



TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Engineering and Design

Automating the Transition of Lift-to-Cruise eVTOL Aircraft

Valentin Adamov Marvakov, M.Sc.

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: Prof. Dr.-Ing. Markus Ryll

Prüfer der Dissertation: 1. Prof. Dr.-Ing. Florian Holzapfel
2. Prof. Dr.-Ing. Stephan Myschik

Die Dissertation wurde am 24.02.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 06.06.2023 angenommen.

Abstract

This thesis provides novel solutions in the automation of lift-to-cruise aircraft. These types of vehicles are capable of powered-lift and wingborne flight and the envisioned mission profile involves the transition between the two modes while airborne. The methods that this thesis proposes address the automation of the transition process in the context of simplified vehicle operations.

Methods for both high- and low-level of automation control laws are developed. They include the derivation of procedures for the transition processes and the design solutions that enable them. The high-level of automation concept enables a fully automatic transition capability in the failure-free case. The design is resilient with regards to failures and in addition enforces safe system state at all times. The low-level of automation concept ensures full operator authority in all flight phases and robustness in the presence of failures. A high-degree of consistency in the operation with both levels of automation is enabled by the design and demonstrated in the thesis.

The developed solutions lean on the concepts of human-centered automation and are compliant with the currently available requirements imposed by the regulatory organs. The operator involvement in the transition process is considered in all aspects of the system operation by ensuring adequate and intuitive information supply between human and automation and vice versa. In addition, the pilot decision-making process in both nominal and abnormal scenarios is rendered non time-critical by procedure and automation design.

Analyses demonstrate the validity of the proposed solutions. In addition, a method is developed in this thesis, with which the automation functions can be modeled and tested in a time-efficient manner within the scope of the whole aircraft operation. The proposed solution is largely system architecture-agnostic and can therefore be applied in early stages of the product development cycle. Simulation results generated with the above-mentioned method serve as additional proof for the correctness and validity of the developed transition automation proposals.

Zusammenfassung

Diese Arbeit bietet neuartige Lösungen für die Automatisierung von lift-to-cruise Flugzeugen. Diese Fahrzeugtypen sind in der Lage, sowohl mit angetriebenem als auch mit voll aerodynamischem Auftrieb zu fliegen, und das vorgesehene Missionsprofil beinhaltet den Übergang zwischen den beiden Modi während des Fluges. Die in dieser Arbeit vorgeschlagenen Methoden befassen sich mit der Automatisierung des Übergangsprozesses im Rahmen eines vereinfachten Fahrzeugbetriebs.

Es werden Methoden für hoch- und niedrigstufige Automatisierungssteuergesetze entwickelt. Sie beinhalten die Ableitung von Prozeduren für die Übergangsprozesse und die Entwurfslösungen, die sie ermöglichen. Das Konzept des hohen Automatisierungsgrades ermöglicht eine vollautomatische Übergangsfähigkeit im störungsfreien Fall. Der Entwurf ist ausfallsicher und erzwingt darüber hinaus zu jedem Zeitpunkt einen sicheren Systemzustand. Das Low-Level-Automation-Konzept gewährleistet volle Bedienerautorität in allen Flugphasen und Robustheit im Falle von Ausfällen. Ein hoher Grad an Konsistenz im Betrieb mit beiden Automatisierungsgraden wird durch den Entwurf ermöglicht und in der Arbeit demonstriert.

Die entwickelten Lösungen lehnen sich an die Konzepte der menschenzentrierten Automatisierung an und entsprechen den aktuell verfügbaren Anforderungen der Regulierungsorgane. Die Einbeziehung des Bedieners in den Übergangsprozess wird in allen Aspekten des Systembetriebs berücksichtigt, indem eine adäquate und intuitive Informationsversorgung zwischen Mensch und Automatisierung und umgekehrt sichergestellt wird. Darüber hinaus wird der Entscheidungsprozess des Piloten sowohl in nominalen als auch in anormalen Szenarien durch die Gestaltung der Verfahren und der Automatisierung zeitlich unkritisch gemacht.

Analysen zeigen die Gültigkeit der vorgeschlagenen Lösungen. Darüber hinaus wird in dieser Arbeit eine Methode entwickelt, mit der die Automatisierungsfunktionen im Rahmen des gesamten Flugzeugbetriebs zeiteffizient modelliert und getestet werden können. Die vorgeschlagene Lösung ist weitgehend systemarchitekturunabhängig und kann daher in frühen Phasen des Produktentwicklungszyklus eingesetzt werden. Simulationsergebnisse, die mit der oben genannten Methode generiert wurden, dienen als zusätzlicher Nachweis für die Korrektheit und Gültigkeit der entwickelten Vorschläge zur Übergangsautomatisierung.

Contents

Abstract	i
Kurzfassung	iii
Table of Contents	ix
List of Figures	xi
List of Tables	xv
Acronyms	xvii
Symbols and Indices	xix
Symbols	xix
Indices	xx
1 Introduction	1
1.1 Motivation	3
1.1.1 Novel Aircraft Configurations	3
1.1.2 Shift in the Human Role	5
1.1.3 Novel Concept of Operations	6
1.1.4 Thesis Scope	6
1.2 State of the Art and Mission Statement	7
1.2.1 The Transition Process of Lift-to-Cruise eVTOL	8
1.2.2 Simplified Vehicle Operations	10
1.2.3 Regulatory Effort	12
1.2.4 Early-Stage Concept of Operations and its Validation	12
1.3 Contributions	15
1.4 Outline	16
2 Theoretical Background	19
2.1 Common Terminology	20
2.1.1 Characterization of System Design	20
2.1.2 Characterization of Functional Operation	20
2.2 The Automation of a Flight Control System	21

2.2.1	Automation Aspects and Challenges	22
2.2.2	Automation Design Principles	26
2.2.3	Automation Design Methods	29
2.2.4	Automation Implementation Methods	39
2.3	Structural and Performance Considerations of eVTOL Lift-to-Cruise Aircraft	44
2.3.1	The Properties of Distributed Hover Propulsion	44
2.3.2	The Properties of the High-Lift System	48
2.3.3	The Properties of the Traction System	49
2.4	Aspects of the eVTOL Aircraft Operation and Control Design	50
2.4.1	Simplified Vehicle Operations	51
2.4.2	Flight Phases of Lift-To-Cruise Vertical Take-Off and Landing (VTOL) Aircraft	57
2.4.3	Aircraft Control Inceptors, Discrete Inputs and Indications	58
2.5	Regulatory Efforts and Standards	63
2.5.1	Requirements on the Transition and Retransition	64
2.5.2	Vertical Take-off and Landing	65
2.5.3	Departure	65
2.5.4	Approach	67
3	High-Degree of Automation Transition and Retransition	69
3.1	Problem Description	70
3.1.1	Coherence and Supplementation of the Simplified Vehicle Operations Concept	71
3.1.2	Increasing the Robustness	72
3.1.3	Operator Support	74
3.2	Process Breakdown	74
3.2.1	Software Architecture and Function Assignment	76
3.3	Transition and Retransition Functional Flow	78
3.3.1	Normal Transition	78
3.3.2	Normal Retransition	81
3.3.3	Transition Mitigations	84
3.3.4	Retransition Mitigations	85
3.3.5	Procedure Summary	86
3.4	Automation Design	87
3.4.1	Automation Strategy Without a High-Lift System	88
3.4.2	Introducing the High-Lift System Automation	100
3.4.3	Supplementing the Powered-Lift Automation to Account for High- Lift System Operation	106
3.4.4	Decision-Execution	108
3.5	Design Analysis	117
3.5.1	Integrated System Behavior	118

