 minutes, performing a series of small altitude acquire and capture maneuvers combined with simultaneous heading changes, shown in Figure 8.28. Climb and descent were performed at a vertical speed of 600 feet per minute, maintained within 150 feet per minute.

Results from the initial flight tests of the AFCS were very satisfactory, and no major retuning of the path loop parameters was required, only minor redesigns based on updated inner loop gain and parameter design iterations.

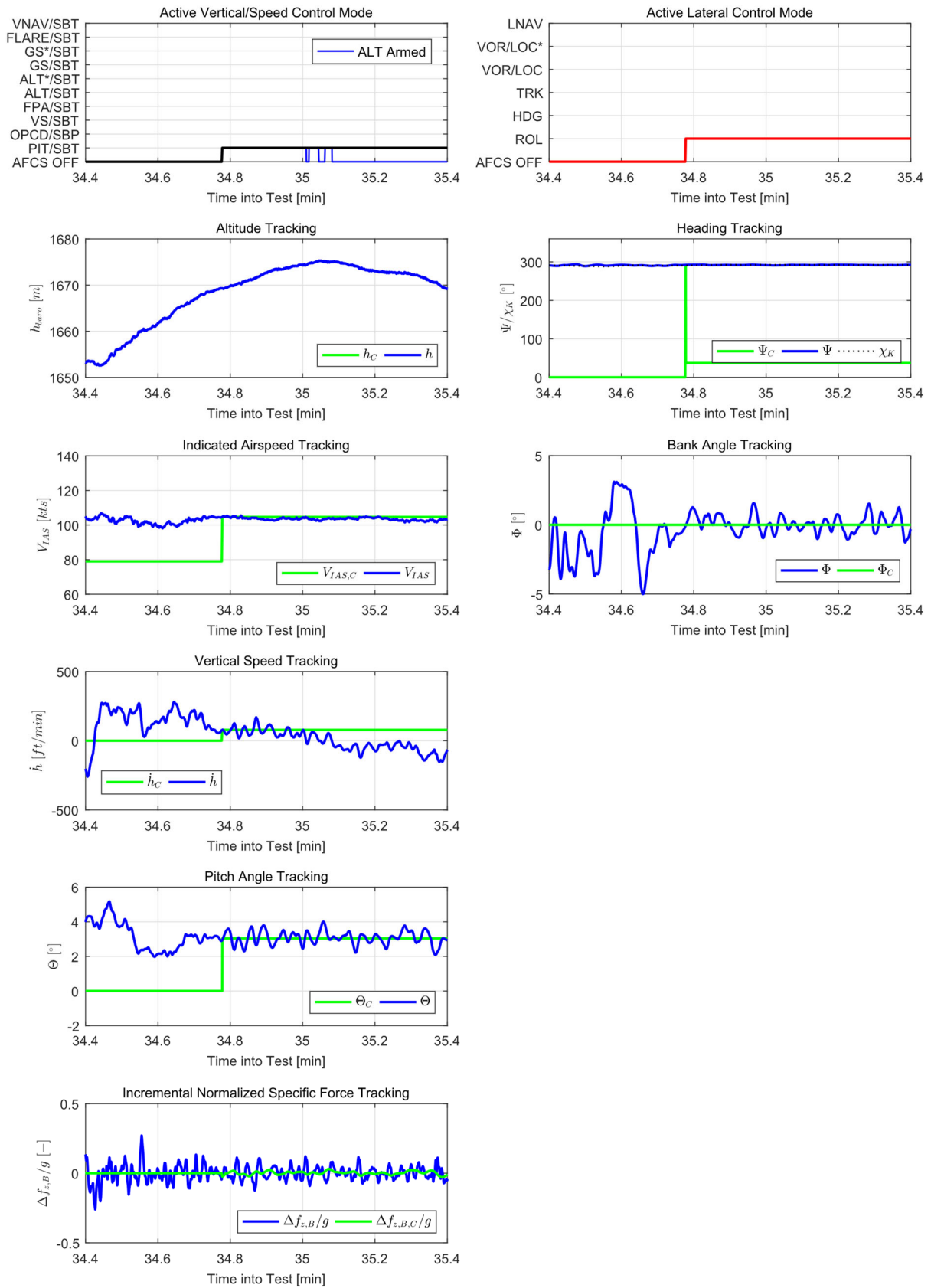


Figure 8.26: First AFCS controller activation, flight test #1, 2016-01-15. Flight test card 1, activation of pitch and roll hold modes, the default activation modes of the DA42 specific implementation.

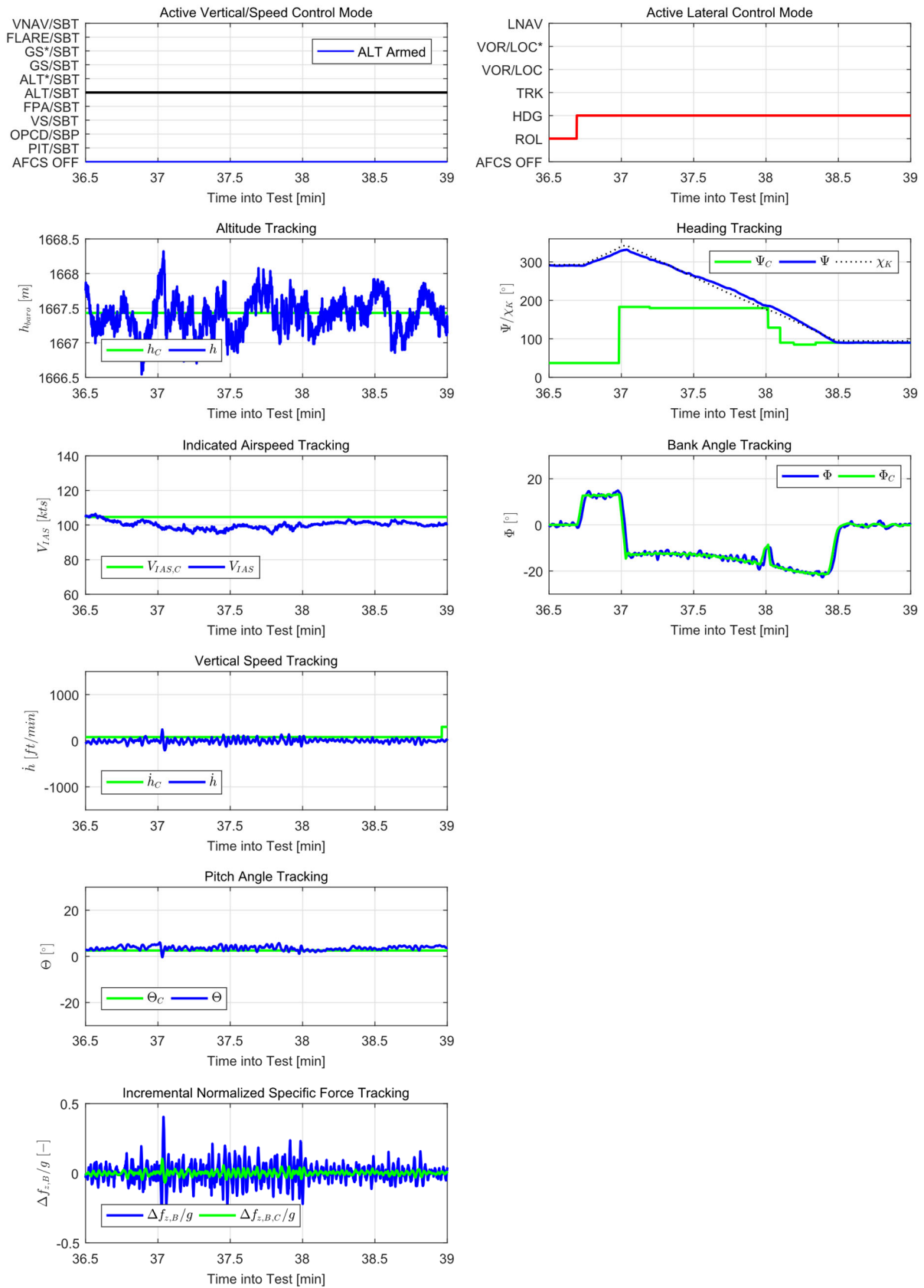


Figure 8.27: First AFCS lateral path mode activation, flight test #1, 2016-01-15. Flight test card 6, activation of heading mode, acquiring and tracking heading commands from the MCP.

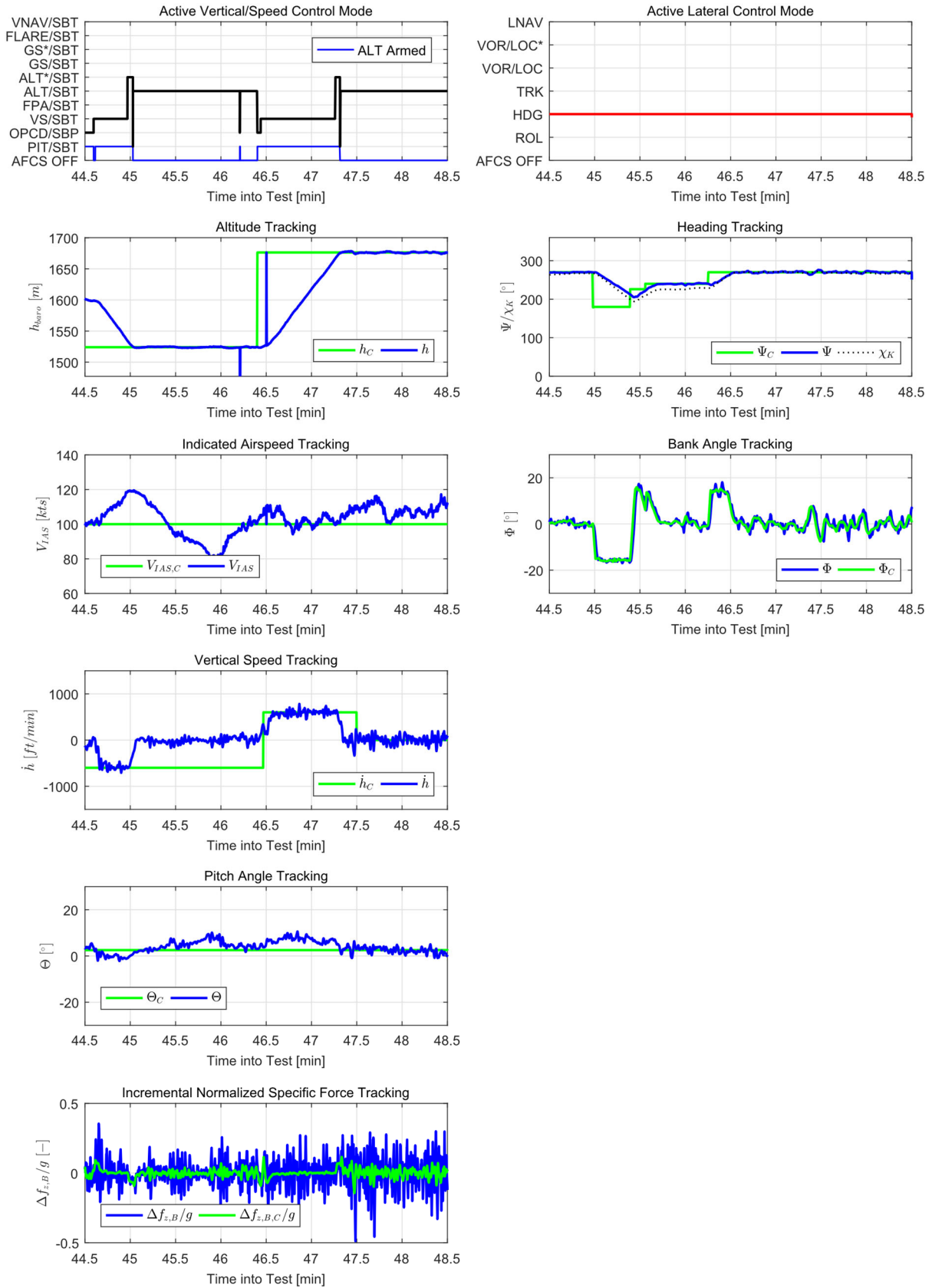


Figure 8.28: First AFCS vertical path mode activation, flight test #1, 2016-01-15. Flight test card 10, activation of vertical speed mode, tracking vertical speed commands from the MCP and acquiring and capturing altitude commands.

8.4.2 Nominal Flight Path Tracking

This section presents results from continued flight testing performed in August 2016, see Table 8.1, where additional AFCS control modes were demonstrated. Four flights were performed over two days. The full test flights are included in Appendix D.2.

Figure 8.29 shows a segment of the flight test #2, August 17. Atmospheric conditions were lightly turbulent. During the segment, the AFCS was active in all control planes: in flight path angle mode and speed by thrust, and in track angle mode. The indicated airspeed target is 100 knots with the maximum indicated airspeed set at 125 knots and the minimum indicated airspeed set at 90 knots, well within the airspeed envelope for the flaps up, gear up configuration.

The flight path angle mode is activated at time into test $t = 62.5$ minutes at a flight path angle command of $\gamma_C = -5$ degrees. The flight path angle mode shows good tracking performance with the flight path angle target smoothly acquired without overshoot and maintained within 0.2 degrees. The indicated airspeed builds up as the autothrust is saturated, to a point where the high-speed protection is briefly activated, reducing the flight path angle slightly before a new, positive flight path angle command of $\gamma_C = 5$ degrees is set at around $t = 63.1$ minutes. The flight path angle target is again smoothly acquired, and the indicated airspeed returns to the commanded 100 knots.

The track angle mode is active during the entire segment shown in Figure 8.29, showing good tracking performance, initially maintaining a track angle target of $\chi_C = 210$ degrees and a sideslip angle of about $\beta_A = 7$ degrees. At $t = 62.5$ minutes, a left-hand turn to track $\chi_C = 260$ degrees is commanded, acquired at a standard turn rate of 3 degrees per second and a bank angle of around $\Phi = 20$ degrees, smoothly captured and maintained within 0.5 degrees. Subsequently, a right-hand turn to $\chi_C = 160$ degrees is commanded, acquired, and smoothly captured.

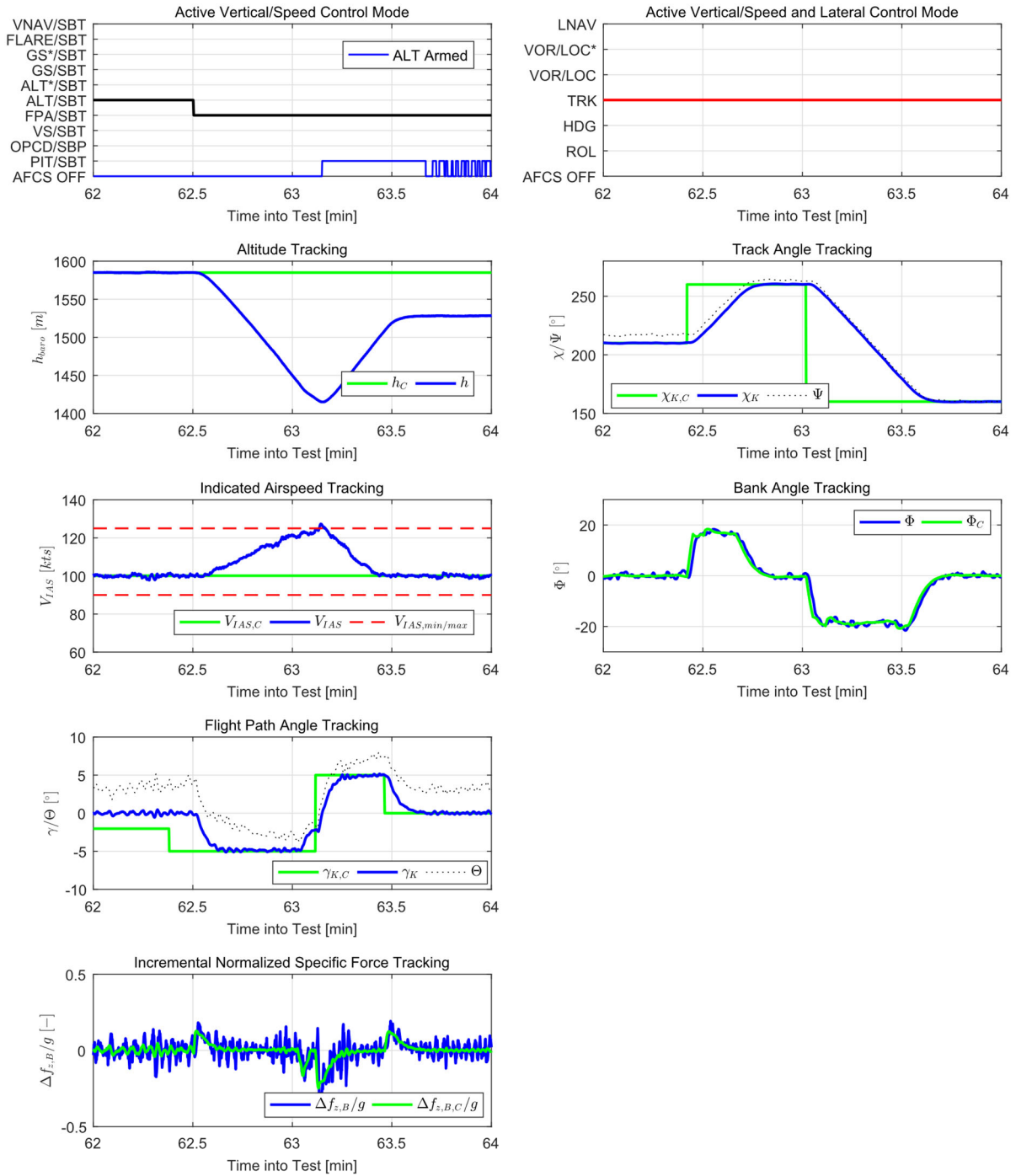


Figure 8.29: Section of flight test #2, 2016-08-17. Vertical and lateral path as well as speed command tracking in flight.

8.4.3 Energy Protections

Figure 8.30 shows a different segment of the flight test #2, August 17, compared to Figure 8.29, where low and high-speed protections are actively triggered. Indicated airspeed target is again 100 knots with maximum indicated airspeed set at 125 knots and minimum indicated airspeed set at 90 knots, well within the airspeed envelope for the flaps up, gear up configuration.

The low-speed protection is triggered by a large flight path angle change ($\gamma_c = 10$ degrees), where the throttle command is saturated and the airspeed bleeds off. The flight path angle is reduced in order not to undershoot the minimum airspeed limit, which is smoothly acquired and maintained within 2 knots.

The high-speed protection is analogously triggered by a large negative flight path angle change ($\gamma_c = -10$ degrees), where the speed quickly builds up. The flight path angle is slowly increased in order not to overshoot the maximum airspeed limit, which is maintained within 3 knots. This is a slightly larger tracking variation compared to the low-speed protection test segment, but it can be seen from the normal specific force noise levels, that the atmospheric conditions are somewhat more turbulent during the high-speed protection test segment.

Due to restricted flight testing time available, and suboptimal weather conditions, the level of correspondence with simulation results was hard to establish, although correct functionality was verified.

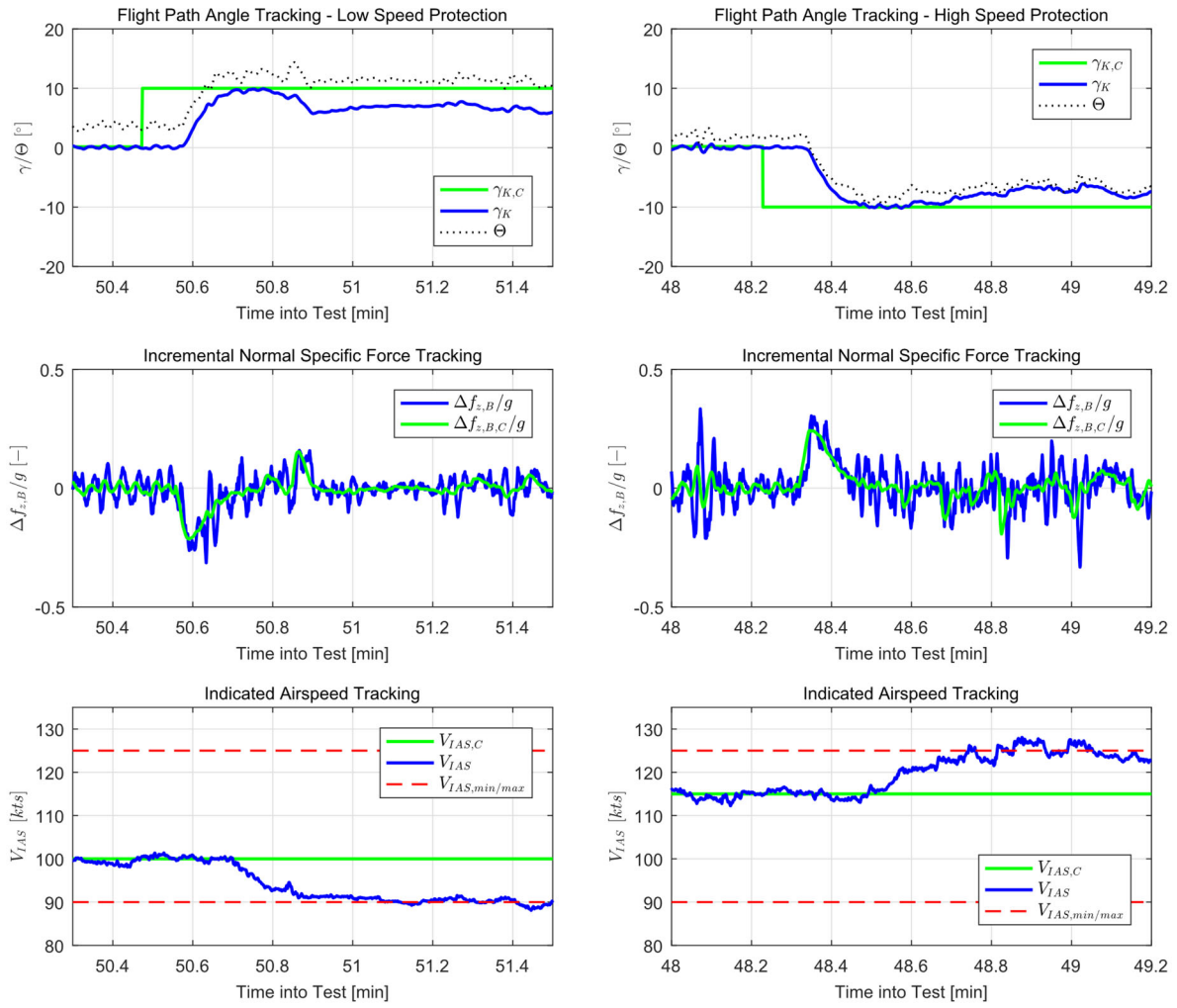


Figure 8.30: Section of flight test #2, 2016-08-17. Vertical and lateral path as well as speed command tracking in flight.

9 Conclusions and Outlook

In this thesis, the development and flight testing of an automatic flight control system, as part of a modular flight guidance and control system developed at TUM-FSD, has been presented, along with flight test results from initial controller deployment and subsequent test campaigns on the DA42 OE-FSD flying testbed.

First, a tailored system development process considering applicable guidance material for the development of civil aircraft and systems in general, and automatic flight control systems in particular, was defined. System functions and requirements were developed on the basis of a principal automatic flight control system architecture with generic architecture elements. The modular functional architecture supports the definition of generic as well as application-specific functions and corresponding system requirements. Thus, the automatic flight control system may be tailored for a range of operational scenarios for manned and unmanned aircraft of different sizes, envelopes, and regulatory environments.

Subsequently, the controller architecture was developed, based on a nonlinear inversion of the aircraft flight path dynamics. A cross-coupled reference model architecture was implemented to enable an active control objective prioritization strategy between the desired speed and flight path tracking as well as vertical and lateral flight path curvature tracking. A mode control and monitoring interface was concurrently developed for the operation of the automatic flight control system in desktop simulations, hardware laboratory setup, and flight tests. Both the controller architecture and mode control and monitoring interface were based on a modular implementation and customized for different demonstration platforms, including the DA42 OE-FSD, the Do-228 D-CODE experimental autopilot, and the SAGITTA Research Demonstrator.

A functional hazard assessment was performed based on the functional design, and different system architecture strategies and their implication on the development assurance levels of the control system elements discussed.

The implementation of the design model for the flight control algorithms was based on a model-based software development framework, with which the tailored system development process aligns. The automatic flight control system design model was integrated with the interfacing flight control algorithm modules and extensively tested in a set of model-in-the-loop environments before automatic source code generation, hardware-software integration, and hardware-in-the-loop testing. Controller functional and performance requirements were

verified in linear as well as nonlinear assessment with automated requirements evaluation over the envelope for applicable aircraft configurations. Finally, the automatic flight control system was deployed, and flight tested on the DA42 OE-FSD with satisfactory results.

9.1 Contribution 1: Tailored System Development Process

The first contribution of the thesis is a tailored system-level application of a subset of the ARP4754B development process model for the purpose of a model-based flight control law development, described in Chapter 2. The development and implementation of the automatic flight control system module was performed according to a modular model-based software development process developed at TUM-FSD. The tailored system-level process aligns with the model-based software development process and contributes with a system-level perspective on the development, including a life cycle model for demonstration platforms, development of system requirements, functional safety assessment, system architecture considerations, and assignment of development assurance levels in an automatic flight control system context.

9.2 Contribution 2: NDI-Based Flight Path Control Application

The second contribution of the thesis is an application example of a reference model architecture for NDI-based flight path control with design, implementation, and flight test demonstration. The basic control principle for the control of the vertical and lateral flight path modes, i.e. vertical speed/flight path angle and heading/track angle, as well as the airspeed along the flight path, is a reference model-based dynamic inversion of the point mass equations of motion, with pseudo-control hedging to account for the inner loop and propulsion dynamics not considered in the inversion, described in Section 5.2. The coupled architecture with cross-feeds and internal limiters allows for the easy integration and configuration of control objective prioritization strategies and additional control modes such as speed by pitch, flare, and direct flight path rate modes.

Verification and validation activities, described in Chapter 8, comprise linear and nonlinear controller assessment, model-level simulations, hardware- and aircraft-in-the-loop testing, and finally flight testing. The controller deployment strategy with corresponding flight test cards is described, with initial flight test results showing good path tracking and disturbance performance, also in moderately turbulent atmospheric conditions. Additional flight testing of the automatic flight control system showed excellent overall system performance and control module interactions. The low and high protections were validated in flight for a reduced airspeed envelope.

9.3 Contribution 3: Active Control Objective Prioritization Concept

The third contribution of the thesis is the design, implementation, and flight test demonstration of an active control objective prioritization concept. The control objective prioritization concept is integrated into the reference model architecture by cross-coupling the reference models for speed, flight path angle, and track angle, described in Sections 5.2.8 and 5.2.9. The reference

model limiters are actively adjusted based on saturated energy rate and force control constraints, providing prioritization of desired speed or vertical flight path, as well as prioritization of vertical or lateral flight path curvature during maneuvers. The approach also introduces a mixed speed versus flight path priority as well as mixed vertical versus lateral flight path curvature. The feasibility of the concept is demonstrated in high-fidelity simulations of the closed loop system, described in Sections 8.3.3 and 8.3.4, as well as flight tests, described in Section 8.4.3.

9.4 Contribution 4: Mode Control and Monitoring Interface

The fourth contribution of the thesis is the development of a compact mode control and monitoring interface, consisting of a mode control panel and a mode control and monitoring display, described in Section 5.4. The mode control and monitoring interface enables control and provides state awareness of the automatic flight control system. The architecture and design of the interface allow the configuration, installation, and operation on multiple application platforms for test pilot control of the system in-flight, as demonstrated on the Do-228 D-CODE, or configuration as a UAV or OPV ground station providing a remote operator with control and awareness via data link, as demonstrated on the DA42 OE-FSD.

The interface layout and functionality were initially developed as executable desktop applications for the model-in-the-loop simulation environment, which greatly supported control law prototyping and early validation of system operation. The design was iteratively improved based on test pilot feedback before hardware manufacturing and hardware-in-the-loop integration. Display software and mode control logic were fine-tuned in the laboratory setup and aircraft integration environments before operation and validation in flight tests and remote-control operation.

9.5 Outlook

One of the design objectives for the integrated modular FGCS developed at TUM-FSD was adherence to applicable regulations and guidance material to enable certifiability of the overall system design. The presented system development process covers important elements of the system-level ARP4754B process framework that may provide input to future system applications and the tailoring of the application-specific development process and life cycle model. One key aspect that is only briefly discussed in this thesis, which requires great attention with respect to certifiability, is the development and verification traceability from the system requirements and verification activities to the software level.

The thesis only covers the system-level functional hazard assessment of the safety assessment process. In a certifiable application scenario, this analysis would receive input from and provide information to aircraft-level safety assessment, as well as be complemented by the application-specific architecture analysis and system safety assessment activities. Here, the model-based development process could leverage an extended system simulation model, including the modeling of component failure modes, supporting, and validating the functional hazard assessment and providing architectural safety analyses with simulations of failure propagations and system effects.

The flight path controller architecture, with reference model-based dynamic inversion and an interface directly commanding the specific forces corresponding to a desired path curvature or acceleration, was developed for and applied to fixed-wing aircraft. The applicability of the approach to other aerial platform classes, such as rotorcraft, VTOL aircraft, and other configurations, has not been studied as part of the presented work, but there is nothing in principle in the path control architecture that would preclude those types of platforms.

The flight testing of the automatic flight control system was performed based on manual commands from a test pilot onboard the aircraft or an operator on the ground. Continued flight testing should include the maneuver injection capabilities of the system automation module of the modular FGCS to perform reproducible identical maneuvers over multiple envelope points, aircraft configurations, and atmospheric conditions, in order to systematically establish the performance envelope of the implemented system and its conformance to the nonlinear simulation.

Bibliography

- [1] D. McRuer and D. Graham, "Flight Control Century: Triumphs of the Systems Approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 161-173, 2004.
- [2] S. S. Vakil and R. J. Hansman Jr., "Approaches to Mitigating Complexity-Driven Issues in Commercial Autoflight Systems," *Reliability Engineering and System Safety*, vol. 75, pp. 133-145, 2002.
- [3] R. Brockhaus, W. Alles and R. Luckner, *Flugregelung*, 3rd ed., Berlin: Springer-Verlag, 2011.
- [4] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*, New York, New York: John Wiley & Sons, Inc, 1996.
- [5] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2 ed., Hoboken, New Jersey: John Wiley and Sons, 2003.
- [6] A. A. Lambregts, "Integrated System Design for Flight and Propulsion Control using Total Energy Principles," in *AIAA Aircraft Design, Systems and Technology Meeting*, 1983.
- [7] A. A. Lambregts, "Vertical Flight Path and Speed Control Autopilot Design using Total Energy Principles," in *Guidance and Control Conference*, 1983.
- [8] A. A. Lambregts, "Flight Envelope Protection for Automatic and Augmented Manual Control," in *EuroGNC*, 2013.
- [9] A. A. Lambregts, "TECS Generalized Airplane Control System Design - An Update," in *EuroGNC*, 2013.
- [10] H. K. Khalil, *Nonlinear Systems*, 3 ed., Pearson, 2001.
- [11] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, New Jersey: Prentice Hall, 1991.

- [12] G. Looye and H.-D. Joos, "Design of Robust Dynamic Inversion Control Laws using Multi-Objective Optimization," in *Guidance, Navigation and Control Conference and Exhibit*, 2001.
- [13] G. Looye, H.-D. Joos and D. Willemsen, "Application of an Optimization-based Design Process for Robust Autoland Control Laws," in *Guidance, Navigation and Control Conference and Exhibit*, 2001.
- [14] D. J. Bugajski and D. F. Enns, "Nonlinear Control Law with Application to High Angle-of-Attack," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 761-767, 1992.
- [15] S. A. Snell, D. F. Enns and W. L. Garrard Jr, "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, p. 976–984, 1992.
- [16] F. Holzapfel, *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*, Technische Universität München, 2004.
- [17] E. N. Johnson, Limited Authority Adaptive Flight Control, Georgia Institute of Technology, Dissertation, 2000.
- [18] E. N. Johnson and A. J. Calise, "Limited Authority Adaptive Flight Control for Reusable Launch Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 3, p. 906–913, 2003.
- [19] F. Holzapfel, F. Schuck, L. Höcht and G. Sachs, "Flight Dynamics Aspects of Path Control," in *AIAA Guidance, Navigation and Control Conference and Exhibition*, 2007.
- [20] J. H. Blakelock, *Automatic Control of Aircraft and Missiles*, New York: John Wiley and Sons, 1991.
- [21] I. K. Peddle, *Acceleration Based Manoeuvre Flight Control System for Unmanned Aerial Vehicles*, Stellenbosch University, 2008.
- [22] D. P. Boyle and G. E. Chamitoff, "Autonomous Maneuver Tracking for Self-Piloted Vehicles," *Journal of Guidance, Control and Dynamics*, vol. 22, no. 1, pp. 58-68, 1999.
- [23] K. Schreiter, S. Müller, R. Luckner and M. Dietrich, "A Flight Simulator Study of an Energy Control System for Manual Flight," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 6, pp. 672-683, 2019.
- [24] European Commission, *Commission Regulation No 216/2008*, 2008.
- [25] Federal Aviation Administration, *FAR Part 21: Certification Procedures for Products and Parts*, 1964.

-
- [26] Federal Aviation Administration, *FAR Part 23: Airworthiness Standards: Normal-Category Airplanes (Amdt. 23–64)*, 2016.
- [27] Federal Aviation Administration, *FAR Part 25: Transport Category Airplanes*, 1964.
- [28] European Aviation Safety Agency, *CS-23 Normal-Category Aeroplanes (Amendment 6)*, 2023.
- [29] European Commision, *Commission Regulation No 748/2012*, 2012.
- [30] Federal Aviation Administration, *Special Conditions: Airbus, Model A350-900 Series Airplane; Electronic Flight Control System: Lateral-Directional and Longitudinal Stability and Low Energy Awareness*, 143 ed., vol. 79, 2014, pp. 43233-43236.
- [31] Federal Aviation Administration, *Special Conditions: Airbus Model A350-900 Series Airplane; Flight-Envelope Protection (Icing and Non-Icing Conditions); High-Incidence Protection and Alpha-Floor Systems*, 214 ed., vol. 79, 2014, pp. 65562-65571.
- [32] European Aviation Safety Agency, *EASA.A.151 Type-Certificate Data Sheet AIRBUS A350*, 25 ed., 2022.
- [33] Federal Aviation Administration, *TSO-C198: Automatic Flight Guidance and Control System (AFGCS) Equipment*, 2011.
- [34] European Aviation Safety Agency, *ETSO-C198 Automatic Flight Guidance and Control System (AFGCS) Equipment*, 2013.
- [35] RTCA Inc., *DO-325: Minimum Operation Performance Standards (MOPS) for Automatic Flight Guidance and Control Systems and Equipment*, 2010.
- [36] Federal Aviation Administration, *AC 23-17C Systems and Equipment Guide for Certification of Part 23 Airplanes and Airships*, 2011.
- [37] SAE International, *ARP4754B Guidelines for Development of Civil Aircraft and Systems*, 2023.
- [38] SAE International, *ARP4761A: Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*, 2023.
- [39] RTCA Inc., *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*, 2011.
- [40] RTCA Inc., *DO-254: Design Assurance Guidance For Airborne Electronic Hardware*, 2000.
- [41] RTCA Inc., *DO-160G: Environmental Conditions and Test Procedures for Airborne Equipment*, 2010.

- [42] RTCA Inc., *DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, 2005.
- [43] SAE International, *ARP5150A Safety Assessment of Transport Airplanes in Commercial Service*, 2019.
- [44] SAE International, *ARP5151A Safety Assessment of General Aviation Airplanes and Rotorcraft in Commercial Service*.
- [45] RTCA Inc., *DO-335: Guidance and Installation of Automatic Flight Guidance and Control Systems (AFGCS) for Part 23 Airplanes*, 2012.
- [46] SAE International, *ARP419B Automatic Pilot Installations*, 2020.
- [47] SAE International, *ARP5366A Autopilot, Flight Director and Autothrust Systems*, 2008.
- [48] RTCA Inc., *DO-330: Software Tool Qualification Considerations*, 2011.
- [49] RTCA Inc., *DO-331: Model-Based Development and Verification Supplement to DO-178C and DO-278A*, 2011.
- [50] RTCA Inc., *DO-332: Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A*, 2011.
- [51] RTCA Inc., *DO-333: Formal Methods Supplement to DO-178C and DO-278A*, 2011.
- [52] L. Rierson, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*, Boca Raton, Florida: CRC Press, 2013.
- [53] National Transportation Safety Board, *Aircraft Accident Report NTSB/AAR-14/01: Descent Below Visual Glidepath and Impact With Seawall, Asiana Airlines Flight 214*, 2014.
- [54] The Dutch Safety Board, *Accident Report: Crashed During Approach, Boeing 737, near Amsterdam Schipol Airport, 25 February 2009*, 2010.
- [55] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*, The MIT Press, 2012.
- [56] Federal Aviation Administration, *AC 23.1311-1C Installation of Electronic Display in Part 23 Airplanes*, 2012.
- [57] SAE International, *ARP4102 Flight Deck Panels, Controls, and Displays*, 2007.
- [58] SAE International, *AS8008A Flight Director Equipment*, 2020.
- [59] SAE International, *AS420B Flight Directors (Reciprocating Engine Powered Aircraft)*, 2008.

-
- [60] E. T. Evans, L. J. Kramer, T. J. Etherington, T. S. Daniels, S. D. Young, Y. Santiago-Espada and J. R. Barnes, in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019.
- [61] T. M. Adami, E. Theunissen, D. Sizoo and R. McGuire, "An Energy Management Display for General Aviation Safety Enhancements," in *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, 2014.
- [62] T. Lambregts, R. Rademaker and E. Theunissen, "A New Ecological Primary Flight Display Concept," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, 2008.
- [63] ISO/IEC/IEEE, *ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes*, 2015.
- [64] ISO/IEC/IEEE, *ISO/IEC/IEEE 24748-4:2016 Systems and software engineering — Life cycle management — Part 4: Systems engineering planning*, 2016.
- [65] ISO/IEC/IEEE, *ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering*, 2018.
- [66] ISO/IEC/IEEE, *ISO/IEC/IEEE 42010:2022 Software, systems and enterprise — Architecture description*, 2022.
- [67] ISO/IEC/IEEE, *ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes*, 2017.
- [68] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4 ed., Hoboken, NJ: John Wiley and Sons, Inc, 2015.
- [69] Airlines for America, *ATA iSpec 2200: Information Standards for Aviation Maintenance*, 2021.
- [70] S1000D Steering Committee, *S1000D - International Specification for Technical Publications Using a Common Source Database*, 2019.
- [71] E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiege, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Automatic Flight Path Control of an Experimental DA42 General Aviation Aircraft," in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.
- [72] E. Karlsson, A. Gabrys, S. P. Schatz and F. Holzapfel, "Dynamic Flight Path Control Coupling for Energy and Maneuvering Integrity," in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.

- [73] E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. C. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Development of an Automatic Flight Path Controller for a DA42 General Aviation Aircraft," in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017*, Cham, Springer, 2017, pp. 121-139.
- [74] E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. C. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Active Control Objective Prioritization for High-Bandwidth Automatic Flight Path Control," in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017*, Cham, Springer, 2017, pp. 141-161.
- [75] S. P. Schatz and F. Holzapfel, "Modular Trajectory / Path Following Controller Using Nonlinear Error Dynamics," in *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, 2014.
- [76] S. P. Schatz and F. Holzapfel, "Nonlinear Modular 3D Trajectory Control of a General Aviation Aircraft," in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017*, Cham, Springer, 2017, pp. 163-183.
- [77] V. Schneider, P. S. S. P. Piprek, T. Baier, C. Dörhöfer, M. Hochstrasser, A. Gabrys, E. Karlsson, C. Krause, P. J. Lauffs, N. C. Mumm, K. Nürnberger, L. Peter, P. Spiegel, L. Steinert, A. Zollitsch and F. Holzapfel, "Online Trajectory Generation Using Clothoid Segments," in *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, 2016.
- [78] V. Schneider and F. Holzapfel, "Modular Trajectory Generation Test Platform for Real Flight Systems," in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017*, Cham, Springer, 2017, pp. 185-202.
- [79] C. Krause and F. Holzapfel, "System Automation of a DA42 General Aviation Aircraft," in *2018 Aviation Technology, Integration, and Operations Conference*, Atlanta, Georgia, 2018.
- [80] C. Krause and F. Holzapfel, "Designing a System Automation for a Novel UAV Demonstrator," in *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, 2016.

-
- [81] M. E. Kügler and F. Holzapfel, "Autoland for a Novel UAV as a State-Machine-based Extension to a Modular Automatic Flight Guidance and Control System," in *2017 American Control Conference*, Seattle, USA, 2017.
- [82] M. E. Kügler and F. Holzapfel, "Designing a Safe and Robust Automatic Take-off Maneuver for a Fixed-Wing UAV," in *14th International Conference on Control, Automation, Robotics & Vision*, Phuket, Thailand, 2016.
- [83] A. W. Zollitsch, N. C. Mumm, S. Wulf, F. Holzapfel, M. Hochstrasser, P. J. Lauffs and L. Peter, "Automatic Takeoff of a General Aviation Research Aircraft," in *11th Asian Control Conference (ASCC)*, Gold Coast, QLD, Australia, 2017.
- [84] N. C. Mumm and F. Holzapfel, "Development of an Automatic Landing System for Diamond DA 42 Aircraft Utilizing a Load Factor Inner Loop Command System," in *Euro GNC 2017 - 4th CEAS Specialist Conference on Guidance, Navigation & Control*, Warsaw, Poland, 2017.
- [85] S. P. Schatz, A. C. Gabrys, D. M. Gierszewski and F. Holzapfel, "Inner Loop Command Interface in a Modular Flight Control Architecture for Trajectory Flights of General Aviation Aircraft," in *5th International Conference on Control, Decision and Information Technologies*, Thessaloniki, Greece, 2018.
- [86] S. Braun, M. Geiser, M. Heller and F. Holzapfel, "Configuration Assessment and Preliminary Control Law Design for a Novel Diamond-Shaped UAV," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, 2014.
- [87] A. W. Zollitsch, S. P. Schatz, N. C. Mumm and F. Holzapfel, "Model-in-the-Loop Simulation of Experimental Flight Control Software," in *2018 AIAA Modeling and Simulation Technologies Conference*, Kissimmee, Florida, 2018.
- [88] P. J. Lauffs, M. Hochstrasser and F. Holzapfel, "Real-time Simulation of Nonlinear Transmission," in *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, 2014.
- [89] K. Nürnberger, M. Hochstrasser and F. Holzapfel, "Execution time analysis and optimisation techniques in the model-based development of a flight control software," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 2, pp. 57-64, 2017.
- [90] L. Peter, "DA42 MNG FBW Research Aircraft," in *In-Flight Simulators and Fly-by-Wire/Light Demonstrators*, Cham, Springer, 2017, pp. 146-148.
- [91] M. Kreienfeld, K. Giese, J. Heider and S. Kaltenhäuser, "Development of a RPV-Demonstrator for ATM Research," in *SCI-269 Symposium on Flight Testing of Unmanned Aerial Systems (UAS)*, Ottawa, 2015.