3.5.2	What-If Analysis	125
3.6	Chapter Summary	128
4	Manual Transition and Retransition and the Industry Standard Compliance	131
4.1	Problem Description	133
4.1.1	Flight in the Presence of Faults	133
4.1.2	Facilitate Maximum Operator Authority	133
4.1.3	Simplicity	133
4.1.4	Adequate Operator Feedback and Reaction Times	134
4.1.5	Law Functional Decomposition	134
4.1.6	Harmonization of the Transition and Retransition Procedures . . .	134
4.1.7	Compliance with Industry Standards	135
4.2	Automation Design	135
4.2.1	Decision-Atomics	137
4.2.2	Decision-Making	139
4.2.3	Takeover State Evaluation	141
4.2.4	Decision-Execution	143
4.3	Design Analysis	146
4.3.1	Transition and Retransition Procedures with the Fallback System .	148
4.3.2	Fallback Throttle Mapping and Blending Requirements	157
4.3.3	State Allocation to Flight Phases	158
4.3.4	What-If Analysis	159
4.4	Nominal and Fallback System Integration in the Aircraft Operation	164
4.4.1	Control Inceptor Barrier Operation	165
4.4.2	Completing the Human-Machine-Interface Processing of both Nominal and Fallback System	167
4.4.3	Nominal and Fallback Transition and Retransition Comparison . . .	170
4.4.4	Takeover Correctness	181
4.4.5	Fitting the Transition and Retransition in the Special Condition for Vertical Take-Off and Landing Aircraft (SC-VTOL) Mission Profile	182
4.5	Chapter Summary	185
5	Operational Concept Validation During Early Development Stages	187
5.1	Method Description	190
5.1.1	Degree of Rapid Prototyping	191
5.1.2	Degree of System Architecture Independence	192
5.1.3	Simulation Capabilities and Tools	193
5.2	Data and Functional Management	196
5.2.1	Repository and File Structure	197
5.2.2	Change Management	199

5.3	Model Architecture	201
5.3.1	Utilization of Simulink Libraries	201
5.3.2	Error Detection	202
5.3.3	Integration Models	202
5.3.4	Vehicle Automation	203
5.4	Haptic Feedback Automation and Pilot Input Processing	207
5.5	Nominal Automation	211
5.5.1	Powered-Lift System Operation	211
5.5.2	Powered-Lift Mode Selection of the Nominal System Law	217
5.5.3	High-Lift System Operation	217
5.6	Fallback Automation	221
5.6.1	Takeover Starting State Calculation	223
5.6.2	Decision-Atomics	224
5.6.3	Decision-Making	225
5.7	Conclusion	229
6	Simulation Results	235
6.1	Fault-Free Nominal System Transition and Retransition	237
6.1.1	Transition	238
6.1.2	Retransition	242
6.1.3	Takeover State Evaluation During Nominal System Operation	249
6.2	Fault-Free Fallback System Transition and Retransition	251
6.3	Abnormal Scenarios	256
6.3.1	Retransition Mitigation - Reversion to Wingborne Flight	257
6.3.2	Retransition - Confirming the Entry to Powered-Lift Flight	262
6.4	Conclusion	265
7	Conclusion	267
7.1	Completed Topics	270
7.2	Ongoing Efforts	274
7.3	Outlook and Perspectives	276
A	Relationship Between Transition Sets, Conditions and Functions	I
B	High-Automation Transition, Retransition and Mitigation Flow	III
C	High-Automation Flight Failure Mode and Effects Analysis	V
D	How Variable Starting States can be Achieved	XV
E	Fallback Transition, Retransition and Mitigation Flow	XVII
F	Behavioral Specification Model Implementation Examples	XIX

G TUM Institute of Flight System Dynamics (TUM) Simulator Pilot Checklist	XXV
H Additional Simulation Results	XLVII
H.1 Transition Mitigation - Reversion to Powered-Lift Flight	XLVII
H.2 Transition - Prolonged High-Speed Flight	LI

List of Figures

- 1.1 Airplane Evolution. **Upper Left:** Otto Doppeldecker at Schleiheim in 1913. Source: [1]. **Upper Right:** Douglas DC-3 at the Technik Museum Speyer. Source: [2]. **Bottom:** Airbus A350 at Munich Airport. Source: [3]. 2
- 1.2 Different Electrical Vertical Take-Off and Landing (eVTOL) Airframes. **Upper Left:** VoloCity by Volocopter [4] - A Multirotor Configuration. **Upper Right:** CityAirbus by Airbus [5] - A Multirotor Configuration. **Lower Left:** S4 Air Taxi 2.0 by Joby Aviation [6] - A Tilt-Rotor Configuration. **Lower Right:** VoloConnect by Volocopter [7] - A Lift-to-Cruise Configuration with Dedicated Traction System. 4
- 1.3 Exemplary eVTOL aircraft. The drawing of the vehicle body and vertical propulsion system was inspired by [4]. 7
- 2.1 Automation Design Considerations and Aspects 23
- 2.2 Graphical Representation of the Combinational Logic (Left) and the Sequential Logic (Right) 29
- 2.3 A Latch as a Mealy Machine 37
- 2.4 Automation Functional Layout 38
- 2.5 Decision-Atomics - Implementation Example 41
- 2.6 Decision-Making - Implementation Example 42
- 2.7 Decision-Execution - Implementation Example 43
- 2.8 A Lift/Thrust Unit in an Arbitrary Freestream 46
- 2.9 Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics (TUM-FSD) Command Variables in the different flight phases. The Image Stems From [8]. 53
- 2.10 Lift-to-Cruise Aircraft Flight Phase Allocation with Relation to the Speed. The Region Sizes are Chosen for Better Visibility and Need Not be to Scale. **Top:** The Region Allocation for Constellations Without High-Lift Systems. **Bottom:** The Region Allocation for Constellations with High-Lift Systems. 59
- 2.11 The Pilot Control Inceptors. **Left:** Throttle Stick. The Up/Down Movement is Responsible for Longitudinal Control. The Left/Right Movement in the Hover and Transition regions is Responsible for Lateral Control. **Right:** Climb Stick. The Up/Down Movement is Responsible for Vertical Control. The Left/Right Movement is Responsible for Directional Control. 61