- [92] M. E. Kügler and F. Holzapfel, "Planning, Implementation, and Execution of an Automatic First Flight of a UAV," in *31st Congress of the International Council of the Aeronautical Sciences (ICAS)*, Belo Horizonte, Brazil, 2018.
- [93] M. E. Kügler and F. Holzapfel, "Designing Autonomous Flight Capabilities for UAV and OPV – A Comparison of Development and Flight Test Experiences," in *AIAA Aviation 2020 Forum*, 2020.
- [94] M. Hochstrasser, S. P. Schatz, K. Nürnberger, S. Myschik and F. Holzapfel, "Aspects of a Consistent Modeling Environment for DO-331 Design Model Development of Flight Control Algorithms," in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Fourth CEAS Specialist Conference on Guidance, Navigation and Control Held in Warsaw, Poland, April 2017*, Cham, Springer, 2017, pp. 69-86.
- [95] M. Hochstrasser, S. Myschik and F. Holzapfel, "Application of a Process-Oriented Build Tool for Flight Controller Development Along a DO-178C/DO-331 Process," in *Model-Driven Engineering and Software Development: 6th International Conference, MODELSWARD 2018, Funchal, Madeira, Portugal, January 22–24, 2018, Revised Selected Papers*, Cham, Springer, 2019, pp. 380-405.
- [96] Deutsches Zentrum für Luft- und Raumfahrt, "DO 228-101 (D-CODE)," [Online]. Available: <https://www.dlr.de/content/de/grossforschungsanlagen/dornier-228-101.html>. [Accessed 14 April 2023].
- [97] Deutsches Zentrum für Luft- und Raumfahrt, "Dornier DO 228-101 D-CODE," [Online]. Available: <https://www.dlr.de/content/en/articles/aeronautics/research-fleet-infrastructure/dlr-research-aircraft/dornier-do-228-101-d-code.html>. [Accessed 14 April 2023].
- [98] M. Hochstrasser, *Modular model-based development of safety-critical flight control software*, Dissertation, Technische Universität München, 2020.
- [99] F. Holzapfel, F. Schuck, L. Höcht, F. Kurth and G. Sachs, "Non-Linear High Bandwidth Control of UAVs for Autonomous Mission Capability," in *45th Aerospace Sciences Meeting and Exhibition*, 2007.
- [100] F. Holzapfel, L. Höcht, F. Schuck, S. Myschik and G. Sachs, "Nonlinear Flight-Path Control – A Flight Dynamics Perspective," in *Jahrbuch 2007 der Deutschen Gesellschaft für Luft- und Raumfahrt* 6, 2008, pp. 3627-3640.
- [101] R. L. Flood and E. R. Carson, *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, New York, NY: Plenum Press, 1993.
- [102] K. Schmiechen, S. A. Zafar, K. Dmitriev, C. Krammer, M. Maly and F. Holzapfel, "A Requirements Management Template in Polarion for Model-Based Development of

Airborne Systems," in *Software Engineering 2021 Satellite Events - Workshops and Tools & Demos*, Braunschweig/Virtual, Germany, 2021.

- [103] SAE International, *AS94900A Aerospace - Flight Control Systems - Design, Installation and Test of Piloted Military Aircraft, General Specification For*, 2018.
- [104] Federal Aviation Administration, *AC 23.1309-1E System Safety Analysis and Assessment for Part 23 Airplanes*, 2011.
- [105] International Organization for Standardization (ISO), *Standard Atmosphere*, 1975.
- [106] Air Force Systems Command, *MIL-F-8785C: Military Specification - Flying Qualities of Piloted Airplanes*, 1980.
- [107] D. J. Moorhouse and R. J. Woodcock, *Background Information and User Guide for MIL-F-8785C Military Specification - Flying Qualities of Piloted Vehicles*, Flight Dynamics Laboratory, Air Force Systems Command, 1982.
- [108] Department of Defense, *MIL-HDBK-1797: Flying Qualities of Piloted Aircraft*, 1997.
- [109] IEEE Aerospace Electronics Systems Society, Ed., *IEEE standard specification format guide and test procedure for single-axis laser gyros*, New York, N.Y., USA: Institute of Electrical and Electronics Engineers, 2008.
- [110] R. T. Bluth, P. A. Durkee, J. H. Seinfeld, R. C. Flagan, L. M. Russell, P. A. Crowley and P. Finn, "Center for Interdisciplinary Remotely-Piloted Aircraft Studies (CIRPAS)," *Bulletin of the American Meteorological Society*, vol. 77, no. 11, pp. 2691-2699, 1996.
- [111] S. P. Schatz, V. Schneider, E. Karlsson, T. Baier, C. Dörhöfer, M. Hochstrasser, A. Gabrys, K. Christoph, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Flightplan Flight Tests of an Experimental DA42 General Aviation Aircraft," in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.
- [112] J. W. Rice and R. D. McCorkle, "Digital Flight Control Reliability - Effects of Redundancy Level, Architecture and Redundancy Management Technique," in *AIAA Guidance and Control Conference*, Boulder, Colorado, 1979.
- [113] European Aviation Safety Agency, *Notice of Proposed Amendment 2016-05: Reorganisation of CS-23*, 2016.

A Scientific Publications

During the period at the Institute of Flight System Dynamics at TUM, the following scientific publications have been published by the author. Besides, the author contributed to the notes for the lecture *Development of Flight Control Systems* and to the notes for the lab course *Flight Control Systems Lab*.

- E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Development of an Automatic Flight Path Controller for a DA42 General Aviation Aircraft" in *Dołęga, B., Głębocki, R., Kordos, D., Żugaj, M. (eds) Advances in Aerospace Guidance, Navigation and Control*, pp. 121-139, Springer, Cham, 2018
- E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Active Control Objective Prioritization for High-Bandwidth Automatic Flight Path Control" in *Dołęga, B., Głębocki, R., Kordos, D., Żugaj, M. (eds) Advances in Aerospace Guidance, Navigation and Control*, pp. 141-161, Springer, Cham, 2018
- E. Karlsson, S. P. Schatz, T. Baier, C. Dörhöfer, A. Gabrys, M. Hochstrasser, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, V. Schneider, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Automatic Flight Path Control of an Experimental DA42 General Aviation Aircraft" in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016
- E. Karlsson, A. Gabrys, S. P. Schatz and F. Holzapfel, "Dynamic Flight Path Control Coupling for Energy and Maneuvering Integrity" in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016
- S. P. Schatz, V. Schneider, E. Karlsson, T. Baier, C. Dörhöfer, M. Hochstrasser, A. Gabrys, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, P. Spiegel, L. Steinert, A. W. Zollitsch and F. Holzapfel, "Flightplan Flight Tests of an Experimental

DA42 General Aviation Aircraft" in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016

- V. Schneider, P. Piprek, S. P. Schatz, T. Baier, C. Dörhöfer, M. Hochstrasser, A. Gabrys, E. Karlsson, C. Krause, P. J. Lauffs, N. Mumm, K. Nürnberger, L. Peter, P. Spiegel, L. Steinert and F. Holzapfel, "Online Trajectory Generation Using Clothoid Segments" in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016
- E. Karlsson, S. Myschik and F. Holzapfel, "Eigenstructure Assignment and Robustness Improvement Using a Gradient-Based Method" in *Holzapfel, F., Theil, S. (eds) Advances in Aerospace Guidance, Navigation and Control*, pp. 67-77, Springer, Berlin, Heidelberg, 2011

B Coordinate Systems and Transformations

This appendix lists the coordinate systems used throughout the thesis, and the sequences of rotation, transformation matrices and angular velocities between those systems.

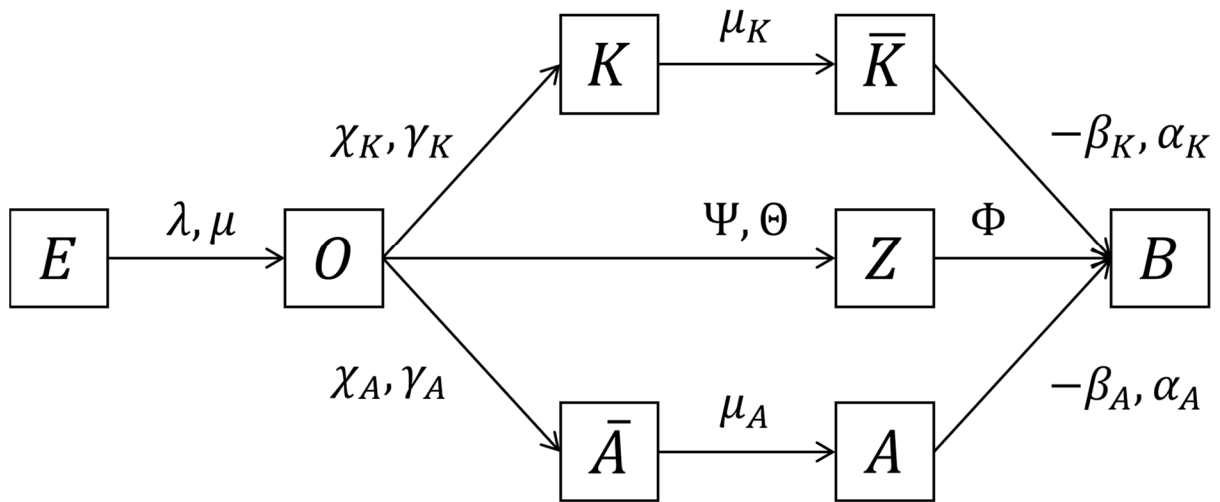


Figure B.1: Coordinate systems and rotation angles.

B.1 Coordinate Systems

Earth-Centered Inertial (ECI) System

Index:	I
Role:	Notation frame for Newtonian Inertial Physics (i.e., valid Euclidean Frame)
Origin:	Center of the Earth
Translation:	Around the Sun with the solar system
Rotation:	None
x -axis:	In equatorial plane, pointing towards vernal equinox
y -axis:	In equatorial plane to form a right-hand system
z -axis:	Rotation axis of the Earth

Earth-Centered Earth-Fixed (ECEF) System

Index:	E
Role:	Notation frame for positioning and navigation
Origin:	Center of the Earth
Translation:	Moves with ECI-Frame
Rotation:	Earth rotation about z -axis with the angular rate of the Earth
x -axis:	In equatorial plane, pointing through Greenwich meridian
y -axis:	In equatorial plane to form a right-hand system
z -axis:	Rotation axis of the Earth

North-East Down (NED) System

Index:	O
Role:	Notation frame for velocity and orientation
Origin:	Reference point of aircraft
Translation:	Moves with aircraft reference point
Rotation:	Rotates with transport rate to keep the NED-alignment
x -axis:	Parallel to local geoid surface, pointing to geographic north pole
y -axis:	Parallel to local geoid surface, pointing east to form a right-hand system
z -axis:	Pointing downwards, perpendicular to local geoid surface

Kinematic Flight Path System

Index:	K
Role:	Notation frame for flight path
Origin:	Reference point of aircraft
Translation:	Moves with aircraft reference point
Rotation:	Rotates with direction of kinematic aircraft motion
x -axis:	Aligned with the kinematic velocity, pointing into the direction of the kinematic velocity
y -axis:	Pointing to the right, perpendicular to the x - und z -axes
z -axis:	Pointing downwards, parallel to the projection of the local surface normal of the WGS-84 ellipsoid into a plane perpendicular to the x -axis (i.e., perpendicular to the kinematic velocity)

Intermediate Kinematic Flight Path System

Index:	\bar{K}
Role:	Notation frame for flight path
Origin:	Reference point of aircraft
Translation:	Moves with aircraft reference point
Rotation:	Rotates with direction of kinematic aircraft motion
x -axis:	Aligned with the kinematic velocity, pointing into the direction of the kinematic velocity
y -axis:	Pointing to the right, perpendicular to the x - und z -axes
z -axis:	z -axis of the Kinematic Flight Path Frame K rotated clockwise by the kinematic flightpath bank angle μ_K

Aerodynamic System

Index:	A
Role:	Notation frame for aerodynamic flow
Origin:	Aerodynamic reference point of aircraft
Translation:	Moves with aircraft aerodynamic reference point
Rotation:	Rotates with direction of airflow
x -axis:	Aligned with aerodynamic velocity, pointing into the direction of the aerodynamic velocity
y -axis:	Pointing to the right, perpendicular to the x - und z -axes
z -axis:	Pointing downwards in the symmetry plane of the aircraft, perpendicular to the x -axis

Intermediate Aerodynamic System

Index:	\bar{A}
Role:	Notation frame for aerodynamic flow
Origin:	Aerodynamic reference point of aircraft
Translation:	Moves with aircraft aerodynamic reference point
Rotation:	Rotates with direction of airflow
x -axis:	Aligned with aerodynamic velocity, pointing into the direction of the aerodynamic velocity
y -axis:	Pointing to the right, perpendicular to the x - und z -axes
z -axis:	z -axis of the Aerodynamic Frame A rotated counterclockwise by the aerodynamic flightpath bank angle μ_A

Body-Fixed System

Index:	B
Role:	Notation frame for aircraft attitude and rotation
Origin:	Reference point of aircraft
Translation:	Moves with aircraft reference point
Rotation:	Rotates with rigid body aircraft
x -axis:	Pointing towards aircraft nose in symmetry plane
y -axis:	Pointing to right (starboard) wing to form an orthogonal right-hand system
z -axis:	Pointing downwards in symmetry plane, perpendicular to x - and y -axes

Intermediate Body-Fixed System

Index:	Z
Role:	Notation frame for aircraft attitude and rotation
Origin:	Reference point of aircraft
Translation:	Moves with aircraft reference point
Rotation:	Rotates with rigid body aircraft
x -axis:	Pointing towards aircraft nose in symmetry plane
y -axis:	Parallel to local geoid surface
z -axis:	Pointing downwards in symmetry plane, perpendicular to x - and y -axes

B.2 Transformation Matrices and Angular Rates

ECEF to NED Frame

Angles: Geodetic longitude λ
Geodetic latitude μ

Rotation sequence: $\lambda \rightarrow \mu$

Transformation matrix:

$$\mathbf{M}_{OE} = \begin{bmatrix} -\sin \mu \cos \lambda & -\sin \mu \sin \lambda & \cos \mu \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \mu \cos \lambda & -\cos \mu \sin \lambda & \sin \mu \end{bmatrix} \quad (\text{D.1})$$

Angular velocities:

$$(\overline{\boldsymbol{\omega}}^{EO})_E = \begin{bmatrix} \dot{\mu} \sin \lambda \\ -\dot{\mu} \cos \lambda \\ \dot{\lambda} \end{bmatrix}. \quad (\text{D.2})$$

NED to Kinematic Frame

Angles: Kinematic Flight Path Azimuth (Course) Angle χ_K
Kinematic Flight Path Inclination (Climb) Angle γ_K

Rotation sequence: $\chi_K \rightarrow \gamma_K$

Transformation matrix:

$$\mathbf{M}_{KO} = \begin{bmatrix} \cos \chi_K \cos \gamma_K & \sin \chi_K \cos \gamma_K & -\sin \gamma_K \\ -\sin \chi_K & \cos \chi_K & 0 \\ \cos \chi_K \sin \gamma_K & \sin \chi_K \sin \gamma_K & \cos \gamma_K \end{bmatrix} \quad (\text{D.3})$$

Angular velocities:

$$(\overline{\boldsymbol{\omega}}^{OK})_O = \begin{bmatrix} -\dot{\gamma}_K \sin \chi_K \\ \dot{\gamma}_K \cos \chi_K \\ \dot{\chi}_K \end{bmatrix}. \quad (\text{D.4})$$

NED to Intermediate Kinematic Frame

Angles: Kinematic Flight Path Azimuth (Course) Angle χ_K
Kinematic Flight Path Inclination (Climb) Angle γ_K
Kinematic Flight Path Bank Angle μ_K

Rotation sequence: $\chi_K \rightarrow \gamma_K \rightarrow \mu_K$

Transformation matrix:

$$\mathbf{M}_{\bar{K}O} = \begin{bmatrix} \cos \chi_K \cos \gamma_K & \sin \chi_K \cos \gamma_K & -\sin \gamma_K \\ \cos \chi_K \sin \gamma_K \sin \mu_K & \sin \chi_K \sin \gamma_K \sin \mu_K & \cos \gamma_K \sin \mu_K \\ -\sin \chi_K \cos \mu_K & +\cos \chi_K \cos \mu_K & \\ \cos \chi_K \sin \gamma_K & -\sin \chi_K \sin \gamma_K & \cos \gamma_K \end{bmatrix} \quad (D.5)$$

Angular velocities:

$$(\vec{\omega}^{O\bar{K}})_O = \begin{bmatrix} \dot{\mu}_K \cos \chi_K \cos \gamma_K - \dot{\gamma}_K \sin \chi_K \\ \dot{\mu}_K \sin \chi_K \sin \gamma_K + \dot{\gamma}_K \cos \chi_K \\ \dot{\chi}_K - \dot{\mu}_K \sin \gamma_K \end{bmatrix}. \quad (D.6)$$

NED to Aerodynamic Frame

Angles: Aerodynamic Flight Path Azimuth (Course) Angle χ_A
Aerodynamic Flight Path Inclination (Climb) Angle γ_A
Aerodynamic Flight Path Bank Angle μ_A

Rotation sequence: $\chi_A \rightarrow \gamma_A \rightarrow \mu_A$

Transformation matrix:

$$\mathbf{M}_{AO} = \begin{bmatrix} \cos \chi_A \cos \gamma_A & \sin \chi_A \cos \gamma_A & -\sin \gamma_A \\ \cos \chi_A \sin \gamma_A \sin \mu_A & \sin \chi_A \sin \gamma_A \sin \mu_A & \cos \gamma_A \sin \mu_A \\ -\sin \chi_A \cos \mu_A & +\cos \chi_A \cos \mu_A & \\ \cos \chi_A \sin \gamma_A & -\sin \chi_A \sin \gamma_A & \cos \gamma_A \end{bmatrix} \quad (D.7)$$

Angular velocities:

$$(\vec{\omega}^{OA})_O = \begin{bmatrix} \dot{\mu}_A \cos \chi_A \cos \gamma_A - \dot{\gamma}_A \sin \chi_A \\ \dot{\mu}_A \sin \chi_A \sin \gamma_A + \dot{\gamma}_A \cos \chi_A \\ \dot{\chi}_A - \dot{\mu}_A \sin \gamma_A \end{bmatrix}. \quad (D.8)$$

NED to Intermediate Aerodynamic Frame

Angles: Aerodynamic Flight Path Azimuth (Course) Angle χ_A
Aerodynamic Flight Path Inclination (Climb) Angle γ_A

Rotation sequence: $\chi_A \rightarrow \gamma_A$

Transformation matrix:

$$\mathbf{M}_{\bar{A}O} = \begin{bmatrix} \cos \chi_A \cos \gamma_A & \sin \chi_A \cos \gamma_A & -\sin \gamma_A \\ -\sin \chi_A & \cos \chi_A & 0 \\ \cos \chi_A \sin \gamma_A & \sin \chi_A \sin \gamma_A & \cos \gamma_A \end{bmatrix} \quad (D.9)$$

Angular velocities:

$$(\vec{\omega}^{OA})_O = \begin{bmatrix} -\dot{\gamma}_A \sin \chi_A \\ -\dot{\gamma}_A \cos \chi_A \\ \dot{\chi}_A \end{bmatrix}. \quad (D.10)$$

NED to Body-Fixed Frame

Angles: Azimuth (Heading) Angle Ψ
 Inclination (Pitch) Angle Θ
 Bank Angle Φ

Rotation sequence: $\Psi \rightarrow \Theta \rightarrow \Phi$

Transformation matrix:

$$\mathbf{M}_{BO} = \begin{bmatrix} \cos \Psi \cos \Theta & \sin \Psi \cos \Theta & -\sin \Theta \\ \cos \Psi \sin \Theta \sin \Phi & \sin \Psi \sin \Theta \sin \Phi & \cos \Theta \sin \Phi \\ -\sin \Psi \cos \Phi & +\cos \Psi \cos \Phi & \\ \cos \Psi \sin \Theta \cos \Phi & \sin \Psi \sin \Theta \cos \Phi & \cos \Theta \cos \Phi \\ +\sin \Psi \sin \Phi & -\cos \Psi \sin \Phi & \end{bmatrix} \quad (D.11)$$

Angular velocities:

$$(\vec{\omega}^{OB})_B = \begin{bmatrix} \dot{\Phi} \cos \Theta \cos \Psi + \dot{\Theta} \sin \Psi \\ \dot{\Phi} \cos \Theta \sin \Psi + \dot{\Theta} \cos \Psi \\ \dot{\Psi} - \dot{\Phi} \sin \Theta \end{bmatrix} \quad (D.12)$$

NED to Intermediate Body-Fixed Frame

Angles: Azimuth (Heading) Angle Ψ
 Inclination (Pitch) Angle Θ

Rotation sequence: $\Psi \rightarrow \Theta$

Transformation matrix:

$$\mathbf{M}_{KO} = \begin{bmatrix} \cos \Psi \cos \Theta & \sin \Psi \cos \Theta & -\sin \Theta \\ -\sin \Psi & \cos \Psi & 0 \\ \cos \Psi \sin \Theta & -\sin \Psi \sin \Theta & \cos \Theta \end{bmatrix} \quad (D.13)$$

Angular velocities:

$$(\vec{\omega}^{OZ})_O = \begin{bmatrix} -\dot{\Theta} \sin \Psi \\ -\dot{\Theta} \cos \Psi \\ \dot{\Psi} \end{bmatrix}. \quad (D.14)$$

Kinematic to Body-Fixed Frame

Angles: Kinematic Flight Path Bank Angle μ_K
 Kinematic Sideslip Angle β_K
 Kinematic Angle of Attack α_K

Rotation sequence: $\mu_K \rightarrow -\beta_K \rightarrow \alpha_K$

Transformation matrix:

$$\mathbf{M}_{BK} = \begin{bmatrix} \cos \alpha_K \cos \beta_K & \sin \beta_K & \sin \alpha_K \cos \beta_K \\ -\cos \alpha_K \sin \beta_K \cos \mu_K + \sin \alpha_K \sin \mu_K & \cos \beta_K \cos \mu_K & -\sin \alpha_K \sin \beta_K \cos \mu_K - \cos \alpha_K \sin \mu_K \\ -\cos \alpha_K \sin \beta_K \sin \mu_K - \sin \alpha_K \cos \mu_K & \cos \beta_K \sin \mu_K & -\sin \alpha_K \sin \beta_K \sin \mu_K + \cos \alpha_K \cos \mu_K \end{bmatrix} \quad (\text{D.15})$$

Angular velocities:

$$(\vec{\omega}^{KB})_B = \begin{bmatrix} -\dot{\mu}_K \cos \alpha_K \cos \beta_K + \dot{\beta}_K \sin \alpha_K \\ -\dot{\alpha}_K - \dot{\mu}_K \sin \beta_K \\ \dot{\mu}_K \sin \alpha_K \cos \beta_K - \dot{\beta}_K \cos \alpha_K \end{bmatrix} \quad (\text{D.16})$$

Intermediate Aerodynamic to Body-Fixed Frame

Angles: Aerodynamic Flight Path Bank Angle μ_A
Aerodynamic Sideslip Angle β_A
Aerodynamic Angle of Attack α_A