LIST OF FIGURES

2.12	Transition and Retransition Procedure Indication Item. The Image Stems From [9].	63
2.13	The Take-Off and Landing Reference Volume as Found in [10]	65
2.14	The Take-off Trajectories as Found in [10]	66
2.15	The Approach Trajectory Including a Rejected Landing as Found in [10].	68
3.1	High-Degree of Automation Software Modules, involved in the Transition and Retransition. The list is not exhaustive with regards to actual implementation.	76
3.2	Transition Process Flowchart	79
3.3	Retransition Process Flowchart	82
3.4	Transition Abnormal Procedure Flowchart	84
3.5	Retransition Abnormal Procedure Flowchart	85
3.6	High-Degree of Automation Lift/Thrust Unit (LTU) State Machine	89
3.7	High-Degree of Automation High-Lift System State Machine	101
3.8	Lift to Cruise Aircraft Flight Phase Allocation with Relation to the Airspeed. This Figure Supplements Figure 2.10 with All Airspeed Values, Used by the Automation. The Region Sizes are Chosen for Better Visibility and Need Not be to Scale.	122
4.1	State Machine for the Fallback System Control Mode Selection	136
4.2	Transition Process Flowchart	151
4.3	Retransition Process Flowchart	153
4.4	Transition Mitigation Process Flowchart	154
4.5	Retransition Mitigation Process Flowchart	155
5.1	V-Model as Found in [11]	188
5.2	Repository and File Structure of the Exemplary Behavioral Specification	197
5.3	Development Cycle for Functions that Tackle System Interactions	200
5.4	Overview of the Model Architecture	201
5.5	Behavioral Specification Vehicle Automation Architecture	204
5.6	Barrier Behavior	208
5.7	Pilot Input Processing	210
5.8	Processing of the Human-Machine-Interface	211
5.9	Processing of the Airdata	212
5.10	Processing of the LTU Feedback	213
5.11	Powered-Lift Automation Provisions for High-Lift System Operation	214
5.12	M_{LTU} Implementation	215
5.13	Powered-Lift Mode Selection	216
5.14	The Decision-Atomics of the High-Lift System Automation	218
5.15	M_{HL} Implementation	219

5.16	High-Lift System Scheduling over the Airspeed and Automation Mode . . .	220
5.17	<i>Initialize</i> Function Implementation	222
5.18	Fallback System Control Mode Selection Decision-Atomics	224
5.19	The Multi-Level Finite State Machine Architecture that Implements M_{FB}	226
5.20	Fallback Automation State Machine First Level	227
5.21	Fallback Automation State Machine Second Level	228
5.22	The Development Stages of the eVTOL Simulator of TUM-FSD	233
6.1	Mission Profile	236
6.2	Nominal System Powered-Lift Management Simulation Results During Transition	239
6.3	Nominal System High-Lift Management Simulation Results During Transition	241
6.4	Nominal System Airspeed Protection Management Simulation Results Dur- ing Transition	243
6.5	Nominal System High-Lift Management Simulation Results During Retran- sition	244
6.6	Nominal System Powered-Lift Management Simulation Results During Retransition	246
6.7	Nominal System Airspeed Protection Management Simulation Results Dur- ing Retransition	248
6.8	Results of the Fallback System Initial State Evaluation Following a Po- tential Takeover During Nominal System Flight. Left: Evaluation During Transition. Right: Evaluation During Retransition	250
6.9	Fallback System Automation Operation During Transition	252
6.10	Fallback System Automation Operation During Retransition	255
6.11	Nominal System Retransition Mitigation to Wingborne Flight	258
6.12	Nominal System Airspeed Protection Operation During Retransition Miti- gation to Wingborne Flight	260
6.13	Fallback System Retransition Mitigation to Wingborne Flight	261
6.14	Nominal System Abnormal Entry into Powered-Lift Flight	263
6.15	Nominal System Airspeed Protection Management During Abnormal Entry into Powered-Lift Flight	264
7.1	Nominal System Ground/Airborne Mode Provision	271
A.1	State Machine and Chart Relationship	II
B.1	Complete Transition and Retransition Process Flow. The Chart Assumes no Crew Deviations.	IV
D.1	Example of How Different Starting States can be Achieved	XVI

LIST OF FIGURES

E.1 Complete Transition and Retransition Process Flow. The Chart Assumes no Crew Deviations. XVIII

F.1 Error Detection XX

F.2 Law Integration Model XXI

F.3 Automation Integration Model XXII

F.4 Nominal Design Reference Model (DRM) Integration Model: The Error Injections and Interface Modules are Colored in Orange XXIII

F.5 Nominal System Automation: The Decision-Making Module is in an Enabled Subsystem XXIV

H.1 Nominal System Transition Mitigation to Powered-Lift Flight XLVIII

H.2 Nominal System Airspeed Protection Operation During Transition Mitigation to Powered-Lift Flight L

H.3 Nominal System Abnormal Entry into High Dynamic Pressures with a Partially Disengaged Powered-Lift System LII

H.4 Nominal System Airspeed Protection Management During Abnormal Entry into High Dynamic Pressures with a Partially Disengaged Powered-Lift System LIII

List of Tables

- 1.1 Objective-Contribution and Chapter-Contribution Traceability Matrix . . . 17
- 2.1 List of Relational Operators 31
- 2.2 Boolean Algebra: Basic Operators 32
- 2.3 Example of an Exhaustive Truth Table 32
- 2.4 Example of a Simplified Truth Table 32
- 2.5 Truth Table of a Latch if the Memory is Treated as an Input 34
- 2.6 Fallback Command Variables in the Different Flight Phases 56
- 2.7 Overview of the Law Sensor Dependency 57
- 3.1 Sections of the High-Degree of Automation Control Concept FHA that
Implicates Requirements on the Automation 73
- 3.2 Summary of the M_{LTU} Inputs 90
- 3.3 Summary of the M_{HL} Inputs 102
- 3.4 M_{LTU} Input Supplement for High-Lift System Operation 106
- 3.5 Underspeed Protection Limit Truth Table 110
- 3.6 Overspeed Protection Limit Truth Table 111
- 3.7 High-Degree of Automation Transition Indication Item Truth Table 114
- 3.8 Indication Item Causal Behavior during Transition. The Color Patterns
Themselves are Not in the Scope of this Thesis and Can be Found in [9] . . 115
- 3.9 Indication Item Causal Behavior during Retransition. The Color Patterns
Themselves are Not in the Scope of This Thesis and Can be Found in [9] . 116
- 3.10 State Machine State to Flight Phase Allocation 123
- 4.1 State Compatibility Matrix of M_{FB} 136
- 4.2 Summary of the M_{LTU} Inputs 137
- 4.3 Takeover Function Truth Table 142
- 4.4 Control Allocation Action Truth Table 144
- 4.5 Fallback System Transition Indication Item Truth Table 147
- 4.6 State Machine State to Flight Phase Allocation 159
- 4.7 Barrier Status Truth Table 167
- 4.8 Comparison: Normal Transition Procedure - Fault-free Case 172
- 4.9 Comparison: Normal Retransition Procedure - Fault-free Case 174
- 4.10 Comparison: Abnormal Transition Procedure - Powered-Lift Flight Reversion 176

LIST OF TABLES

- 4.11 Comparison: Abnormal Transition Procedure - Reconfiguration to Fixed-Wing Mode 177
- 4.12 Comparison: Abnormal Transition Procedure - Entry to Higher Airspeed . 178
- 4.13 Comparison: Abnormal Retransition Procedure - Reversion to Wingborne Flight 179
- 4.14 Comparison: Abnormal Retransition Procedure - Confirm Powered-lift Flight 180

- 5.1 Overview of Behavioral Specification Model Method Abstraction 194

- C.1 High-Degree of Automation Control Concept Exhaustive Failure-Modes and Effects Analysis VI