Rotation sequence: $\mu_A \rightarrow -\beta_A \rightarrow \alpha_A$

Transformation matrix:

$$\mathbf{M}_{B\bar{A}} = \begin{bmatrix} \cos \alpha_A \cos \beta_A & \sin \beta_A & \sin \alpha_A \cos \beta_A \\ -\cos \alpha_A \sin \beta_A \cos \mu_A + \sin \alpha_A \sin \mu_A & \cos \beta_A \cos \mu_A & -\sin \alpha_A \sin \beta_A \cos \mu_A - \cos \alpha_A \sin \mu_A \\ -\cos \alpha_A \sin \beta_A \sin \mu_A - \sin \alpha_A \cos \mu_A & \cos \beta_A \sin \mu_A & -\sin \alpha_A \sin \beta_A \sin \mu_A + \cos \alpha_A \cos \mu_A \end{bmatrix} \quad (\text{D.17})$$

Angular velocities:

$$(\vec{\omega}^{\bar{A}B})_B = \begin{bmatrix} -\dot{\mu}_A \cos \alpha_A \cos \beta_A + \dot{\beta}_A \sin \alpha_A \\ -\dot{\alpha}_A - \dot{\mu}_A \sin \beta_A \\ \dot{\mu}_A \sin \alpha_A \cos \beta_A - \dot{\beta}_A \cos \alpha_A \end{bmatrix} \quad (\text{D.18})$$

Intermediate Body-Fixed to Body-Fixed Frame

Angles: Bank Angle Φ

Rotation sequence: Φ

Transformation matrix:

$$\mathbf{M}_{BZ} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \quad (\text{D.19})$$

Angular velocities:

$$(\vec{\omega}^{ZB})_Z = \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} \quad (\text{D.20})$$

C Mode Control and Monitoring Interface Specification

C.1 MCP/MCMD Command Logic

Speed Command Logic

Speed Command Logic Input/Output Overview

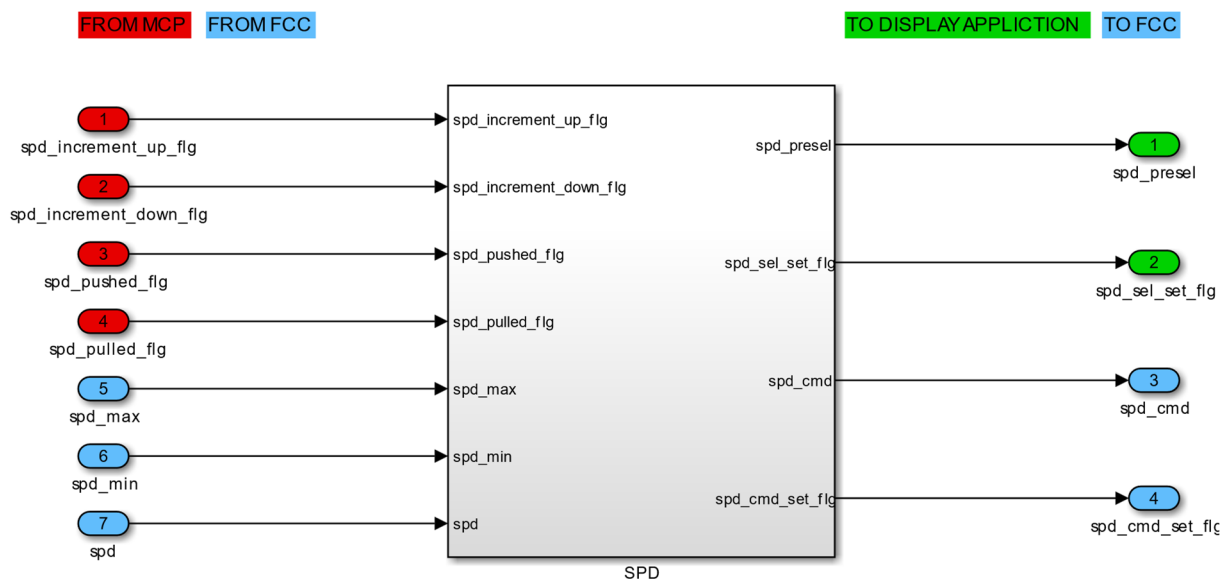


Figure C.1: Speed command logic inputs and outputs.

Speed Command Logic Structure

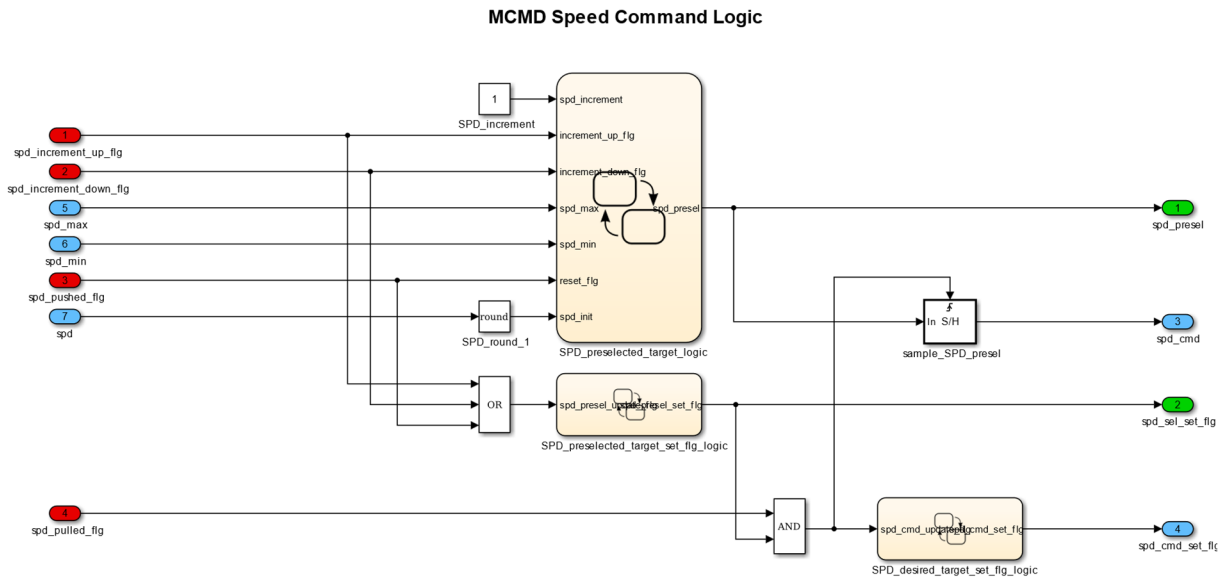


Figure C.2: Speed command logic structure.

Speed Preselect Logic

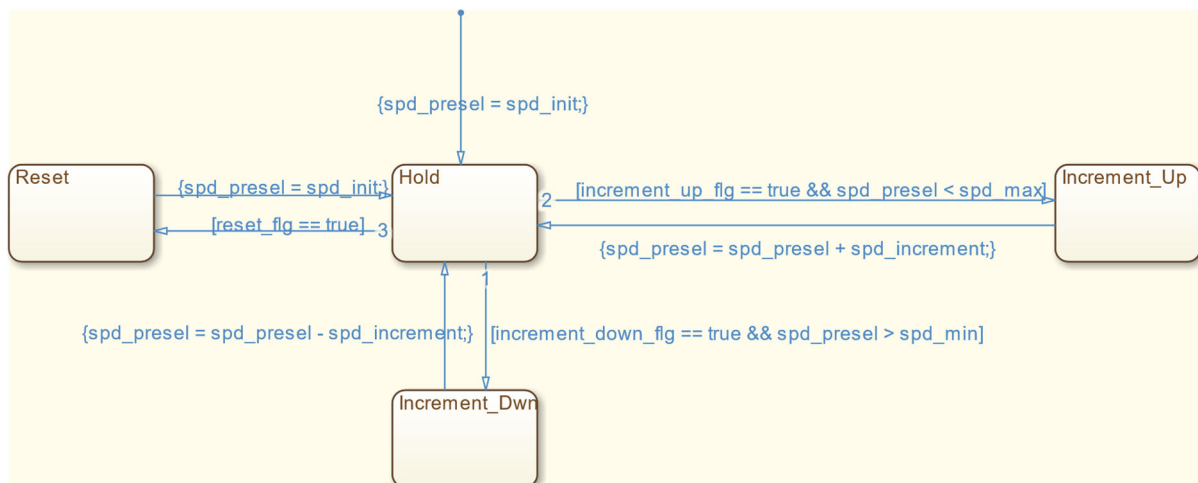


Figure C.3: Speed preselect logic.

Speed Preselect Target Set Flag Logic

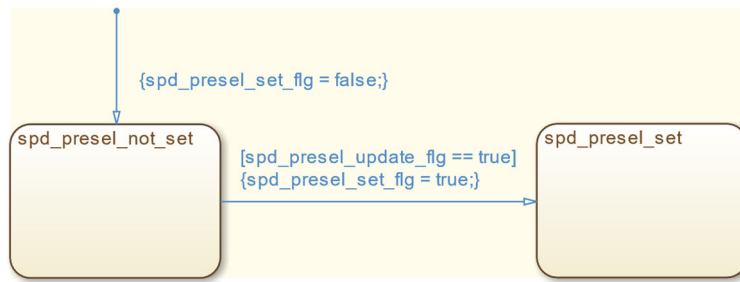


Figure C.4: Speed preselect set flag logic.

Speed Desired Target Set Flag Logic

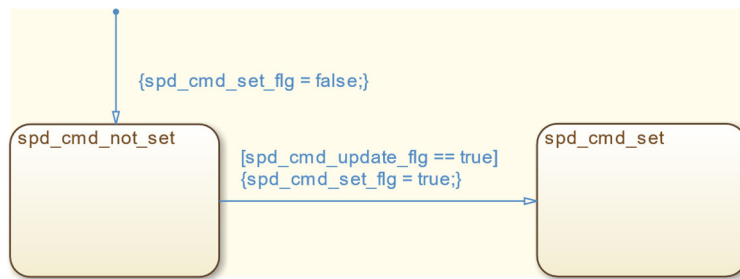


Figure C.5: Speed desired target set flag logic.

Heading/Track Command Logic

Heading/Track Command Logic Input/Output Overview

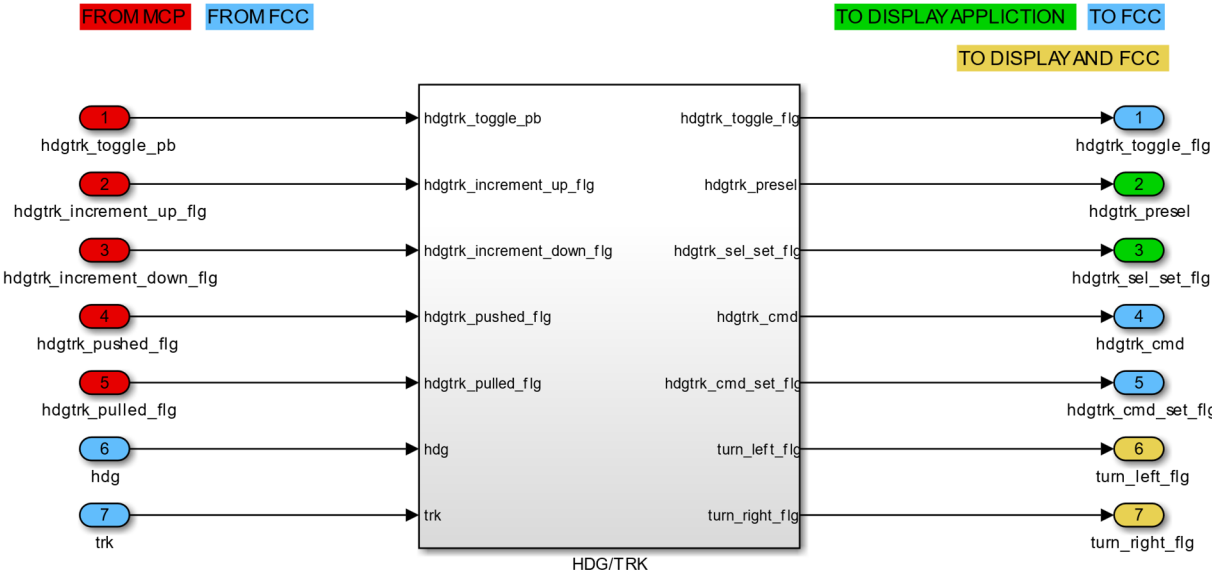


Figure C.6: Heading/Track command logic inputs and outputs.

MCMD Heading / Track Angle Command Logic

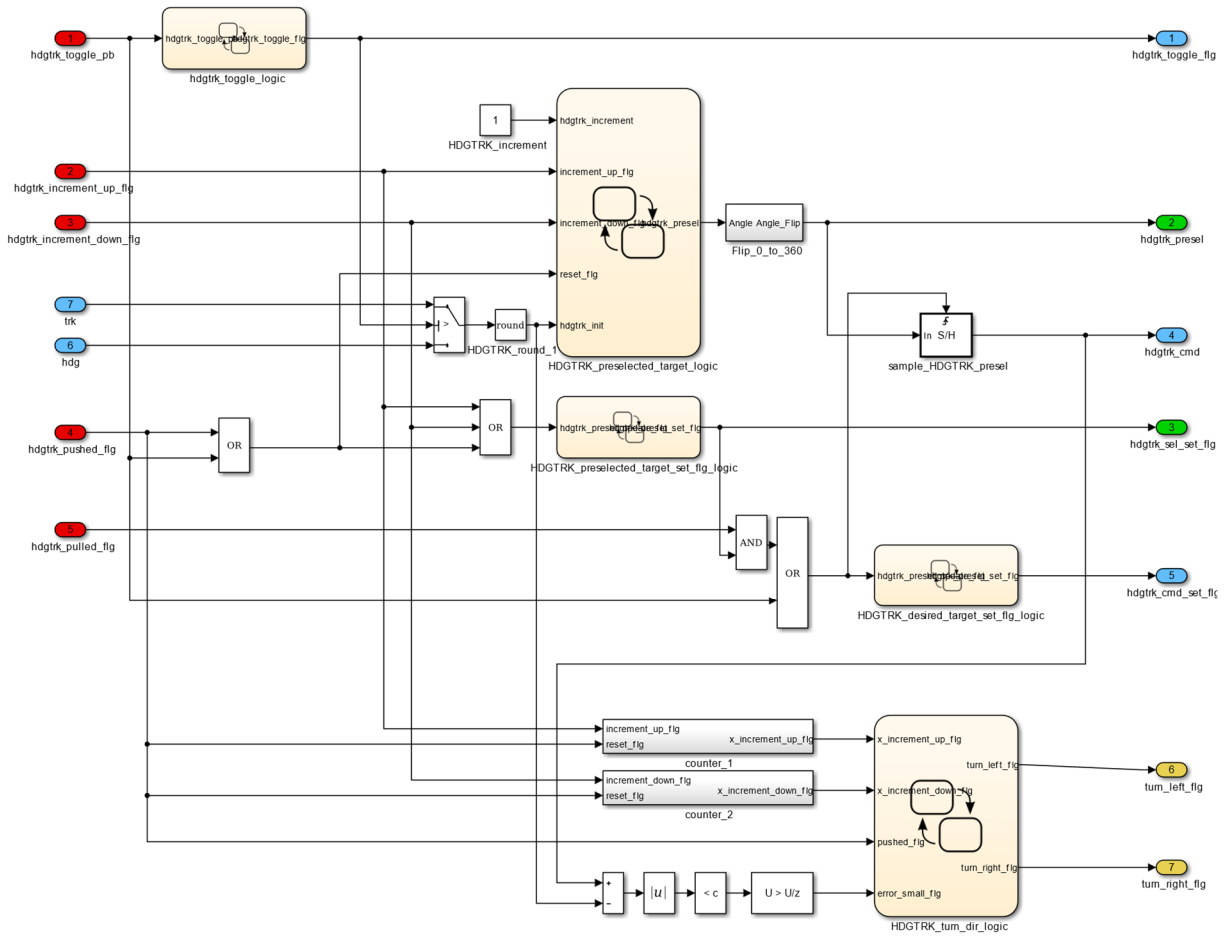


Figure C.7: Heading/Track command logic structure.

Heading/Track Toggle Logic

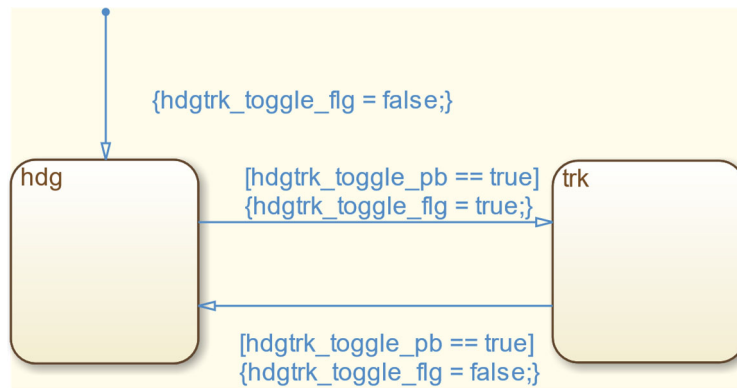


Figure C.8: Heading/Track toggle logic.

Heading/Track Preselect Logic

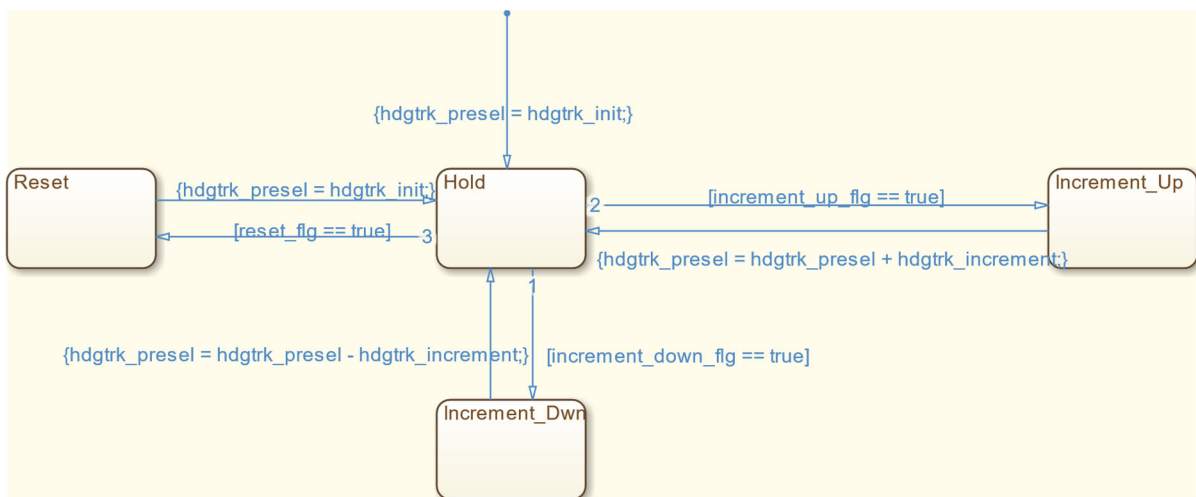


Figure C.9: Heading/Track prelect logic.

Heading/Track Preselect/Desired Target Set Flag Logic

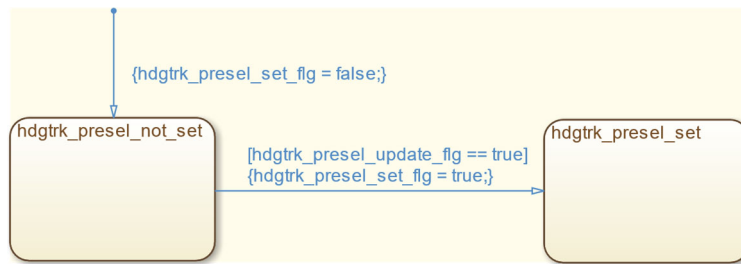


Figure C.10: Heading/Track preselect set flag logic.

Heading/Track Preselect/Desired Target Set Flag Logic

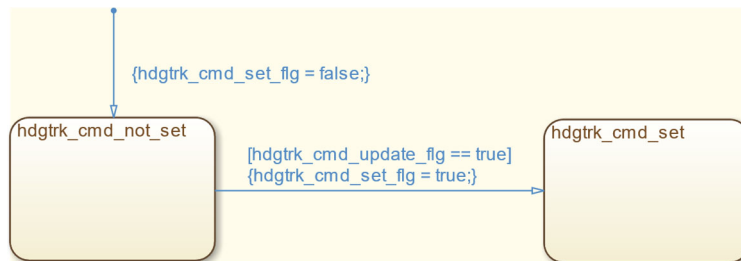


Figure C.11: Heading/Track desired target set flag logic.

Turn Right/Left Flag Logic

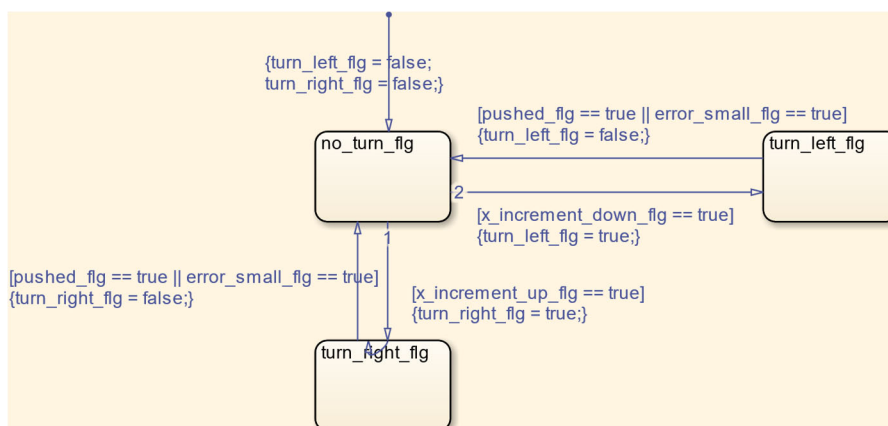


Figure C.12: Heading/Track turn left/right flag logic.

Altitude Command Logic

Altitude Command Logic Input/Output Overview

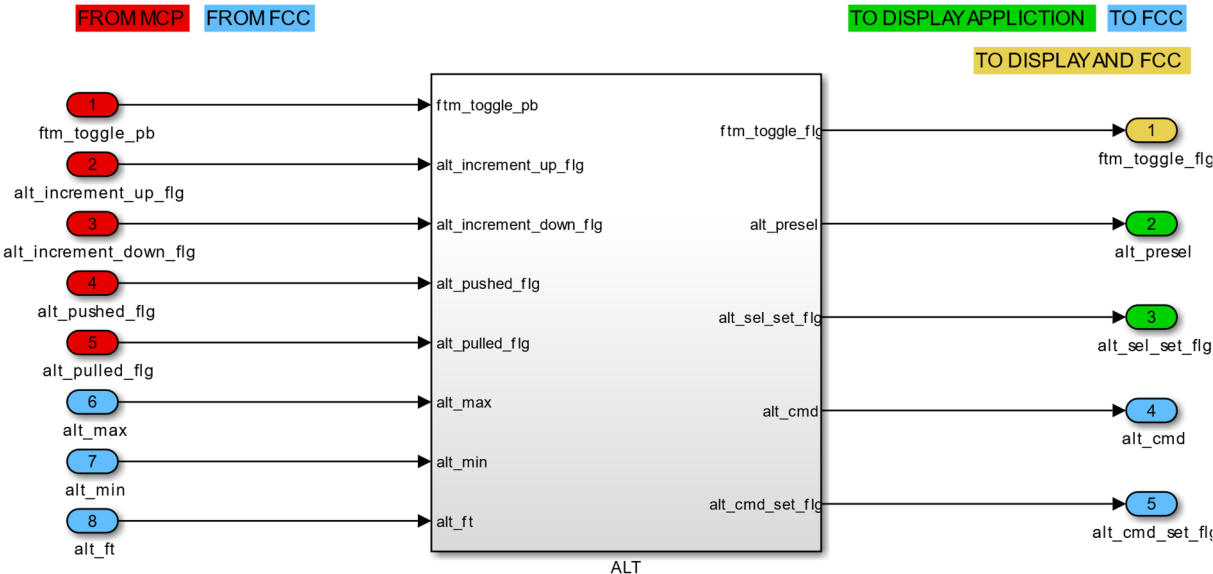


Figure C.13: Altitude command logic inputs and outputs.

Altitude Command Logic Structure

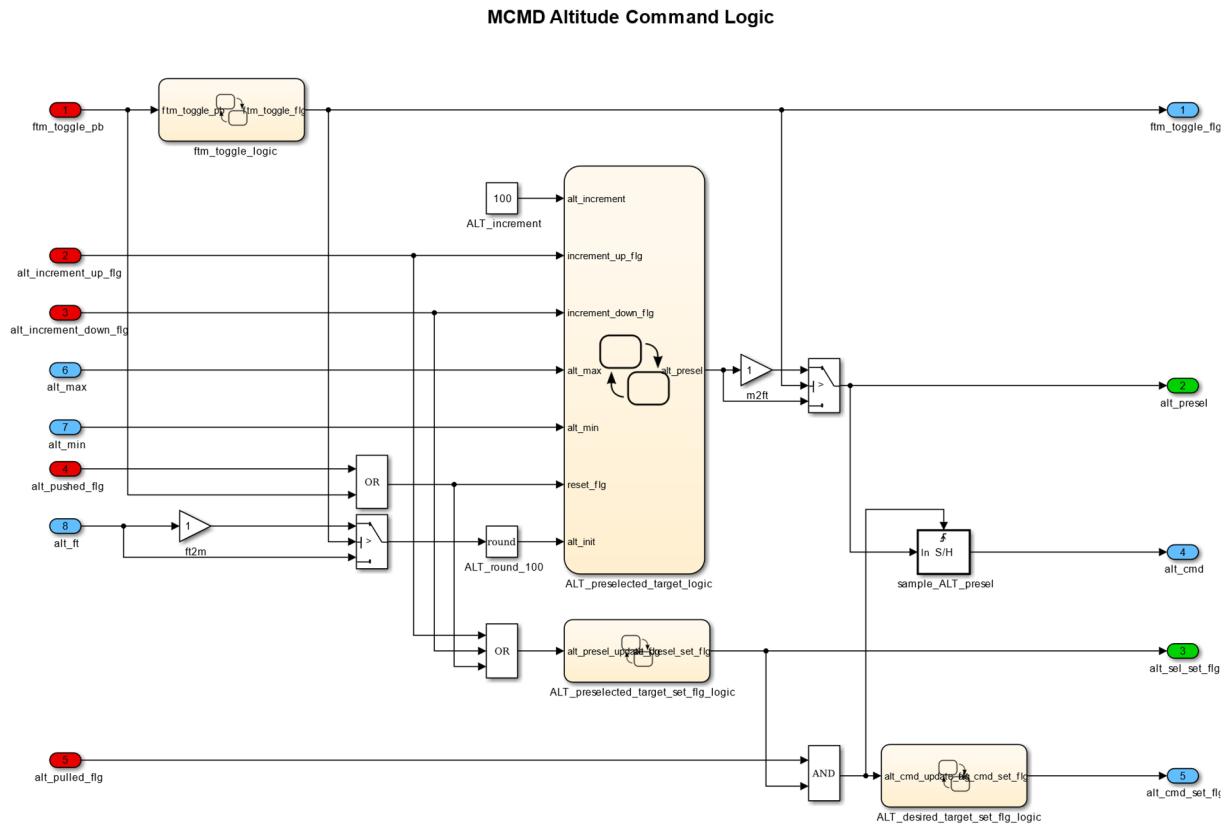


Figure C.14: Altitude command logic structure.

Feet/Meter Toggle Logic

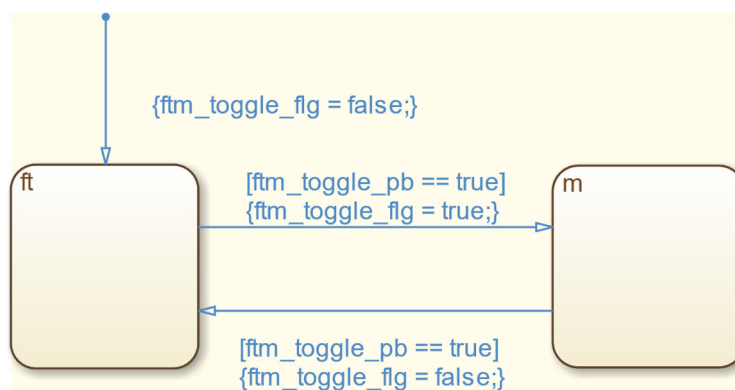


Figure C.15: Altitude feet/meter toggle logic.

Altitude Preselect Logic

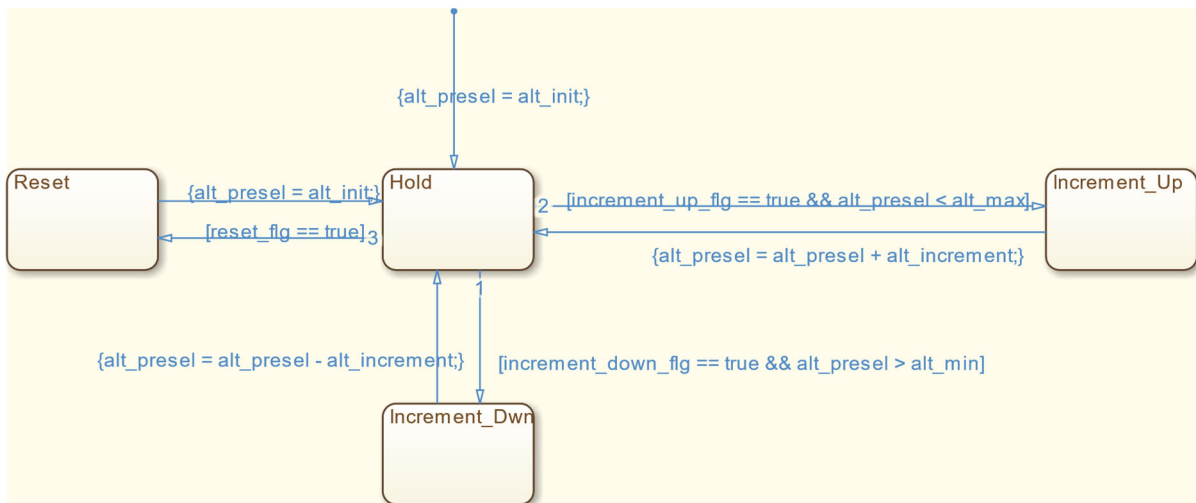


Figure C.16: Altitude preselect logic.

Altitude Preselect Target Set Flag Logic

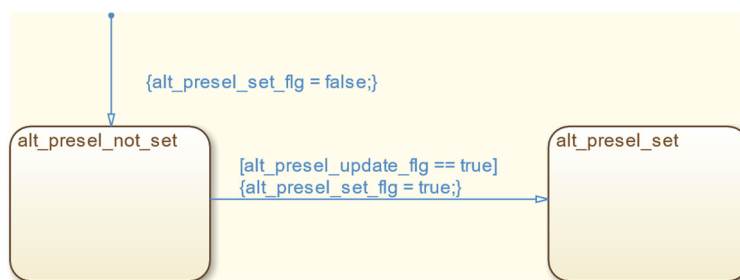


Figure C.17: Altitude preselect target set flag logic.

Altitude Desired Target Set Flag Logic

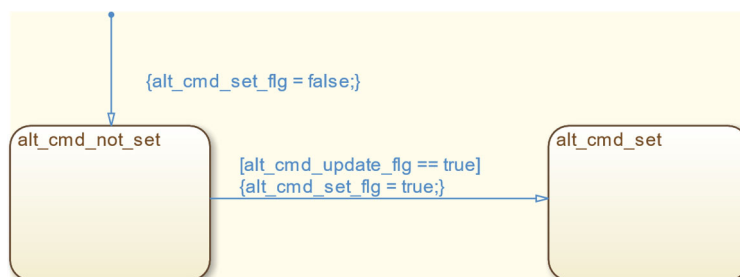


Figure C.18: Altitude desired target set flag logic.

Vertical Speed / Flight Path Angle Command Logic

Vertical Speed / Flight Path Angle Command Logic Input/Output Overview

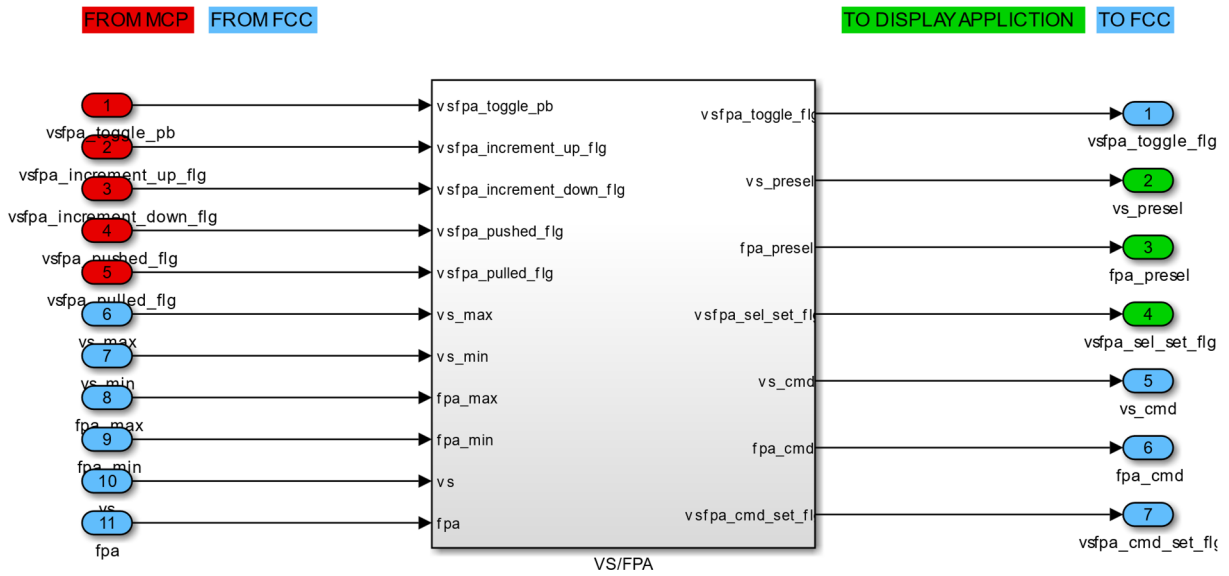


Figure C.19: Vertical speed / flight path angle command logic inputs and outputs.

Vertical Speed / Flight Path Angle Command Logic Structure

MCMD Vertical Speed / Flight Path Angle Command Logic

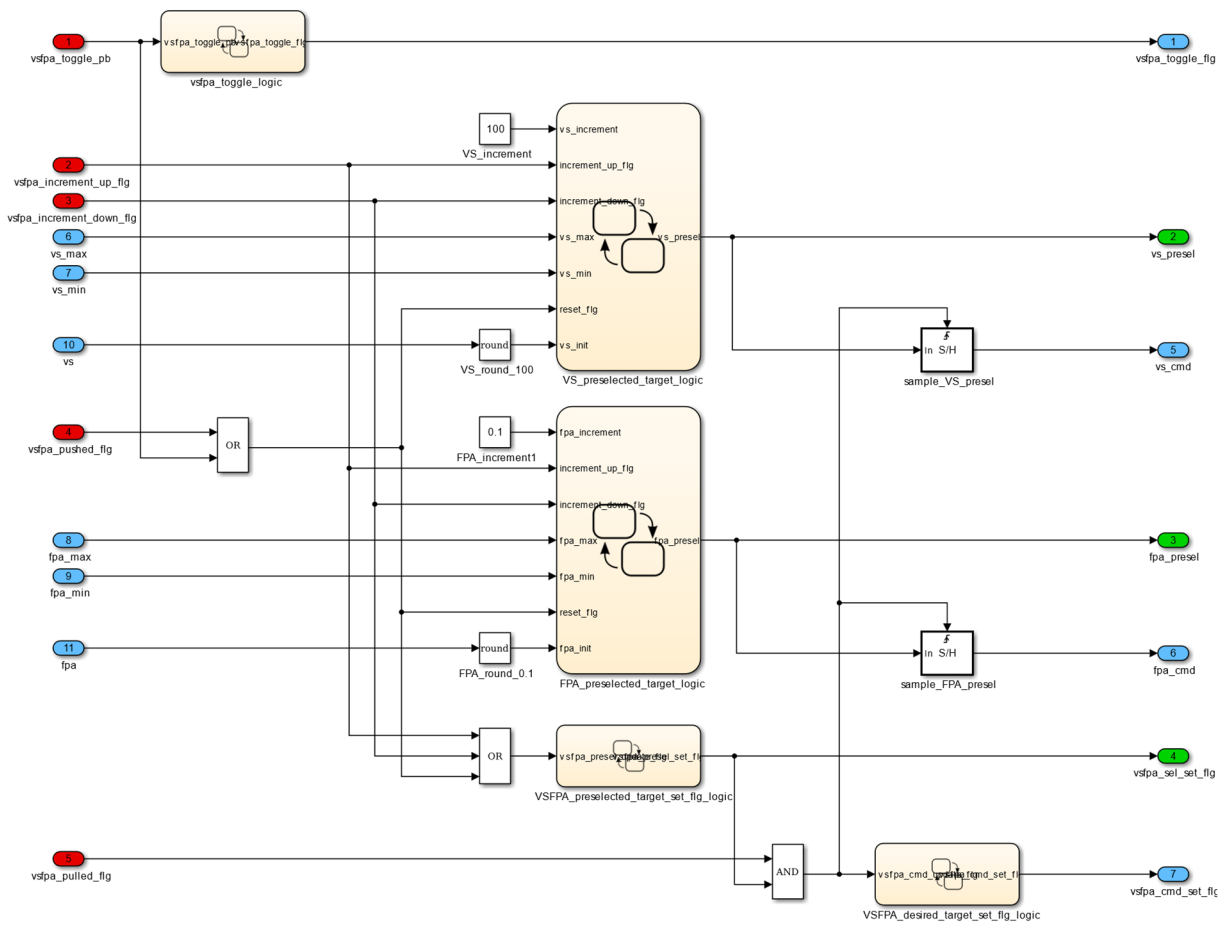


Figure C.20: Vertical speed / flight path angle command logic structure.

Vertical Speed / Flight Path Angle Toggle Logic

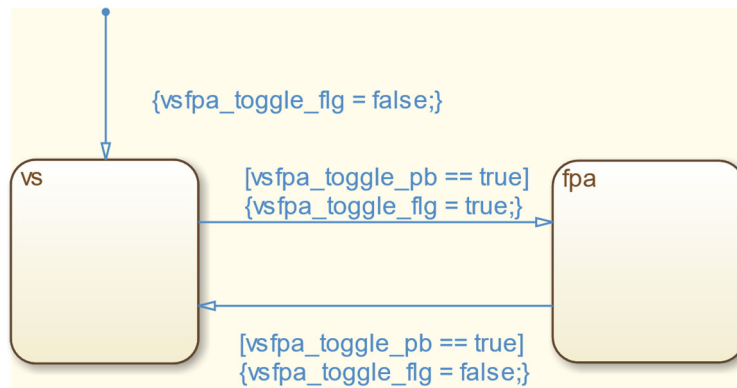


Figure C.21: Vertical speed / flight path angle toggle logic.

Vertical Speed Preselect Logic

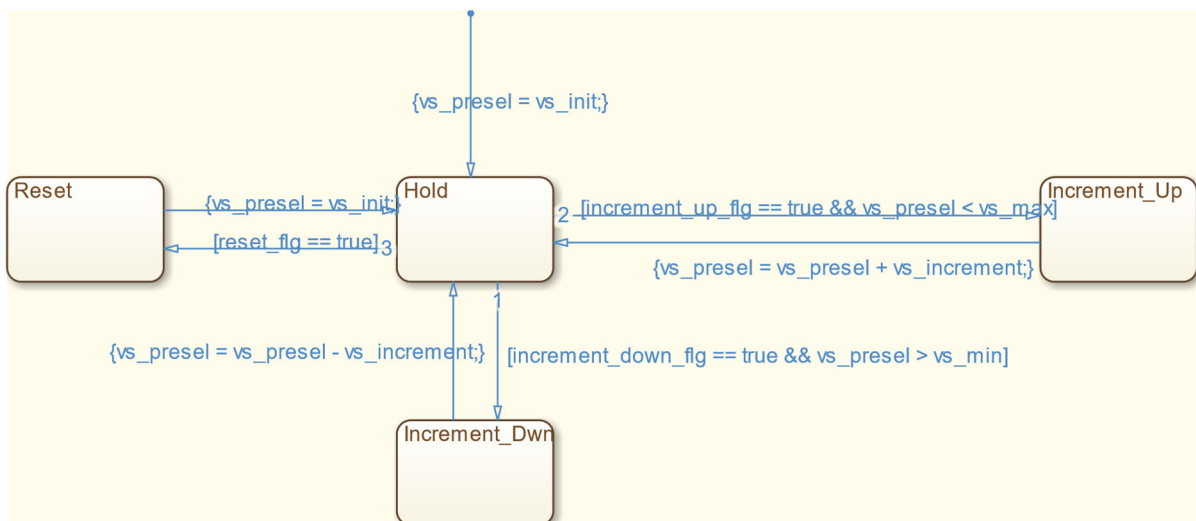


Figure C.22: Vertical speed preselect logic.

Flight Path Angle Preselect Logic

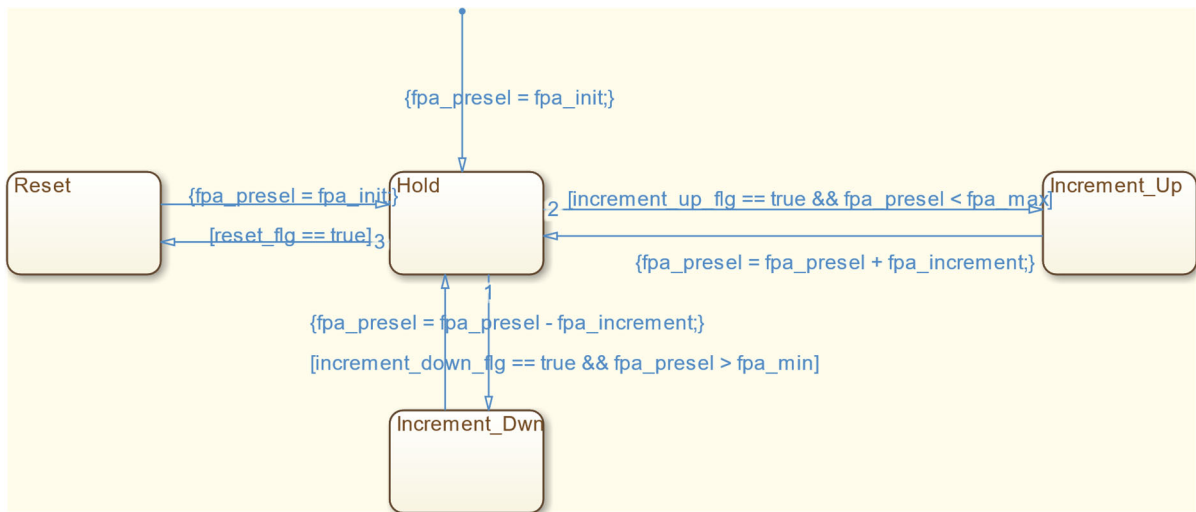


Figure C.23: Flight Path Angle preselect logic.

Vertical Speed / Flight Path Angle Preselect Target Set Flag

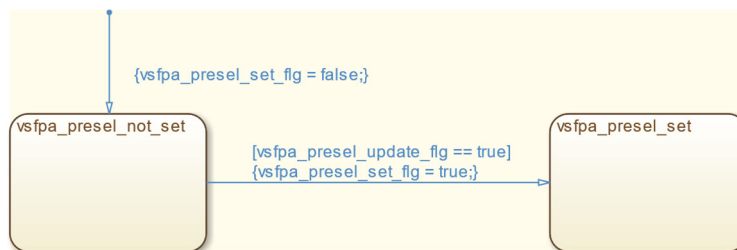


Figure C.24: Vertical speed / flight path angle preselect set flag logic.

Vertical Speed / Flight Path Angle Desired Target Set Flag

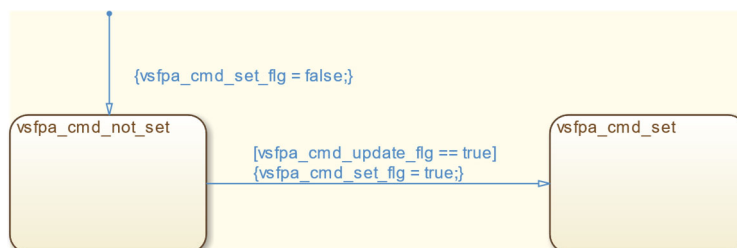


Figure C.25: Vertical speed / flight path angle desired target set flag logic.