Acronyms

AS94900A	Vehicle Management Systems - Flight Control Function, Design, Installation and Test of Piloted Military Aircraft, General Specification
Back-EMF	Counter-Electromotive Force
BIT	Built-In Test
CFP	Critical Failure of Performance
CFR	Code of Federal Regulations
ConOps	Concept of Operations
DRM	Design Reference Model
EASA	European Aviation Safety Agency
eVTOL	Electrical Vertical Take-Off and Landing
FCS	Flight Control System
FSD-SVO	Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics
FTO	Final Take-Off Speed
HMI	Human-Machine-Interface
INDI	Incremental Nonlinear Dynamic Inversion
LDP	Landing Decision Point
LTU	Lift/Thrust Unit
MOC SC-VTOL	Means of Compliance with the Special Condition VTOL
RPM	Revolutions per Minute
SC-VTOL	Special Condition for Vertical Take-Off and Landing Aircraft
SVO	Simplified Vehicle Operations
TDP	Take-Off Decision Point
TOSS	Take-Off Safety Speed
TTU	Traction Thrust Unit
TUM	Technical University of Munich
TUM-FSD	TUM Institute of Flight System Dynamics
UAM	Urban Air Mobility
VTOL	Vertical Take-Off and Landing

Symbols and Indices

Symbols

M	State Machine
S	Finite State Set
V	Velocity
χ	Blending Variable
δ_T	Throttle Inceptor Deflection
δ_{F_i}	Flap Deflection
$\delta_{T,D}$	Throttle Inceptor Detent Position
$\delta_{T,G}$	Throttle Inceptor Gate Position
δ	Transition Function
λ	Blending Variable
\mathbb{B}	Boolean Set
\mathbb{H}	Throttle Inceptor Hover Region
\mathbb{L}	Throttle Inceptor Gate Left Region
\mathbb{R}	Throttle Inceptor Gate Right Region
\mathbb{T}	Throttle Inceptor Transition Region
\mathbb{W}	Throttle Inceptor Wingborne Region
ω	LTU revolution rate
V_2	Fixed-Wing Takeoff Safety Speed
V_X	Speed for Best Climb
V_{CAS}	Calibrated Airspeed
V_{FE}	Maximum Flap Extended Speed
V_{FTO}	Final Take-Off Speed
V_{HOVER}	Maximum Hover Speed
V_{LSNE}	Lift-System Operation Never Exceed Speed
V_{MC}	Minimum Control Speed with the Critical Engine Inoperative
V_{NE}	Never Exceed Speed
$V_{OEI_{FE}}$	FSD-SVO Safe Speed with the Critical Engine Inoperative with Fully Deployed Flaps
V_{OEI}	FSD-SVO Safe Speed with the Critical Engine Inoperative in the Clean Configuration

V_{REF}	Landing Reference Speed
$V_{SAFE_{FE}}$	FSD-SVO Nominal Safe Speed with Fully Deployed Flaps
V_{SAFE}	FSD-SVO Nominal Safe Speed in the Clean Configuration
$V_{STALL_{FE}}$	Stall Speed with Fully Deployed Flaps
V_{STALL}	Stall Speed in the Clean Configuration
V_{TOSS}	Powered-Lift Take-Off Safety Speed
h_2	High Hover Height
h	Output Function
s	State Machine State
t	Transition Condition

Indices

0	Denotes Starting States
<i>FB</i>	Denotes Quantities, associated with the Fallback Automation
<i>HL</i>	Denotes Quantities, associated with the High-Lift Scheduling of the High-Degree of Automation
<i>HS</i>	Denotes Quantities, associated with the High-Speed Evaluation of the High-Degree of Automation
<i>LTU</i>	Denotes Quantities, associated with the Powered-Lift Scheduling of the High-Degree of Automation

Chapter 1

Introduction

On December seventeenth 1903, brothers and pioneers Orville and Wilbur Wright achieved the first manned flight with a powered airplane, flying for approximately twelve seconds and reaching a distance of less than fifty meters. Looking back at this feat, the advancement of the aviation industry in just one century appears inconceivable.

Today, many see aviation as a form of service. For the average person traveling from point A to point B with an airline is affordable, comfortable and one of the safest means of transportation. Knowing that the first powered manned flight was only in the beginning of the 20th century, it is staggering how fast pioneers in engineering and piloting have advanced the state of technology that we know today.

The drive of man to push the limits of what is deemed possible is immense. Aviation is no exception in this regard. History has recorded the feats of the Wright brothers (first powered flight), John Alcock and Arthur Brown (first transatlantic flight without a stop-over), Chuck Yeager (first supersonic flight) and many others. The efforts of pilots throughout this and the last century have been instrumental. Their strive to push the boundaries of the available technology have surely helped advance its innovation and bring it to the state as we know it today.

This progress is depicted in Figure 1.1, which shows how the airplanes have changed over time due to the technological advancements. There are multiple fields that have largely contributed to these developments. Among others, discoveries in the field of material sciences have produced structures that are sturdier, lighter and more durable. Furthermore, gaining understanding in advanced aerodynamics and the invention of the jet engine have allowed us to fly faster, longer and higher.

Yet, there are also consequences as a result of these advancements. Longer flights put higher pressure on the pilots due to the required constant operation. Additionally, flying at high altitudes and low air densities lowers the aerodynamic damping. Such reduction in aircraft stability and control and the risks of pilot fatigue began to play a bigger role.

These problems have been answered with the invention of the transistor and the integrated circuitry. It has been a building block of innovations that enable us to navigate more precisely and communicate over large distances. It has further allowed to transform



Figure 1.1: Airplane Evolution. *Upper Left: Otto Doppeldecker at Schleißheim in 1913. Source: [1]. Upper Right: Douglas DC-3 at the Technik Museum Speyer. Source: [2]. Bottom: Airbus A350 at Munich Airport. Source: [3].*

the operation of the aircraft from a problem that is purely mechanical and involves the complete operator attention to an electro-mechanical one. In this way more and more pilot tasks have been automated and the autopilots emerged. Today's systems allow for automatic following of waypoints and landings, can optimize the fuel consumption and more.

Another issue is that the push for advancement throughout the years has led to casualties. This was not left unnoticed and caused the emergence of aviation authorities which - since their inception - have guided and supervised the innovations, making sure that lives will not be endangered. Nowadays these authorities are involved in every step of the development, strictly ensuring the system's integrity. This has led the aviation industry to be often seen as the safest form of transportation [12].

We are right now in an age of digitalization. Every year, smaller but yet more powerful processors emerge. Climate change has pushed the need for green energy. Battery efficiency and power density are improving at a staggering pace. From this, a new phase in the history of aviation is emerging. One, in which private individual will be able to afford travel with an electrically-powered, fully automated aircraft taxi-service.

Urban Air Mobility (UAM) is the umbrella term of this new phase in the aviation industry. UAM is novel and unexplored, and as such carries its own hazards and challenges. It requires new solutions to the new problems it imposes. Thus, it requires man to once again push the limits of the state of technology.

1.1 Motivation

The UAM concept emerged organically over the last decade with the advancement of distributed propulsion and battery technology. People began flying small manned multirotor aircraft. To the author’s knowledge, the first official manned flight with such a vehicle was by Volocopter [13]. As more and more airframes emerged, a concept, referred to as “On-Demand Mobility” was introduced in Uber’s white paper [14] in 2016. Therein, the authors estimate that traveling with a VTOL aircraft as opposed to a land vehicle will significantly reduce commute time. With the advancement of the technology this could even occur at a comparative cost. The examples where this holds are in heavily populated areas, such as São Paulo. The white paper acknowledged the possibility to use this novel technology to solve a pressing problem.