Course Command Logic

Course Command Logic Inputs/Outputs

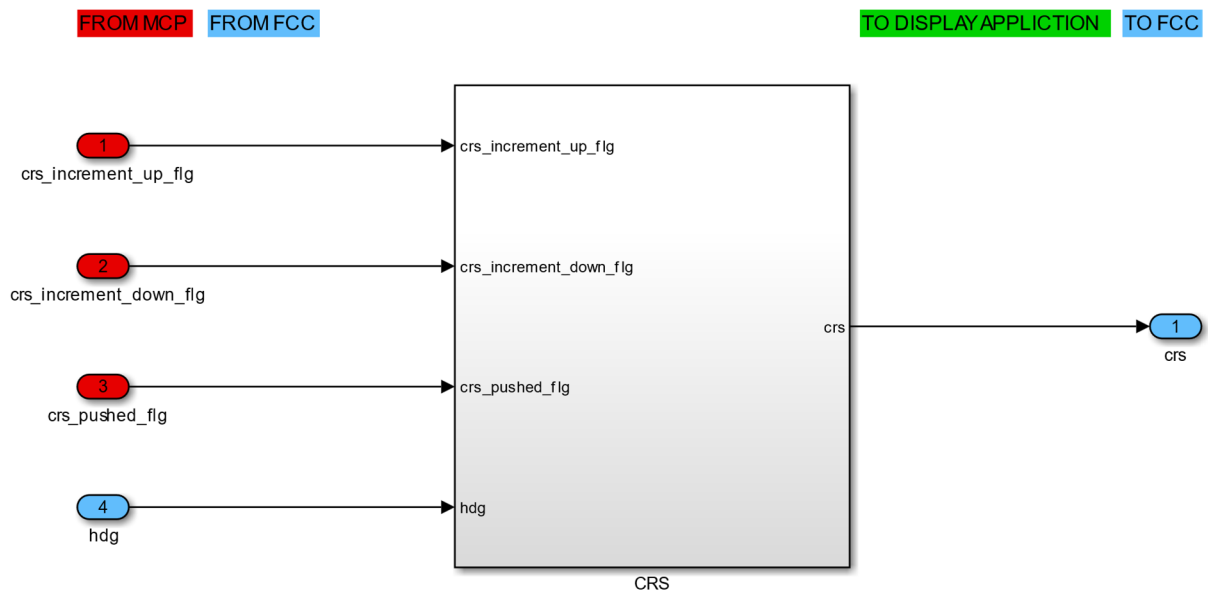


Figure C.26: Course command logic inputs and outputs.

Course Command Logic Structure

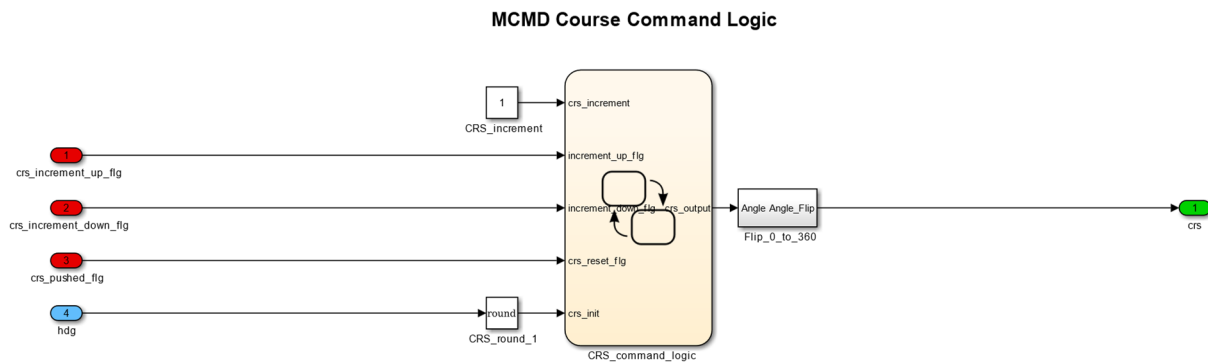


Figure C.27: Course command logic structure.

Course Command Logic

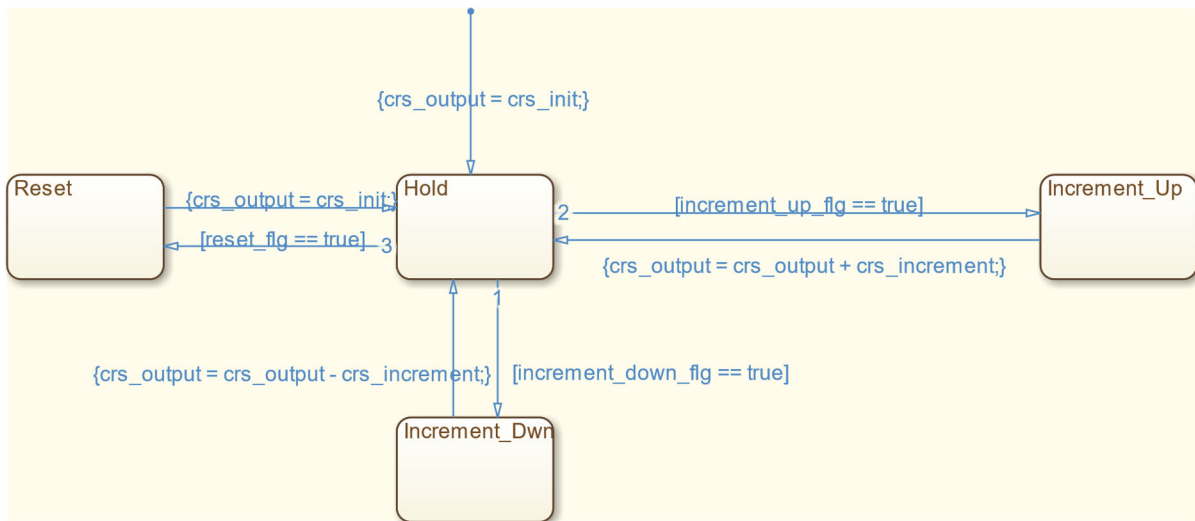


Figure C.28: Course command logic.

C.2 MCMD Elements

This section further details some of the MCMD elements, based on the MCP/MCMD command logic specification detailed in Section C.1. An overview of the MCMD layout is presented in Figure 5.34.

Preselected / Active Target Section

Airspeed Indications

Table C.1: Alternative indications for preselected airspeed (upper row).

Alternative Indications	Description
---	No preselected indicated airspeed set (default indication)
140	Preselected indicated airspeed set

Table C.2: Alternative indications for commanded airspeed (lower row).

Alternative Indications	Description
---	No commanded indicated airspeed set (default indication)
120	Commanded indicated airspeed set, but inactive (no speed function active)
120	Commanded indicated airspeed set and active (speed function active)
MCP	Indicated airspeed command source is MCP (SPD FMS is off)
FMS	Indicated airspeed command source is FMS (SPD FMS is on)

Heading / Track Indications

Table C.3: Alternative indications for preselected heading / track (upper row).

Alternative Indications	Description
---	No preselected heading or track set (default indication)
◀ 120 ▶	Preselected heading or track set
	Arrows indicate right or left turn if preselected heading/track is confirmed as new command
HDG	Heading or track toggle indication
TRK	

Table C.4: Alternative indications for commanded heading / track (lower row).

Alternative Indications	Description
---	No commanded heading or track set (default indication)
000	Commanded heading or track set, but inactive (heading or track function inactive)
000	Commanded heading or track set and active (heading or track function active)
MCP	Heading or track command source is MCP (LNAV FMS is off)
FMS	Lateral plane is managed by FMS (LNAV FMS is on)

Altitude Indications

Table C.5: Alternative indications for preselected altitude (upper row).

Alternative Indications	Description
-----	No preselected altitude set (default indication)
01200	Preselected altitude set
M	Preselected altitude shown in meter

Table C.6: Alternative indications for commanded altitude (lower row).

Alternative Indications	Description
-----	No commanded altitude set (default indication)
12000	Commanded altitude set, but inactive (altitude hold function inactive or not armed)
12000	Commanded altitude set and active (altitude hold function active or armed)
M	Commanded altitude shown in meter
MCP	Altitude command source is MCP (VNAV FMS is off)
FMS	Altitude command source is FMS (VNAV FMS is on)

Vertical Speed / Flight Path Angle Indications

Table C.7: Alternative indications for preselected vertical speed or flight path angle (upper row).

Alternative Indications	Description
---	No preselected heading or track set (default indication)
1000	Preselected vertical speed or flight path angle set
0.0	
V/S	Vertical speed or flight path angle toggle indication
FPA	

Table C.8: Alternative indications for commanded vertical speed or flight path angle (lower row).

Alternative Indications	Description
---	No commanded vertical speed or flight path angle set (default indication)
1000	Commanded vertical speed or flight path angle set, but inactive (vertical speed / flight path angle function inactive)
0.0	
2000	Commanded vertical speed or flight path angle set and active (vertical speed or flight path angle function active)
0.0	
MCP	Vertical speed or flight path angle command source is MCP (VNAV FMS is off)
FMS	Vertical plane is managed by external guidance system (VNAV FMS is on)

D Additional Flight Test Data

D.1 AFCS Deployment Flight Test Cards

Flight Test Cards – General Information Sheet	
<p>For every Trim Point or Flight Test Maneuver an Event Marker needs to be noted. In the designated fields, several Markers can be noted.</p> <p>Initial Trim Position stated at top of each Flight Test Card</p> <p>Information for Flight Test Maneuvers</p> <ul style="list-style-type: none">• Trim aircraft before each test point (airspeed ± 2kts, altitude ± 300ft, body rates 0 rad/s)• Trim point given at top of each flight test card• Hold a/c trimmed position for at least 10 s• Return to trim condition after every test maneuver• Continue with next test point	
Further Information for Data Analysis:	
Temperature (K)	
QNH (Pa)	
Wind (deg/kts)	
Mass (kg)	
CG (%MAC)	

Figure D.1: Flight Test Cards – General information sheet.

Controller Activation / Pitch/Roll Hold Test						Card 1
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
1	Theta: level flight Phi: wings level	100 kts	Theta = current Phi = current	10 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Hold initial trim position - Activate controller via cockpit control panel - Observe aircraft response for about 10 seconds 						

Figure D.2: Flight Test Card 1 – Controller activation, pitch/roll hold test.

Altitude/Roll Hold Test						Card 2
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
2	Controller Active Pitch/Roll Hold	100 kts	ALT = current	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Verify Controller active in Pitch/Roll Hold at approx. horizontal and wings level flight - Press ALT Pushbutton on Mode Control Panel - Verify Altitude Hold engaged at current altitude - Observe aircraft response for about 20 seconds 						

Figure D.3: Flight Test Card 2 – Altitude/roll hold test.

Altitude/Heading Hold Test						Card 3
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
3	Controller Active Altitude/Roll Hold	100 kts	ALT = current HDG = MCP	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Verify Controller active in Altitude/Roll Hold - Select current Heading via HDG/TRK selector knob - Pull HDG/TRK selector knob to confirm HDG target - Press HDG/TRK Pushbutton on Mode Control Panel - Verify Heading Hold engaged at desired heading - Observe aircraft response for about 20 seconds 						

Figure D.4: Flight Test Card 3 – Altitude/heading hold test.

Altitude/Heading Test – Small Right Turn Heading Change						Card 4
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
4	Controller Active Altitude/Heading Hold	100 kts	ALT = current HDG = MCP	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Verify Controller active in Altitude/Heading Hold - Select new Heading 20 degrees to the right of current heading via HDG/TRK selector knob - Pull HDG/TRK selector knob to confirm HDG target - Verify Heading changes to desired heading - Observe aircraft response for about 20 seconds 						

Figure D.5: Flight Test Card 4 – Altitude/heading test, small right turn heading change.

Altitude/Heading Test – Small Left Turn Heading Change					Card 5	
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
5	Controller Active Altitude/Heading Hold	100 kts	ALT = current HDG = MCP	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> – Verify Controller active in Altitude/Heading Hold – Select new Heading 20 degrees to the left of current heading via HDG/TRK selector knob – Pull HDG/TRK selector knob to confirm HDG target – Verify Heading changes to desired heading – Observe aircraft response for about 20 seconds 						

Figure D.6: Flight Test Card 5 – Altitude/heading test, small left turn heading change.

Altitude/Heading Test – Large Right Turn Heading Change					Card 6	
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
6	Controller Active Altitude/Heading Hold	100 kts	ALT = current HDG = MCP	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> – Verify Controller active in Altitude/Heading Hold – Select new Heading 90 degrees to the right of current heading via HDG/TRK selector knob – Pull HDG/TRK selector knob to confirm HDG target – Verify Heading changes to desired heading – Observe aircraft response for about 20 seconds 						

Figure D.7: Flight Test Card 6 – Altitude/heading test, large right turn heading change.

Altitude/Heading Test – Large Left Turn Heading Change						Card 7
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
7	Controller Active Altitude/Heading Hold	100 kts	ALT = current HDG = MCP	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> – Verify Controller active in Altitude/Heading Hold – Select new Heading 90 degrees to the left of current heading via HDG/TRK selector knob – Pull HDG/TRK selector knob to confirm HDG target – Verify Heading changes to desired heading – Observe aircraft response for about 20 seconds 						

Figure D.8: Flight Test Card 7 – Altitude/heading test, large left turn heading change.

Controller Reset / Pitch/Roll Hold Test						Card 8
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
8	Theta: level flight Phi: wings level	100 kts	Theta = current Phi = current	10 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> – EFCS disengage to reset controller – Hold initial trim position – Activate controller via cockpit control panel – Observe aircraft response for about 10 seconds 						

Figure D.9: Flight Test Card 8 – Controller reset, pitch/roll hold test.

Vertical Speed / Roll Hold Test						Card 9
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
9	Controller Active Pitch/Roll Hold	100 kts	V/S = 0 ft/min	20 s		
			V/S = 500 ft/min	20 s		
			V/S = -500 ft/min	20 s		
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Verify Controller active in Pitch/Roll Hold at approx. horizontal and wings level flight - Select new Vertical Speed 0 ft/min - Pull V/S-FPA selector knob to confirm V/S target - Push V/S/FPA pushbutton to activate V/S - Change V/S command to 500 ft/min - Change V/S command to -500 ft/min 						

Figure D.10: Flight Test Card 9 – Vertical speed/roll hold test.

Vertical Speed / Altitude Capture/Hold Test						Card 10
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
10	Controller Active Vertical Speed/Roll Hold	100 kts	V/S = 500 ft/min	20 s		
			ALT = current + 500 ft			
Instructions:				Further Notes:		
<ul style="list-style-type: none"> - Verify Controller active in Vertical Speed/Roll Hold - Select new Vertical Speed 500 ft/min - Pull V/S-FPA selector knob to confirm V/S target - Select new Altitude approx. 500 ft above current altitude - Pull ALTITUDE selector knob to confirm ALT target - Verify transition to ALT* and to ALT 						

Figure D.11: Flight Test Card 10 – Vertical speed/altitude capture/hold test.

Controller Reset / Pitch/Roll Hold Test						Card 11
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
11	Theta: level flight Phi: wings level	100 kts	Theta = current Phi = current	10 s		
Instructions: <ul style="list-style-type: none"> - EFCS disengage to reset controller - Hold initial trim position - Activate controller via cockpit control panel - Observe aircraft response for about 10 seconds 				Further Notes:		

Figure D.12: Flight Test Card 11 – Controller reset, pitch/roll hold test.

Open Climb / Heading Test						Card 12
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
12	Controller Active Pitch/Roll Hold	100 kts	ALT = current +500 ft HDG = MCP	60 s		
Instructions: <ul style="list-style-type: none"> - Activate Altitude Hold via ALT pushbutton - Select Heading and pull HDG/TRK selector knob - Activate Heading via HDG/TRK pushbutton - Verify speed approx. at speed target - Select new Altitude approx. 500 feet above current altitude via ALTITUDE selector knob - Pull ALTITUDE selector knob to confirm ALT target - Verify Pitch Autopilot changes to SPD, Thrust to CLB - Manually slowly increase thrust - Observe Speed by Pitch performance - Verify transition to ALT* and to ALT 				Further Notes:		

Figure D.13: Flight Test Card 12 – Open climb/heading test.

Open Descent / Heading Test						Card 13
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
13	Controller Active Altitude/Heading Hold	100 kts	ALT = current -500 ft HDG = MCP	60 s		
Instructions: <ul style="list-style-type: none"> – Verify Controller active in Altitude/Heading Hold – Verify speed approx. at speed target – Select new Altitude approx. 500 feet below current altitude via ALTITUDE selector knob – Pull ALTITUDE selector knob to confirm ALT target – Verify Pitch Autopilot changes to SPD, Thrust to DES – Manually slowly reduce thrust – Observe Speed by Pitch performance – Verify transition to ALT* and to ALT 				Further Notes:		

Figure D.14: Flight Test Card 13 – Open descent/heading test.

Controller Reset / Pitch/Roll/Speed Hold Test						Card 14
ALT: Level Flight	Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero	
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
14	Theta: level flight Phi: wings level <u>All Axes + ATHR ON</u>	100 kts	Theta = current Phi = current	20 s		
Instructions: <ul style="list-style-type: none"> – EFCS disengage to reset controller – Hold initial trim position – Activate controller via cockpit control panel – Observe aircraft response for about 20 seconds 				Further Notes:		

Figure D.15: Flight Test Card 14 – Controller reset, pitch/roll/speed hold test.

Altitude / Heading / Speed Test						Card 15
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
15	Controller Active Pitch/Roll/Speed Hold	100 kts	ALT = current HDG = MCP	20 s		
			SPD = current + 10 kts	60 s		
			SPD = current – 15 kts	60 s		
Instructions: <ul style="list-style-type: none"> – Activate Altitude Hold via ALT pushbutton – Select Heading and pull HDG/TRK selector knob – Activate Heading via HDG/TRK pushbutton – Select new speed target 10 kts above current speed via SPEED selector knob – Pull SPEED selector knob to confirm speed target – Select new speed target 15 kts below current speed via SPEED selector knob – Pull SPEED selector knob to confirm speed target 				Further Notes:		

Figure D.16: Flight Test Card 15 – Altitude/heading/speed test.

Altitude / Lateral Trajectory / Speed Test						Card 16
ALT: Level Flight		Airspeed: 100 kts	Flaps: Clean	Gear: Up	POWER: TFLF	Body rates zero
Test No.	Initial Conditions	Airspeed	Commands	Duration	Markers	Notes
16	Controller Active Altitude/Heading/Speed Hold	100 kts	ALT = current HDG = MCP	TBD		
Instructions: <ul style="list-style-type: none"> – Verify Controller active in Altitude/Heading/Speed Hold – Activate Lateral Trajectory controller via LNAV FMS pushbutton – Observe aircraft response for as long as desired 				Further Notes:		

Figure D.17: Flight Test Card 16 – Altitude/lateral trajectory/speed test.

D.2 Flight Test Logs

Table D.1: List of flight tests included in this appendix with reference to figures.

Test Date	Test Location	Test Flight	Test Scope	Figure Reference
2016-08-17	LOAN	1	Preparatory demonstration flight, tracking of lateral trajectory	Figure D.18 Figure D.19
2016-08-17	LOAN	2	AFCS flight test, additional control modes and speed protections Test of flare mode in preparation for later ATOL flight testing	Figure D.20 Figure D.21
2016-08-18	LOAN	1	Demonstration flight, tracking of lateral trajectory	Figure D.22 Figure D.23
2016-08-18	LOAN	2	AFCS flight test, additional control modes and speed protections Test of flare mode in preparation for later ATOL flight testing	Figure D.24 Figure D.25

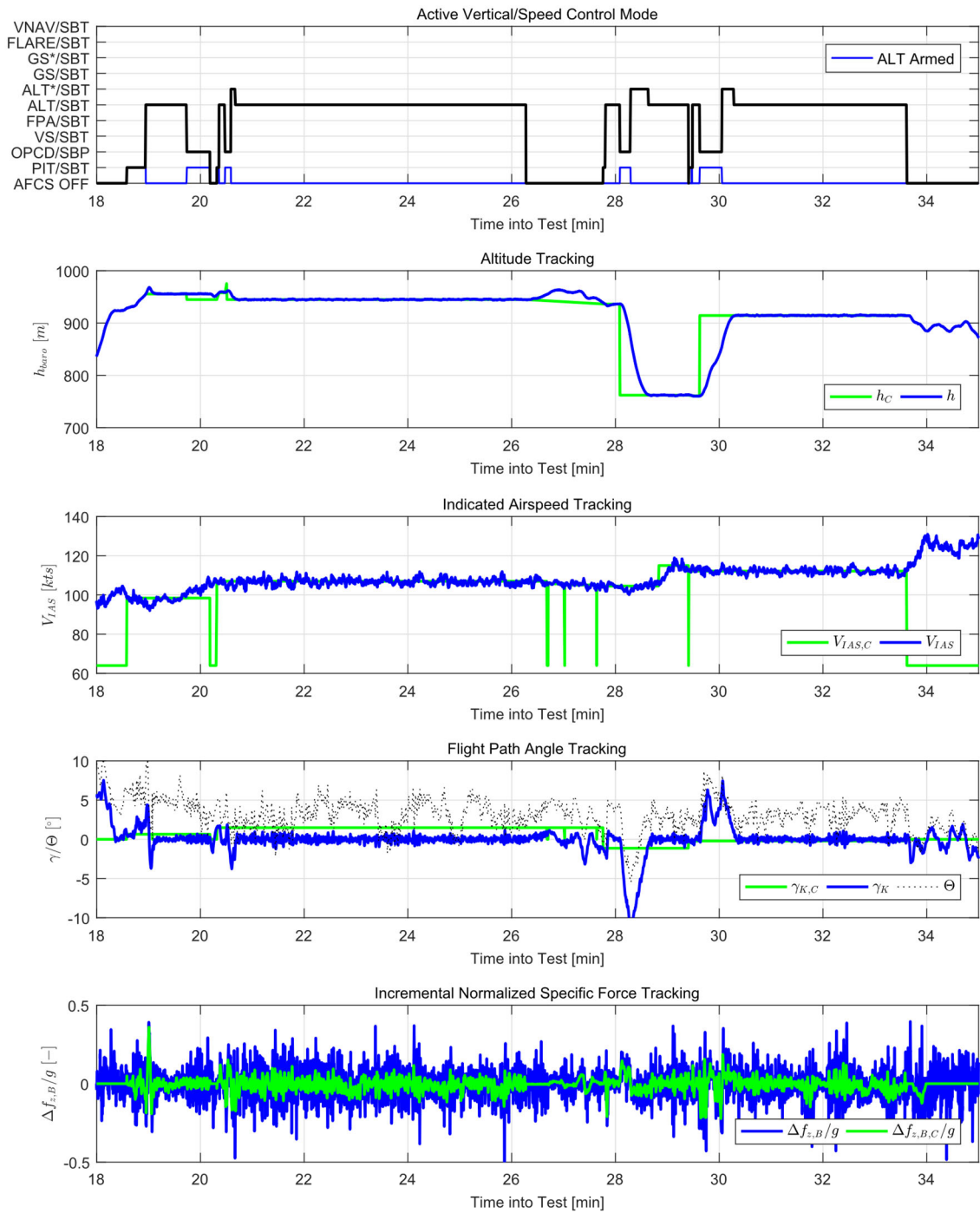


Figure D.18: 2016-08-17, flight test #1. Demonstration check flight, tracking of lateral trajectory. Vertical plane/speed active control modes and command tracking.

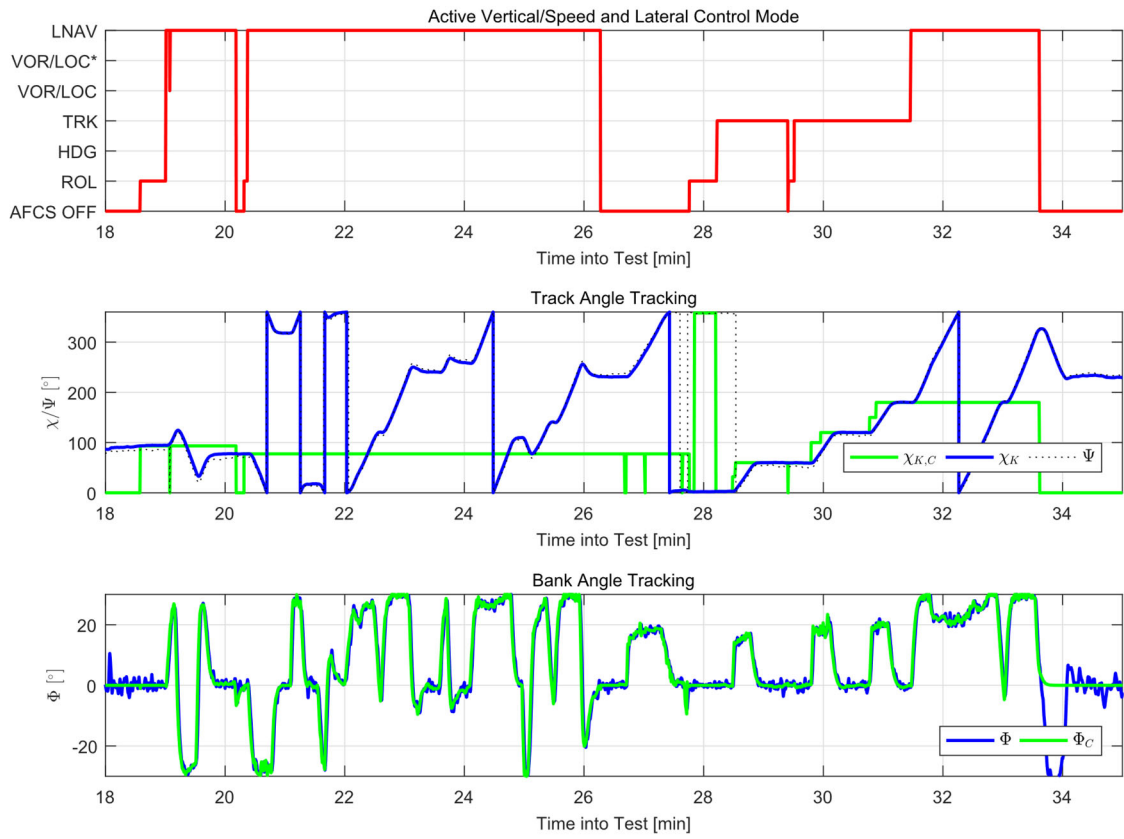


Figure D.19: 2016-08-17, flight test #1. Demonstration check flight, tracking of lateral trajectory. Lateral plane active control modes and command tracking

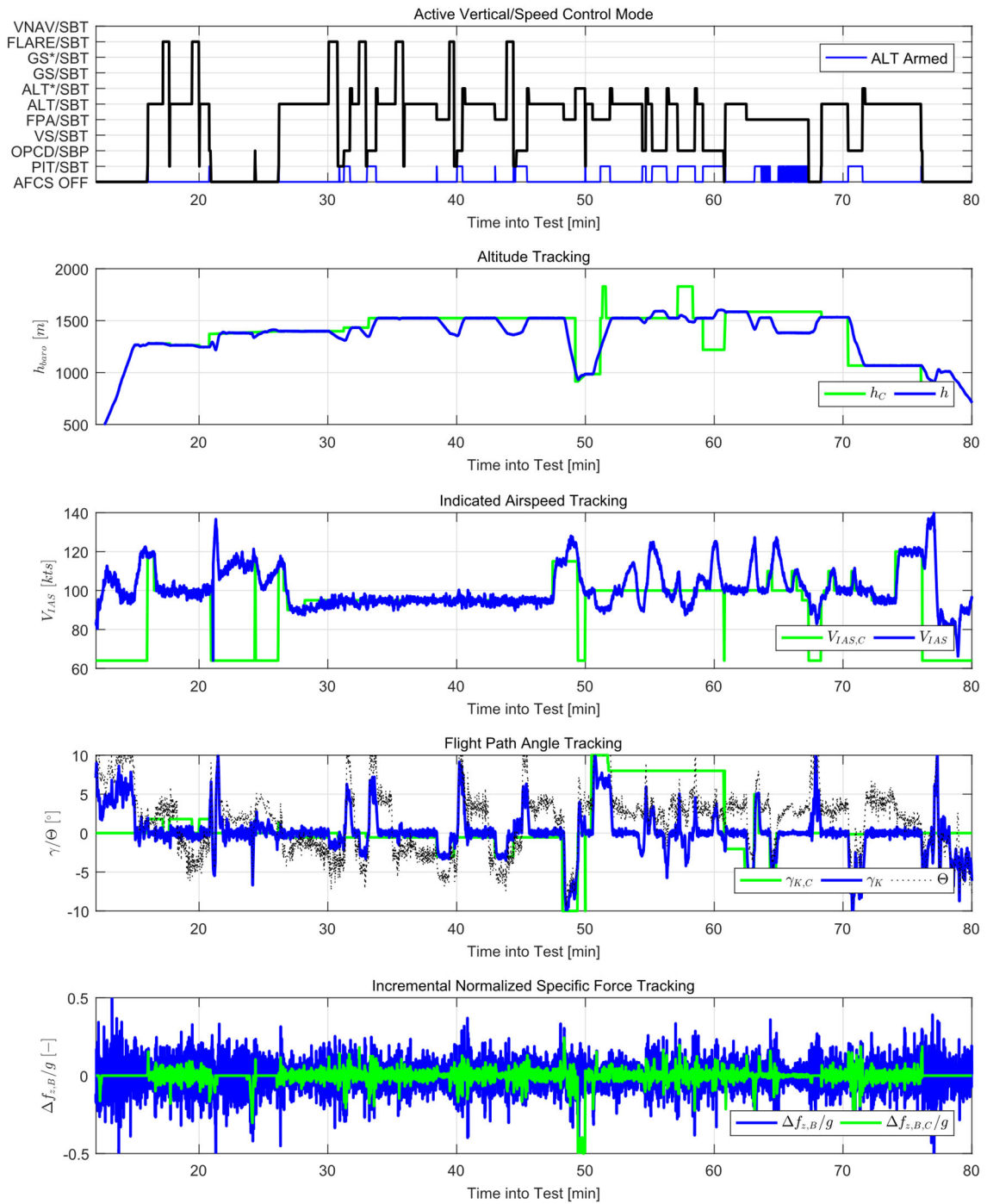


Figure D.20: 2016-08-17, flight test #2. AFCS additional flight tests. Vertical plane/speed active control modes and command tracking.

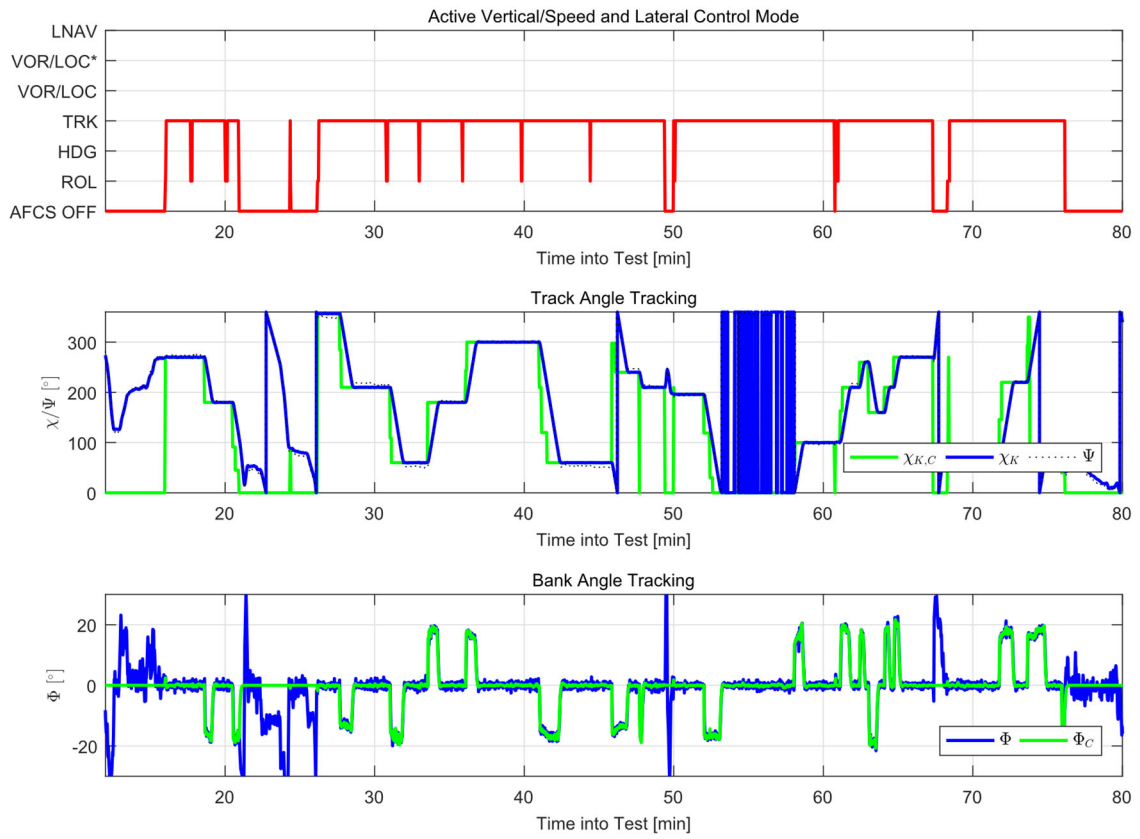


Figure D.21: 2016-08-17, flight test #2. AFCS additional flight tests. Lateral plane active control modes and command tracking.

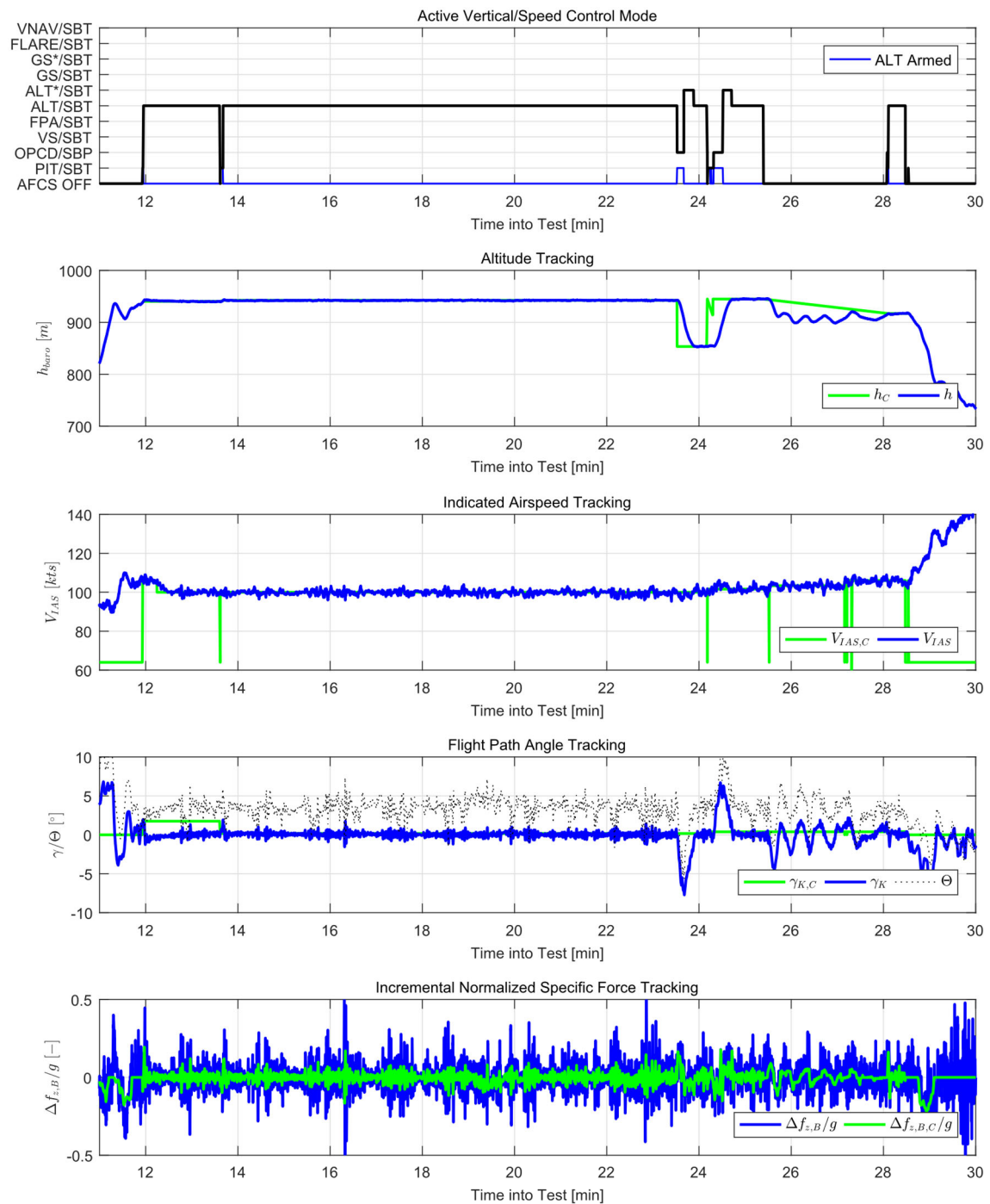


Figure D.22: 2016-08-18, flight test #1. Demonstration flight, tracking of lateral trajectory. Vertical plane/speed active control modes and command tracking.

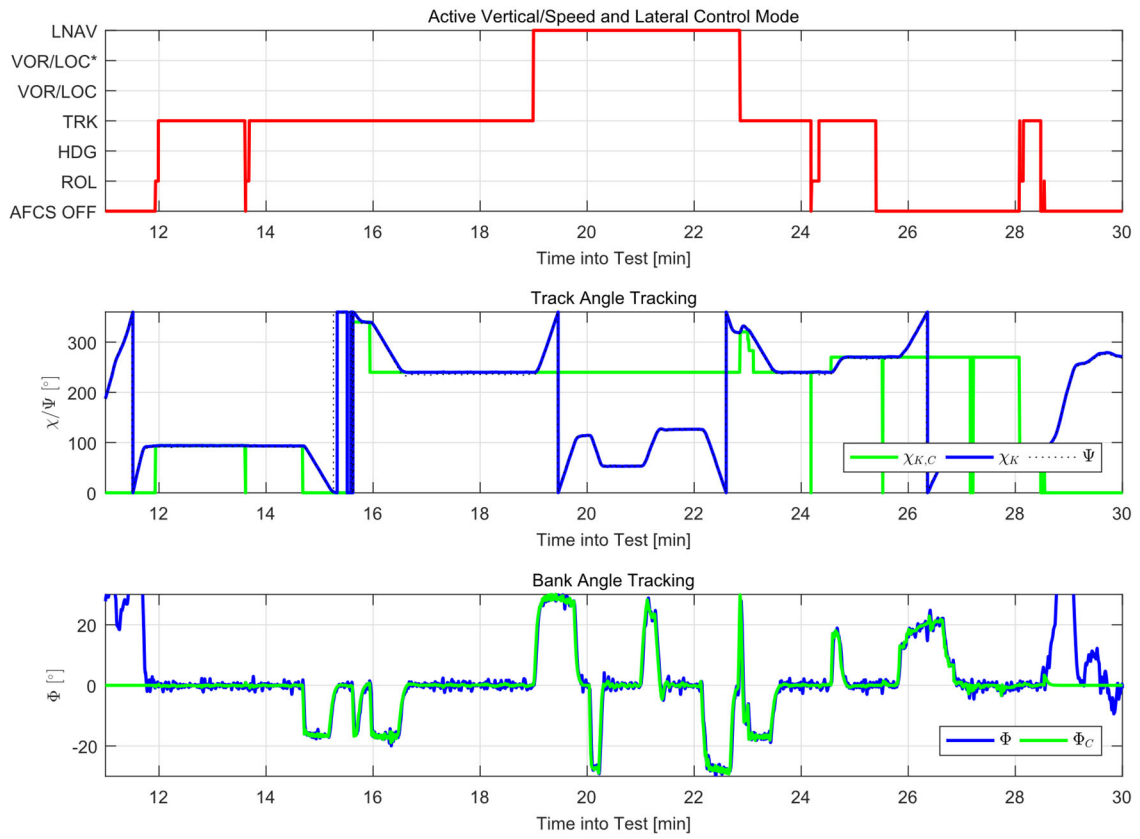


Figure D.23: 2016-08-18, flight test #1. Demonstration flight, tracking of lateral trajectory. Lateral plane active control modes and command tracking.

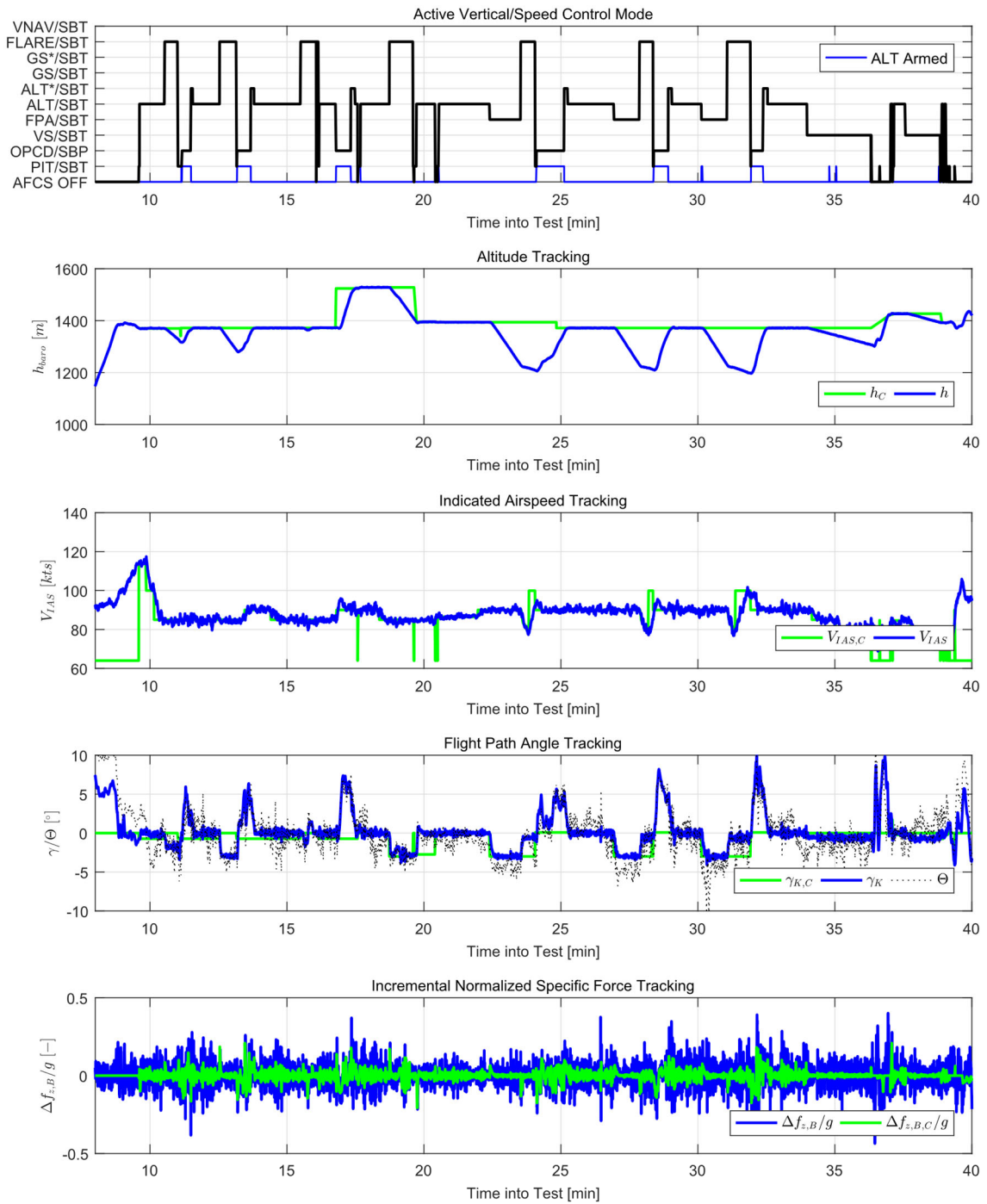


Figure D.24: 2016-08-18, flight test #2. AFCS flight test, ATOL preparatory tests. Vertical plane/speed active control modes and command tracking.