Jumping to 2020, the analysis of [15] values the UAM market to 2.90 billion USD. This aligns with the report of [16], where in 2021 value is estimated at 3.10 billion USD. Both reports project a steady market size increase in the next ten years. This is an indication as to how much resource, effort and attention has been put on this new market niche as well as of its potential to change the industry landscape. In this section, the key drivers and challenges of the technology that is being developed are examined.

1.1.1 Novel Aircraft Configurations

The Urban Air Mobility concept does not prescribe the energy source of the aircraft. However, climate change is arguably one of the most relevant problems of this century and it has played a key role in influencing the technology utilized. In their “Green Deal”, the European Union is implementing an initiative to drastically reduce the carbon emissions of its members [17]. The United States are not far behind [18]. The importance of emission reduction is recognized world-wide.

This has led the future players in the UAM sector to respond accordingly and pursue solutions that beneficially impact the carbon footprint. Nearly all UAM key competitors as of now - Volocopter, Lilium, Joby Aviation, Archer, etc. - have opted for electrically-powered platforms. This shift from the conventional fossil fuel has produced unconventional eVTOL airframes, which are explored here.

The new eVTOL industry has inspired novel aircraft configurations. Figure 1.2 shows how vastly the airframes differ. Nearly all eVTOL platforms have distributed hover propulsion systems. Loosely, the configurations can be grouped into three distinct categories based on the method of operation of the propulsion system.

The first category is the multirotor aircraft. Examples of such configurations can be seen in the upper half of Figure 1.2 and include the VoloCity [4] and the CityAirbus [5]. The key characteristic of the multirotor platforms is that the lift is generated by the propulsion system throughout the whole flight.



Figure 1.2: *Different eVTOL Airframes. **Upper Left:** Volocity by Volocopter [4] - A Multirotor Configuration. **Upper Right:** CityAirbus by Airbus [5] - A Multirotor Configuration. **Lower Left:** S4 Air Taxi 2.0 by Joby Aviation [6] - A Tilt-Rotor Configuration. **Lower Right:** VoloConnect by Volocopter [7] - A Lift-to-Cruise Configuration with Dedicated Traction System.*

The distinct feature of the next two aircraft types is that they are not only capable of vertical take-off and landing, but can also fly with fixed-wing aerodynamic lift. Examples of these airframes can be found in the lower half of Figure 1.2. The two categories differ in the way the forward thrust is generated. This is done either by thrust vectoring or by a dedicated traction system. In Figure 1.2 the lower left image is an example of the former, whereas the lower right - of the latter. The former is commonly referred to as Tilt-Rotor and the latter is referred to as Lift-to-Cruise.

The majority of the mission of both tilt-rotor and lift-to-cruise aircraft is in wingborne flight, where - by definition - lift is generated by aerodynamic surfaces. Thus, they offer an increased flight time and range when compared to their multirotor counterparts. This property has made these two configurations more popular among eVTOL designers. According to the authors of [19], at the time of publication well over sixty percent of the eVTOL vehicles designed are capable of wingborne flight despite being significantly more complex than multirotor aircraft.

In order to achieve wingborne flight after a vertical take-off, tilt-rotor and lift-to-cruise aircraft must accelerate while airborne. This contrasts conventional fixed-wing airplanes, where this occurs on a runway. Thus, these novel aircraft configurations require the joining of the hover and fixed-wing flight envelopes - something hardly explored prior to their emergence.

1.1.2 Shift in the Human Role

The UAM concept envisions a large number of aircraft operating simultaneously. The substantially large number of people operating the vehicles poses the question whether the level of training can be reduced. Reducing the costs of training while maintaining qualified personnel is a challenge by itself.

Electrification partially mitigates this concern. In [20], the authors analyze the number of tasks the crew has to deal with during flight. They found that the utilization of eVTOL platforms will reduce the required operative knowledge by roughly forty percent. This is mainly due to the reduced system complexity. A fully electric aircraft does not require sophisticated hydraulic and fuel systems and thus operator qualification in such topics is not necessary. This in turn reduces the cost of training.

It is rather the necessary level of automation that imposes a paradigm shift as to the role of the human onboard eVTOL aircraft. UAM envisions that a flight from point A to point B would be highly or fully automated. Thus, the human is no longer seen as the operator of the machine, but more as its supervisor. This reduces the capabilities and knowledge required and as a result also the training needs.

The change in the human role not only requires robust algorithms but also a change in the way mitigation strategies are designed. Commonly, the crew was seen as the last line of defense, which evidently will no longer be possible due to their reduced capabilities.

1.1.3 Novel Concept of Operations

The Aircraft Concept of Operations (ConOps) is a type of specification document, which sketches the execution of a mission with a given aircraft [21] and describes how the machine should be operated both on ground and in-flight. The ConOps defines mitigation and contingency strategies, takes into account and ensures compliance with regulations in all modes of operation.

During the prototype stages of a highly automated operation, errors in the design occur, which impose the need for clearly defined mitigation functions. By implication, the actions to be taken in an off-nominal scenario need to be reflected in the ConOps of the prototype aircraft.

The importance of well-written and valid requirements is explored well in systems engineering. In [22] it is illustrated how this initial phase lays the foundation for all future development and hence has the highest influence on the accumulated costs during the product life-cycle.

The ConOps is an input for many requirement breakdown processes and is therefore a big cost-driver. A ConOps for an eVTOL aircraft configuration is hardly trivial to create due to the novelty of the vehicle. However, a ConOps that accounts for all scenarios during aircraft operation is of high economic importance for the developers of the innovative technology.

1.1.4 Thesis Scope

This thesis explores the challenges addressed in the previous sections - the transition from powered lift-flight to wingborne flight and back of lift-to-cruise aircraft as well as the reconfiguration of the automation with the expected shift in human role. Given the novelty of both vehicle and automation as explained in Section 1.1.3, the thesis also aims to provide means of validating the novel ConOps proposals with regard to the in-air operation.

This section specifies the circumstances, under which this thesis was inspired. In an industry funded project, the TUM-FSD is involved in the development of an eVTOL technology demonstrator. The vehicle is subject to non-disclosure, but its main features can be seen in Figure 1.3.

The manned aircraft is an electrically powered lift-to-cruise vehicle and is therefore capable of wingborne flight. The design is no-single point of failure. The forward force is generated by a dedicated traction system, composed of two pusher rotors. In addition, the vehicle has a high-lift system. The Flight Control System (FCS) has a redundant architecture.

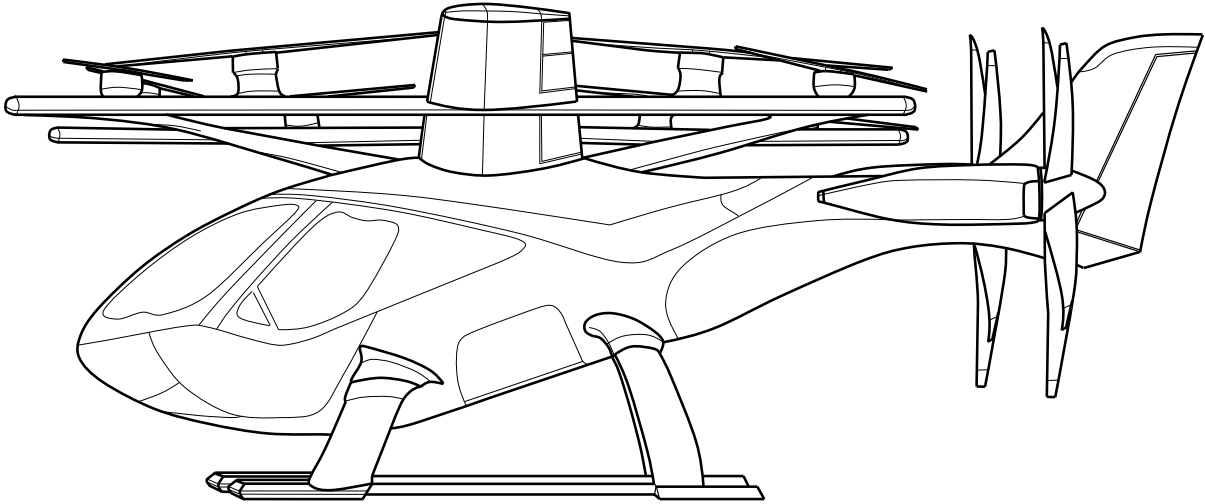


Figure 1.3: *Exemplary eVTOL aircraft. The drawing of the vehicle body and vertical propulsion system was inspired by [4].*

In the project, TUM-FSD is responsible for the complete high-level functional development of the FCS of the prototype aircraft. Thus, this thesis focuses on the above-mentioned manned lift-to-cruise eVTOL aircraft. The developed methods are applied to the project vehicle but are not confined to this airframe. Instead, they can be applied to aircraft with similar configurations.

1.2 State of the Art and Mission Statement

In this section the state of technology in selected topics of the eVTOL development is analyzed. The focus is on the aspects of flight control automation and its verification methods.

This section is structured as follows. It begins with Section 1.2.1, where the reconfiguration process from powered-lift flight to wingborne flight and back is researched. This is followed by Section 1.2.2, in which the same process is analyzed from the perspective of the shifting role of the operator from pilot to supervisor. In Section 1.2.3 the state of technology with relation to the regulatory framework is examined. The state of the art analysis ends with Section 1.2.4, where methods of early validation of novel automation functions and their application to prototype airframes are researched.

In each section gaps in the state of technology are identified. They are subsequently summarized as objectives that this thesis must address and provide solutions for. The objectives are the basis of the contributions of this dissertation, found later in Section 1.3.

1.2.1 The Transition Process of Lift-to-Cruise eVTOL

Fixed-wing VTOL aircraft have been around since the 1960s. Examples of such VTOL airframes include fighter jets such as the Harrier and the F-35 Lightning II or tilt-rotors such as the V-22 Osprey and Leonardo's AW609. eVTOL configurations, capable of wingborne flight gained significant popularity only after the emergence of the UAM concept. Thus, the automation of the transition between the powered-lift and wingborne operation of such aircraft is still a widely unexplored topic.

The majority of publicly available information focuses on the closed-loop control problem during transition between the two flight phases. Arguably having robust algorithms for guaranteeing stability throughout the transition and retransition is a highly-critical task. All found control concepts require some automation function to either determine or command the aircraft configuration state. This automation function is directly or indirectly responsible for the scheduling of the control law. The existing automation mechanisms are summarized below.

For tilt-rotor aircraft, the state of reconfiguration is typically determined by the tilt deflections. The tilt angles are commonly scheduled in an open-loop manner over the vehicle airspeed. In the available literature, the mapping between airspeed and deflection angle is first determined via analytical computations and subsequently either fine-tuned or validated in flight [23]. This type of scheduling is referred to as "Tilt Corridor" [24]. The position within the corridor is utilized for scheduling of the control laws, where the exact strategy depends on the control architecture. In [25], the author uses the tilt corridor to blend between two separate controllers used for hover and wingborne flight respectively. The findings of [24] indicate that for other applications the gains of the control law are scheduled instead. Therefore, it can be derived that the automation of the transition and retransition for tilt-rotor aircraft consists of deflecting the nacelles within the specified tilt corridor and thereby also scheduling the law accordingly.

The airspeed or the estimation of the airspeed is relevant for the automation of fixed-wing VTOL configurations with dedicated traction systems as well. The author of [26] uses the airspeed to select between three distinct control strategies - "VTOL", "Transition" and "FW mode" [26]. The former and the latter have their own control laws, whereas the transition control mode utilizes both. The choice of mode stems from trim point calculations, where the intersecting trim points between hover and fixed-wing mode are attributed to the transition phase. In [27], the authors present three mechanisms on aircraft-level, with which the transition and retransition can be performed at a given airspeed. However the exact algorithms for automation are not mentioned.

Although the patent [28] analyzes the transition and retransition from the perspective of a control task, it also provides a comprehensive description of the underlying automation functions. The law consists of two control elements and the strategy involves a blending of the control elements. Control volume and switching is managed over the aircraft configuration estimate. This is done by filtering the pilot forward speed command, where

the parameters of the low-pass filter are tuned based on simulation and flight test data to account for the dynamics of the aircraft. Thereby, the applicants claim to estimate the aircraft forward speed. This fictive airspeed stems from the operator input and as a consequence the pilot intention with relation to the desired mode of operation is taken into account. Thus, a reconfiguration only takes place once requested by the crew. The filter output is used within a State Machine to specify the aircraft mode of operation, where two modes are foreseen. The reconfiguration is explained to occur in the second mode, where the engagement and disengagement of the hover propulsion system is decided based on whether or not it needs to be utilized by the hover control element.

During the analysis of the state of the art in the transition automation, several challenges were identified. Firstly, though robustness measures are addressed by some (for example in [25, 28]), no methods for managing components failures are discussed. From here the first objective of this thesis is formulated as follows:

Objective 1 Provide transition and retransition automation functions for lift-to-cruise eVTOL that account for possible component malfunctions.

In addition, apart from [28], in all automation strategies analyzed above, the role of the human within the operation of the system is not taken into account. For example, in the solutions, a short disturbance may affect the airspeed such that a mode switch is triggered. This may cause an unwanted short-term activation or deactivation of the hover propulsion system without any change in the input from the crew. This irregularity may cause mode confusion and thus according to [29] is a common automation fallacy called “opacity”.¹ A reconfiguration must occur when explicitly desired by the crew and this state must persist until commanded otherwise.

Even though [28] attempts to address the above-mentioned problem, issues with relation to the human-machine interaction persist there as well. In the described control volume scheduling there is no method to prohibit the inadvertent activation of the hover propulsion system. This can occur if the tracking error of the hover control element causes it to exceed the predefined threshold and thus activate the hover propulsion units.

Another important topic with regards to automation and the human-machine interface is the information supply to the crew. This is not discussed in the publicly available literature. With these points in mind, the next objective of the thesis is summarized as follows:

Objective 2 Provide a human-centered transition and retransition automation concept for lift-to-cruise eVTOL. It must:

Objective 2.1 Provide reconfiguration only when requested by the operator.

Objective 2.2 Prohibit uncommanded reconfiguration changes.

¹In addition, such an event may cause structural damage in certain flight conditions. The origins of the this property is explained in later chapters.

Objective 2.3 Provide adequate feedback and thus situational awareness during the reconfiguration process.

Objective 2.4 In accordance with **Objective 1**, account for and provide sufficient time for operator decisions in the event of component malfunctions that require crew actions during the reconfiguration.