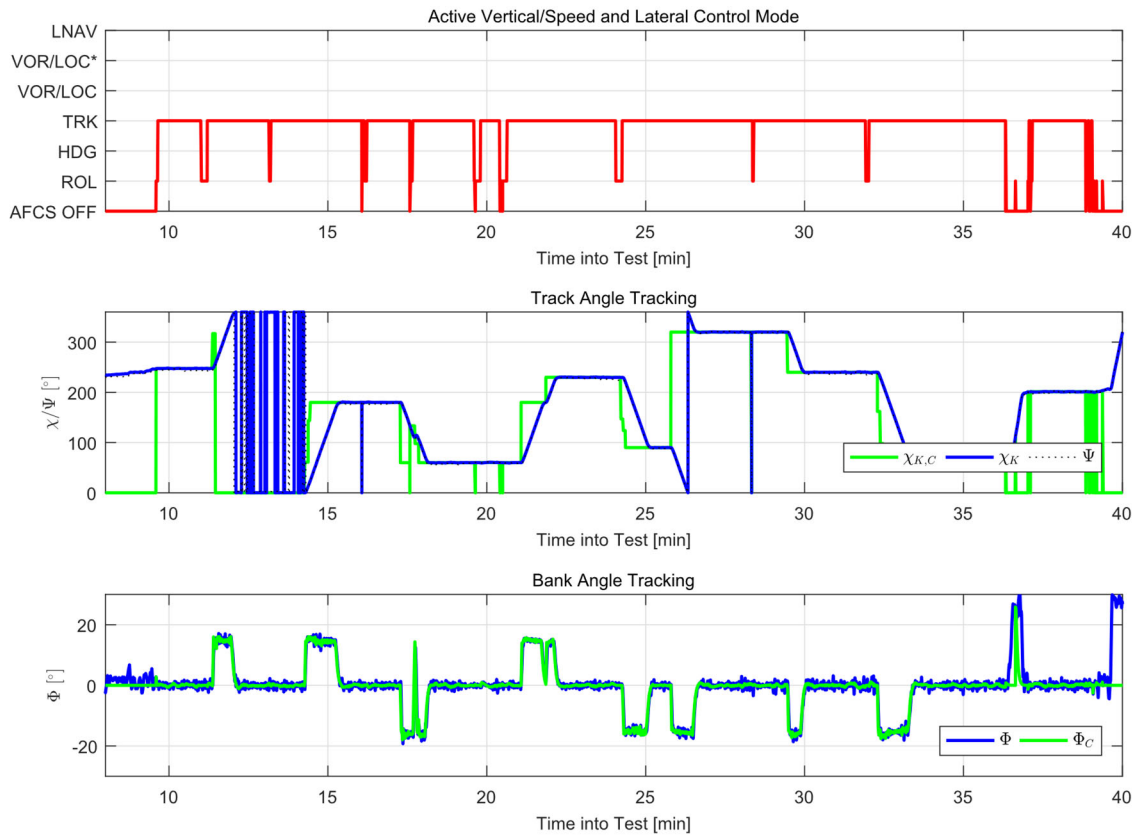


Figure D.25: 2016-08-18, flight test #2. AFCS flight test, ATOL preparatory tests. Lateral plane active control modes and command tracking.

E Functional Hazard Assessment Worksheets

E.1 FHA Worksheet Explanation

Table E.1: FHA worksheet columns and explanation.

Column	Explanation
Function	Function name
Failure Condition	Description of failure condition
Phase	Flight phase negatively affecting the effect of the analyzed failure condition: <ul style="list-style-type: none"> • Departure / Descent • Enroute / Return (Cruise) • Mission (Maneuvering Flight) • Approach
Effect of Failure Condition on Aircraft/Crew	Description of the effect of the failure condition at aircraft level, with respect to the classification criteria in the AC 23.1309-1E
Classification	Resulting classification according to the failure effect: <ul style="list-style-type: none"> • NO = No safety effect • MIN = Minor • MAJ = Major • HAZ = Hazardous • CAT = Catastrophic
Reference to supporting material	Reference to acceptable means of compliance, industry standards, test reports or other documentation justifying the failure effect and/or its classification

Column	Explanation
Verification	<p>Method of showing compliance with classification and related probability requirements</p> <ul style="list-style-type: none">• Qualitative design appraisal• Design and installation appraisal to verify similarity with previously installed system• Quantitative Fault Tree Analysis (for severe failure conditions related to new function or design)<ul style="list-style-type: none">○ PSSA: Using estimated component failure rates○ SSA: Using failure rates of implemented components

E.2 Depth of Analysis

In the column “Verification” of the FHA Worksheet, the analysis required to verify the safety requirements related to the failure condition is defined. Table summarizes the depth of analysis requirements in AC 23.1309-1E [104].

Table E.2: Depth of analysis requirements for failure condition classification and type of system combinations.

Failure condition classification	Type of system	Type and depth of analysis
No Safety Effect & Minor	All	Design and installation appraisal to <ul style="list-style-type: none"> • verify that the analyzed function is independent • verify that the failure condition is isolated from other functions (especially those required for safe operation of the aircraft).
Major	System is similar in its relevant attributes to other already certified systems, and the effects of the functional failure conditions would be the same	Design and installation appraisal to <ul style="list-style-type: none"> • verify claimed similarity
	Simple system but no similarity to previously installed systems	Qualitative assessment (e.g., qualitative FTA or functional FMEA) to <ul style="list-style-type: none"> • verify that the failure conditions of the system are consistent with the ones identified in the FHA
	Complex and non-redundant system	Qualitative and quantitative assessment (e.g., FTA supported by failure rate data) to: <ul style="list-style-type: none"> • verify that malfunctions are indeed remote
	Redundant system	Qualitative analysis (e.g., qualitative FTA or functional FMEA and Common Mode Analysis) to <ul style="list-style-type: none"> • verify that the failure conditions of the system are consistent with the ones identified in the FHA • verify sufficient independence between redundant systems
Hazardous & Catastrophic	Simple and conventional installations	Qualitative analysis to verify <ul style="list-style-type: none"> • similarity with other previously installed systems • degree of redundancy • sufficient independent and isolation of channels • sufficient reliability of the technology involved
	Complex systems with true similarity in all relevant attributes	Qualitative analysis to <ul style="list-style-type: none"> • establish the high degree of similarity in both design and application
	All other systems	Qualitative and quantitative assessment (e.g., FTA)

E.3 Automatic Flight Control Functions

Table E.3: Guidance modes FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Selected Operation	Total loss of flight guidance command from MCP	All	Flight crew detects inability to activate/change AFCS mode or select mode target reference values (e.g., speed or altitude), disconnects AFCS. Reduction in operational capabilities of the system Slight increase in crew workload during manual flight.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous flight guidance command from MCP	All	Flight crew detects incorrect mode or mode target indication on MCMD, disconnects AFCS. Reduction in operational capabilities of the system Slight increase in crew workload during manual flight.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
Managed Operation	Total loss of flight guidance command from external guidance system	All	Flight crew detects inability to activate/change AFCS mode or select mode target reference values (e.g., speed or altitude), disconnects AFCS. Reduction in operational capabilities of the system Slight increase in crew workload during manual flight.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous flight guidance command from external guidance system	All	Flight crew detects incorrect mode or mode target indication on MCMD, disconnects AFCS. Reduction in operational capabilities of the system Slight increase in crew workload during manual flight.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal

Table E.4: Coupled path and trajectory control modes FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Automatic Vertical Path Control	Total loss of vertical path control with indication	Approach	Crew recognizes and takes over control. Slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E	Qualitative design appraisal
	Total loss of vertical path control without invitation	Approach	Crew must recognize by secondary means (deviation from flight path) and take over control. Significant increase in crew workload.	MAJ	C	10 ⁻⁵	AC 23.1309-1E	Design and installation appraisal to verify similarity with previously installed system
	Erroneous vertical path control, slowover malfunction	Approach	Flight crew must monitor autopilot operation, recognizes erroneous or inconsistent vertical control or gradual departure from reference path. Pilot must disconnect autopilot manually to recover flight promptly. Reductions in safety margins and significant increase in crew workload.	MAJ	C	10 ⁻⁵	AC 23.1309-1E Appendix 1, p. A1-6	Design and installation appraisal to verify similarity with previously installed system
	Erroneous vertical path control, hardover malfunction	Approach	Flight crew recognizes unexpected aircraft behavior and disconnects autopilot to recover flight promptly. Large reductions in safety margins and large increase in crew workload. For nose down hardover, risk of not meeting obstacle clearance. For nose up hardover, risk of reaching stall speed	CAT	C	10 ⁻⁷	AC 23.1309-1E Appendix 1, p. A1-6 Multi-axis, unlimited authority AFCS	Quantitative Fault Tree Analysis PSSA and SSA
Automatic Lateral Path Control	Total loss of lateral path control, annunciated	All	Crew recognizes and takes over control. Slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Total loss of lateral path control, no annunciation	Approach	Crew must recognize by secondary means (deviation from flight path) and take over control. Slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous lateral path control, slowover malfunction	Approach	Flight crew must monitor autopilot operation, recognizes erroneous or inconsistent vertical control or	MIN	D	10 ⁻³	AC 23.1309-1E	Qualitative design appraisal

			gradual departure from reference path. Disconnects autopilot to recover flight promptly. Risk of coming too close to adjacent runways.					
	Erroneous lateral path control, hardover malfunction	Approach	Flight crew recognizes unexpected aircraft behavior, e.g., significant bank angle and deviation from intended runway track and disconnects autopilot to recover flight promptly. Large reductions in safety margins and large increase in crew workload.	CAT	C	10 ⁻⁷	AC 23.1309-1E Appendix 1, p. A1-6 Multi-axis, unlimited authority AFCS	Quantitative Fault Tree Analysis PSSA and SSA
Automatic Thrust / Speed Control	Total loss of thrust / speed control with indication	All	Slight reduction in operational capabilities, slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E	
	Total loss of thrust / speed control without indication	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	C	10 ⁻⁵	AC 23.1309-1E	
	Erroneous thrust / speed control with indication	All	Flight crew recognizes unexpected aircraft behavior, e.g., unexpected acceleration or deceleration, and disconnects autothrust. Slight reduction in operational capabilities, slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E	
	Erroneous thrust / speed control without indication	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	C	10 ⁻⁵	AC 23.1309-1E	
	Asymmetric thrust with annunciation	All	Flight crew recognizes autothrust failure, disengages and handles the throttle manually. Slight increase in crew workload.	MIN	D	10 ⁻³	AC 23.1309-1E	
	Asymmetric thrust without annunciation, autopilot engaged	All	Autopilot tries to compensate asymmetric thrust setting with roll/yaw commands, pilot may not immediately recognize the failure	MAJ	C	10 ⁻⁵	AC 23.1309-1E	Quantitative Fault Tree Analysis PSSA and SSA

			condition. Significant reduction in safety margins.					
Manual Control with Vertical Flight Guidance	Total loss of vertical flight guidance command display	Approach	Slight increase in crew workload during manual flight. May cause go-around.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Misleading vertical flight guidance command display	Approach	Guidance inconsistent with intended flight path, identified by the crew. Reduction of safety margins.	MAJ	C	10 ⁻⁵	AC 23.1309-1E Appendix 1, p. A1-6	Design and installation appraisal to verify similarity with previously installed system
Manual Control with Lateral Flight Guidance	Total loss of lateral flight guidance command	Approach	Slight increase in crew workload during manual flight. May cause go-around.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Misleading lateral flight guidance command	Approach	Guidance inconsistent with intended flight path, identified by the crew.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal

Table E.5: Coupled control objective prioritization modes FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Path/Speed Priority	Total loss of path/speed priority	Mission	Inability to achieve desired mission profile. Slight reduction in functional capabilities or safety margins. Slight increase in crew workload to compensate.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous path/speed priority	Mission	Inability to achieve desired mission profile. Slight reduction in functional capabilities or safety margins. Slight increase in crew workload to compensate.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
Vertical/ Lateral Plane Priority	Total loss of vertical/lateral control plane priority	Mission	Inability to achieve desired mission profile. Slight reduction in functional capabilities or safety margins. Slight increase in crew workload to compensate.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous vertical/lateral control plane priority	Mission	Inability to achieve desired mission profile. Slight reduction in	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal

			functional capabilities or safety margins. Slight increase in crew workload to compensate.					
Energy Protection	Total loss of low speed protection	Approach	Flight crew must monitor autopilot operation to avoid low energy state on approach. Slight increase in crew workload and slight reduction in safety margins.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous low speed protection	Approach	Flight crew may not immediately recognize the failure condition. May cause low energy state on approach. Significant reduction in safety margins.	MAJ	C	10 ⁻⁵	AC 23.1309-1E Appendix 1, p. A1-6	Design and installation appraisal to verify similarity with previously installed system
	Total loss of high speed protection	Mission	Flight crew must monitor autopilot operation to avoid overspeed situation. Slight increase in crew workload and slight reduction in safety margins.	MIN	D	10 ⁻³	AC 23.1309-1E Appendix 1, p. A1-6	Qualitative design appraisal
	Erroneous high speed protection	Mission	Flight crew may not immediately recognize the failure condition. May cause overspeed situation during maneuvering in mission phase. Significant reduction in safety margins.	MAJ	C	10 ⁻⁵	AC 23.1309-1E Appendix 1, p. A1-6	Design and installation appraisal to verify similarity with previously installed system

E.4 HMI Functions

Table E.6: HMI Functions FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Mode Selection	Failure to select AFCS mode	All	Mode cannot be activated/changed. Crew disconnects FD and continues without guidance. Slight increase in crew workload.	MIN	D	10 ⁻³		
	Incorrect AGFCS mode selection (active mode indication is correct)	All	Wrong mode is selected and annunciated. Crew disconnects FD and continues without guidance. Slight increase in crew workload.	MIN	D	10 ⁻³		
Annunciation of Active Mode	Loss of AFGCS mode indication	All	Flight crew unable to determine activate AFGCS mode, AFGCS disconnect and manual flying without flight director guidance	MIN	D	10 ⁻³		Qualitative
	Incorrect AFGCS mode indication	All	Flight crew detects incorrect mode indication, however, cannot determine whether mode selection or just indication is incorrect, disconnects AFGCS. Slight reduction in functional capabilities, slight increase in crew workload	MIN	D	10 ⁻³		Qualitative
Annunciation of Mode Transition	No annunciation of automatic mode transition (e.g., from capture to acquire to hold mode)	All	Pilot cannot track expected and automatic mode changes. May cause confusion regarding the operation of the AFGCS.	MIN	D	10 ⁻³		Qualitative
Mode Target Selection	Inability to select mode target reference values (e.g., speed or altitude)	All	Reduction in operational capabilities of the system.	MIN	D	10 ⁻³		Qualitative
Flight Director Engagement	Inability to engage flight director	All	No guided flight available. Slight reduction in operational capabilities.	MIN	D	10 ⁻³		Qualitative design appraisal
	Inadvertent engagement of flight director	All	Crew recognizes unintended activation of the flight director, disengages it manually	NO	N/A	N/A		Qualitative design appraisal
	No flight director engagement annunciation on display or MCP	All	Crew recognizes flight director is active through present guidance commands on the PFDs.	NO	N/A	N/A		Qualitative design appraisal

Flight Director Disengagement	Inability to disengage flight director via Mode Control Panel	All	Crew recognizes that flight director is still active. Guidance commands on display might disturb pilots if intention is other than active flight director mode. Slight increase in crew workload.	MIN	D	10 ⁻³		Qualitative design appraisal
	Inadvertent disengagement of flight director	Approach	Slight increase in crew workload during manual flight. May cause go-around.	MIN	D	10 ⁻³	AC 23.1309-1E	Qualitative design appraisal
	No flight director disengagement annunciation	All	Crew recognizes flight director is disengaged through absent guidance commands on the PFDs.	NO	N/A	N/A		Qualitative design appraisal

E.5 Autopilot Functions

Table E.7: Autopilot Functions FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Autopilot Engagement	Inability to engage autopilot	All	Slight increase in crew workload due to manual flight.	MIN	D	10 ⁻³		Qualitative design appraisal
	Inadvertent engagement of autopilot function	All	Crew recognizes that the autopilot is engaged, disconnects, and resumes manual flight. Slight increase in crew workload.	MIN	D	10 ⁻³		Qualitative design appraisal
	No autopilot engagement annunciation	All	Crew cannot directly determine whether autopilot has been successfully engaged. Slight increase in crew workload.	MIN	D	10 ⁻³		Qualitative design appraisal
Autopilot Disengagement	Loss of ability to disconnect the autopilot via Mode Control Panel	All	Crew disengages autopilot by other means (quick-disconnect switch or circuit breakers). Slight increase in crew workload.	MIN	D	10 ⁻³		Qualitative design appraisal
	Loss of the ability to disengage the autopilot with control column disconnect switch	Approach	Pilot not able to disconnect autopilot at decision height or during autopilot or trim failure. Pilot to recognize this situation and overcome the autopilot servos and control the aircraft. Autopilot must be disengaged by other means (e.g., circuit breakers). Increase in crew workload.	MAJ	C	10 ⁻⁵	AC 23.1309-1E Appendix 1, p. A1-7	Qualitative design appraisal
	No autopilot disengagement annunciation	All	Crew cannot directly determine whether autopilot has been successfully disengaged. Slight increase in crew workload.	MIN	D	10 ⁻³		
	Inadvertent disengagement with annunciation	All	Pilot recognizes that autopilot has disengaged and takes manual control of the aircraft. Slight increase in pilot workload.	MIN	D	10 ⁻³		Qualitative design appraisal
	Inadvertent disengagement without annunciation	All	Pilot recognizes that autopilot has disengaged by deviation from intended flight path, takes manual action to retain control of the aircraft. Increase in crew workload and reduction of safety margins.	MAJ	C	10 ⁻⁵		

E.6 Autothrust Functions

Table E.8: Autothrust Functions FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Autothrust Engagement	Inability to engage autothrust	All	Slight reduction in operational capabilities.	MIN	D	10 ⁻³		
	No autothrust engagement annunciation	All	Pilot must determine whether autothrust is engaged by secondary means (aircraft behavior e.g.). Slight increase in crew workload.	MIN	D	10 ⁻³		
	Inadvertent autothrust engagement during autopilot approach, low thrust setting	Approach	Aircraft suddenly loses speed, pilot recognizes unintended thrust change and overpowers autothrust servo, might cause stall warning. Increase in crew workload and reduction in safety margins.	MAJ	C	10 ⁻⁵		Quantitative Fault Tree Analysis PSSA and SSA
	Inadvertent autothrust engagement during autopilot approach, high thrust setting	All	Aircraft speed suddenly increases, pilot recognizes unintended thrust change and overpowers autothrust servo, might cause go around.	MIN	D	10 ⁻³		
Autothrust Disengagement	Inability to disengage autothrust	All	Autothrust servo must be overpowered. Slight increase in crew workload.	MIN	D	10 ⁻³		
	Inadvertent disengagement of autothrust, with annunciation	All	Pilot recognizes autothrust disengagement, provides manual thrust control, slight increase in crew workload	MIN	D	10 ⁻³		
	Inadvertent disengagement of autothrust, without annunciation	All	Pilot does not immediately recognize autothrust disengagement, aircraft speed drifts away from reference, may cause overspeed or stall warning.	MAJ	C	10 ⁻⁵		
	No autothrust disengagement annunciation		Pilot must verify autothrust disengagement via secondary means	MIN	D	10 ⁻³		

E.7 Safety Functions

Table E.9: Safety Functions FHA work sheet.

Function	Failure Condition	Phase	Effect of Failure Condition on Aircraft/Crew	Class.	FDAL	Prob./FH	Reference to Supporting Material	Verification
Monitor External Guidance	Loss of monitoring of external guidance inputs	All	Inability to engage autopilot Slight increase in crew workload due to manual flight	MIN	D	10 ⁻³		Qualitative design appraisal
	Erroneous monitoring of external guidance inputs	All	Autopilot engaged with erroneous external guidance Flight crew recognizes unexpected aircraft behavior, e.g., deviation from intended flight path, and disconnects autopilot manually. Slight reduction in operational capabilities, slight increase in crew workload.	MIN	C	10 ⁻³	AC 23.1309-1E	
Monitor Sensor Inputs	Loss of monitoring of sensor inputs	All	Inability to engage autopilot Slight increase in crew workload due to manual flight	MIN	D	10 ⁻³		Qualitative design appraisal
	Erroneous monitoring of sensor inputs	All	Autopilot engaged with erroneous sensor inputs Flight crew recognizes unexpected aircraft behavior, e.g., deviation from intended flight path, or unexpected sensor information on MCMD, and disconnects autopilot manually. Slight reduction in operational capabilities, slight increase in crew workload.	MIN	C	10 ⁻³	AC 23.1309-1E	
Monitor Subsystem Health	Loss of monitoring of subsystem health status	All	Inability to engage autopilot Slight increase in crew workload due to manual flight	MIN	D	10 ⁻³		Qualitative design appraisal
	Erroneous monitoring of subsystem health status	All	Autopilot engaged with erroneous subsystem Flight crew recognizes unexpected aircraft behavior, e.g., deviation from intended flight	MIN	C	10 ⁻³	AC 23.1309-1E	

			<p>path, and disconnects autopilot manually.</p> <p>Slight reduction in operational capabilities, slight increase in crew workload.</p>					
Autopilot Automatic Disengagement	Inability to automatically disengage autopilot in the presence of autothrust failure	All	<p>Flight crew recognizes unexpected aircraft behavior, e.g., deviation from intended flight path, and disconnects autopilot manually.</p> <p>Slight reduction in operational capabilities, slight increase in crew workload.</p>	MIN	C	10 ⁻³	AC 23.1309-1E	
Autothrust Automatic Disengagement	Inability to automatically disengage autothrust in the presence of autothrust failure	All	<p>Flight crew recognizes unexpected aircraft behavior, e.g., unexpected acceleration or deceleration, and disconnects autothrust manually.</p> <p>Slight reduction in operational capabilities, slight increase in crew workload.</p>	MIN	C	10 ⁻³	AC 23.1309-1E	