Objective 2.1, **Objective 2.2** and **Objective 2.3** are necessary to address the shortcomings in the general case. **Objective 2.4** is a direct resultant of **Objective 1** and is required to increase the automation robustness and be human-centered in the resulting off-nominal scenarios. Arguably, there are certain aspects of **Objective 2.3** that are derived from **Objective 1**.

1.2.2 Simplified Vehicle Operations

To the author's knowledge, the necessity to alleviate pilot workload of fixed-wing VTOL aircraft was first addressed in [30]. There, the authors propose an integrated control design for a YAV-8B Harrier. This concept provides a control augmentation which simplifies the aircraft operation because the operator could command translational velocities in hover. The transition is still manual but the pilot workload is alleviated. Overall, the approach requires five input axes and still relies on significant pilot training and experience. However, this is perhaps one of the first attempts at what is now known as Simplified Vehicle Operations (SVO) for fixed-wing VTOL aircraft.

Simplified Vehicle Operations can be best described as a concept that provides straightforward aircraft handling and significant reduction of necessary pilot training and level of expertise [31]. It does this on the one hand by automating many of the manual tasks which are otherwise handled by the operator [20]. On the other hand, it introduces protective functions such as envelope protections in order to raise the overall system safety that was reduced by the lack of the well-prepared operator. In terms of technological maturity, the presenters of [31] differentiate between three phases of SVO. For the sake of completeness, those are summarized here:

SVO1: Users are the current generation of pilots. The controls are “unified” [31], i.e. are identical but their interpretation can vary over the different flight phases. It includes very little task automation and the operator is still the last line of defense.

SVO2: Users are operators with a significantly reduced level of training. The controls and their interpretation are identical in all flight phases. Many tasks are automated but the system still requires manual input.

SVO3: Users are people with no piloting skills. The control input is necessary only to specify the desired landing location. The whole system operation is fully automated.

Examples of SVO1 include [32] and [19]. In addition, the author of [19] presents an SVO concept, referred to as “EZ Fly” which is acknowledged to be SVO2 in [31]. This, however, is for a multirotor eVTOL.

The most comprehensive publications of SVO concepts for lift-to-cruise eVTOL found are those of the TUM Institute of Flight System Dynamics. In [33], the author introduces a novel inceptor, specifically tailored for SVO. It aids in the operator situational awareness by design. The author of [34] demonstrates an SVO control concept that utilizes the inceptor of [33]. It provides a dynamic mapping of the inceptors to command variables that are dependent on the aircraft state. Thus, adequate and intuitive handling is ensured throughout the whole envelope. In continuation, in [8] the control concept is much more exhaustively described and the envelope protections that maintain the system integrity throughout the aircraft flight phases are introduced. Hence, the resulting product best fits into the definitions of SVO2. It is from here-on referred to as FSD-SVO.

The FSD-SVO does not provide the exact automation mechanisms of transition and retransition. However, in [8] the author mentions that such algorithms are necessary. They must specify the law’s mode of operation, which is responsible for command variable scheduling. In addition, the automation is required to dictate the allowed usage of the system effectors. Those are a function of the aircraft state of reconfiguration. Third, the automation must provide the necessary information for certain envelope protection scheduling. Lastly, FSD-SVO does not mention operational procedures. Even though being robust against effector malfunctions, FSD-SVO does not consider possible procedural changes in the transition and retransition due to those malfunctions.

It is from here that the next objective is derived. The results of this thesis have to be compatible with FSD-SVO and address the missing functionality summarized in the paragraph above. This on the one hand implies that the outcome of this thesis has to provide all necessary information to the control concept. Thereby it should not negatively impact any of the properties of the concept. On the other hand, during nominal scenarios the automation should not introduce overhead that would increase the required operator training. In the off-nominal scenarios, a simplistic plan of operation must be derived so as to adhere to the SVO philosophy. The resulting objective is defined as follows:

Objective 3 Provide procedures and automation functions for lift-to-cruise eVTOL that are compatible with the FSD-SVO. They must:

Objective 3.1 Fit into the proposed FSD-SVO.

Objective 3.2 Not add operational complexity in nominal operation.

Objective 3.3 Enforce safety-constraints and thus increase the overall system robustness and resilience.

Objective 3.4 Provide for an intuitive operator input in the event of component malfunctions during transition and retransition that require crew actions.

1.2.3 Regulatory Effort

Over the recent years, the European Aviation Safety Agency (EASA) has produced a regulatory framework responsible for type certification of small VTOL aircraft. Thus, their framework covers UAM applications. Among others, the regulatory effort is composed of the SC-VTOL [35] and the corresponding Proposed Means of Compliance with the Special Condition VTOL (MOC SC-VTOL) [10, 36, 37].

MOC SC-VTOL sets airworthiness requirements on all aspects of the aircraft development. Solving the novel problems of UAM in compliance with the new regulatory framework has been the topic of multiple recent publications, found below.

In terms of system development, human-machine-interface solutions to provide adequate handling qualities for lift-to-cruise VTOL aircraft in compliance with SC-VTOL are provided in [33]. The authors of [38] demonstrate how handling quality requirements that are a resultant of the SC-VTOL can be validated with the use of mission task elements. In the papers [39] and [40] functions for path planning and contingency of VTOL aircraft in heavily populated areas are proposed. Coherence with the SC-VTOL is also the topic of discussion. An automatic landing system with landing trajectory generation and tracking is presented in [41]. It covers aspects of the SC-VTOL as well.

MOC SC-VTOL specifies a mission profile for take-off and landing, wherein requirements on the reconfiguration processes are laid out. This creates requirements on the automation functions, responsible for the reconfiguration management of the system. Though not directly mentioned, requirements of the MOC SC-VTOL mission profile with relation to the transition and retransition are implied in the case of fixed-wing VTOL aircraft. During the state of the art research conducted, no publicly available sources were found that address these aspects. The next objective is derived:

Objective 4 Provide a transition and retransition automation concept for lift-to-cruise eVTOL that can fit into the mission profile, defined by the MOC SC-VTOL.

1.2.4 Early-Stage Concept of Operations and its Validation

When flight proving novel functions on technology demonstrators, one of the engineering tasks is how to integrate those functions within the aircraft ConOps. In the case of integrating highly-automated SVO concepts on novel eVTOL aircraft, the importance of this task is exacerbated by the low maturity level of the functions. Therefore, one must assume that probability of malfunctions is inherently higher.

In order to ensure the safety of the crew in these events, in the UAM ConOps proposals of [42] and [43] different stages of automation level (from low to high) are defined. The intent is that over time the level would be increased. The author of the former introduce

the “human-on-the-loop” stage, in which the flight is highly-automated but the pilot is always capable of seizing full control when required. This strategy is referred in this thesis as “fallback” strategy.

In commercial aircraft, hardware redundancy is a common measure to tackle faults [44, 45]. In order to mitigate common-cause, dissimilarity is utilized. The shortcoming of such approaches with relation to the above-mentioned problems is that the redundant components all implement the same specifications. Thus, a malfunction of the concept due to the specification itself would result in a total loss of the FCS. For these novel aircraft, this cannot be excluded.

Though not for eVTOL aircraft, a proposal is introduced in [46], where such a fallback strategy can be realized by having a robust and simplistic fallback control strategy in addition to the highly-automated one. A monitoring function is responsible for switching to the fallback control in the cases where a fault in the high-automation is detected. This approach is significantly different than pure hardware redundancy because the functions are also severely different in nature. A similar notion is followed in the field of run-time assurance [47, 48].

In the case of eVTOL aircraft, proposals of similar runtime assurance strategies is also suggested in [49]. In the context of SVO for such aircraft, the fallback can be realized by utilizing the SVO2 as the primary flight mode and reverting to an SVO1, where the categories of SVO are as defined in Section 1.2.2. This would hold, assuming that SVO1 guarantees full pilot authority in accordance with “human-on-the-loop”.

As seen in the upper paragraph, the idea of the fallback strategy can be found in the context of eVTOL configurations. Yet, publicly available information of realizing it in the context of the reconfiguration between wingborne and powered-lift flight could not be found. This is the basis of the next set of objectives.

A reconfiguration strategy for an SVO1 control concept must be designed. In the event of a reversion to the SVO1 concept, the pilot involvement is considerably higher. Consequently, the awareness of the crew needs to be ensured. This can be satisfied when guaranteeing that aircraft handling between SVO2 (in our case FSD-SVO) and fallback system (SVO1) is consistent. The intended flight control concept is FSD-SVO and thus the reconfiguration concept of SVO1 needs to conform to the one of FSD-SVO. Therefore, the reconfiguration of SVO1 has to be designed such that no properties of the Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics automation are negatively impacted. From here the following is formulated:

Objective 5 Provide transition and retransition automation functions for lift-to-cruise eVTOL that are compatible with an SVO1 control concept in order to enable the fallback strategy. They must:

Objective 5.1 Ensure full pilot authority.

Objective 5.2 Facilitate the proper execution of the fallback concept.

Objective 5.3 Be consistent with the concept of **Objective 3** and the underlying FSD-SVO.

Objective 5.4 Not limit or negatively impact the concept of **Objective 3** and the underlying FSD-SVO.

The UAM ConOps of [43] emphasizes on different aspects of the future UAM aircraft behavior. One of the important points according to the authors is the aircraft-specific operation. This aspect is highly influenced by the stakeholder's desired use-case and mission profile. Given the novelty of the UAM and the aircraft types, designing an adequate ConOps to address the customer wishes but also comply with all regulatory safety concerns is a non-trivial task.

The parts of the ConOps that are related to the mission initiation and execution derive a large volume of the software high-level requirements of the FCS. This is for example the case in the ARP4754 [50] and the DO-178C [51] - the industry standard for software development. There the V-Model is to be followed. According to the standard, the capturing of high-level requirements is the beginning of a sequence of activities that go through the derivation of requirements on the software, the software implementation and its verification. According to the process model, the development effort is finalized with validation of the high-level requirements and hence also the underlying ConOps.

If the ConOps is evaluated to be inadequate for the intended purposes, the whole sequence of development steps needs to be re-initiated after its revision. This has a large impact on the project life-cycle as it introduces delays and high costs [21, 52].

The effort to reduce development time and costs is acknowledged in [33, 53]. In [53] otherwise manually executed tasks are automated. In addition, tedious tasks that do not contribute to the functional development are mitigated. This expedites the development cycle and reduces costs. Continuous integration [54] is leveraged to support in the organization of the tasks and to deploy automatic testing activities. A similar idea is pursued in the newly deployed agile methodologies [55].

The above-mentioned techniques and methods attempt to tackle the problem of the already altered requirements during later phases of the development. They, however, are not designed to address the origin of the problem - improper initial requirement specifications.

For the software development of flight control functions, model-based design [56] is gaining importance. This approach relies heavily on simulation for means of testing and verification. Its common use is acknowledged and endorsed by the certification authorities with the introduction of RTCA DO-331 [57] that lays the foundation for compliance of the model-based design in the context of DO-178C.

The author of [8] proposes a method, where model-based design is used to create executable high-level requirement specifications for closed-loop control functions. In contrast to the full functional development, the so-called DRM is introduced. The DRM is a simplification of the closed-loop response that still takes into account the aircraft

dynamic capabilities. The possibility to simulate the high-level requirement specification can be used to identify conflicts and inconsistencies without the need to go through the whole development cycle.

The methods in [8] cover the aspects of aircraft handling. However, to validate the high-level requirement that stem from the ConOps in a similar fashion, the relevant procedures and automation functions of the FCS that are a consequence of the ConOps need to be made executable as well. This is the basis for the last objective of this thesis. The target is to leverage the benefits of the high-simplification of the DRM methodology and provide an environment, in which the full aircraft flight operation can be simulated. Therefore the last objective is formulated as follows:

Objective 6 Provide a method of modeling the high-level requirements of a flight control system that are derived by the aircraft Concept of Operations. They must:

Objective 6.1 Utilize the DRM method.

Objective 6.2 Model the FCS automation functions in a simplified manner.

Objective 6.3 Be capable of simulating the full aircraft mission.

1.3 Contributions

The research work will contribute in the following aspects beyond the state of technology:

Contribution 1 Safety-driven Transition and Retransition procedures and automation strategy for lift-to-cruise eVTOL aircraft with high-degree of automation control laws.

Contribution 1.1 The reconfiguration procedure integrates a fully automatic transition and retransition seamlessly in the FSD-SVO concept in the nominal case.

Contribution 1.2 In the presence of faults, the pilot decision-making process is rendered non time-critical by procedure design.

Contribution 1.3 During the reconfiguration, safety requirements on the flight envelope and the structural integrity are maintained.

Contribution 1.4 The automation concept is human-centered and implements the above-mentioned procedure.

Contribution 1.5 The automation concept considers and facilitates the operator situational awareness in both nominal and abnormal scenarios.

Contribution 2 Holistic and standard-compliant transition and retransition procedures and automation strategies for no single point of failure lift-to-cruise eVTOL aircraft with a high-degree of automation nominal and a low-degree of automation fallback Flight Control System.

Contribution 2.1 The procedure definition takes the outcome of **Contribution 1** and accounts for scenarios, where a less automated fallback system has to be capable of performing a takeover and reconfiguration.

Contribution 2.2 The safety properties of **Contribution 1** for a high-level of automation are retained. For the low-level of automation, they can still be maintained, but also full operator authority in all flight phases is ensured.

Contribution 2.3 The support for operator awareness is facilitated twofold: Firstly, the interpretation of the relevant operator input is equivalent in both control modes. Secondly, the execution of the mitigation strategies in the presence of faults are harmonized.

Contribution 2.4 The definition of the procedure concept meets the requirements imposed by the currently available standards and certification requirements.

Contribution 3 Methodology of functional development of automation behavior and integration with design reference modeling.

Contribution 3.1 Possibility of simulation of pilot-in-the-loop flight operation without the necessity of system architecture-specific considerations or full FCS development.

Contribution 3.2 Functional decomposition of automation tasks that can be used for the software architecture design.

Contribution 3.3 Practical implementation of the methods, developed in **Contribution 1** and **Contribution 2** as automation behavioral models onto a no-single point of failure experimental lift-to-cruise eVTOL aircraft.

1.4 Outline

This chapter presented the motivation of this thesis. The state of technology with relation to the automation of lift-to-cruise eVTOL aircraft with dedicated traction system was researched. Furthermore, topics of improvement were identified in the form of objectives that this theses solves. The solutions are summarized in the form of contributions. The upper portion of Table 1.1 can be used as a reference as to which contributions addresses which objective.

The remainder of this thesis is structured as follows. Firstly, Chapter 2 describes all theoretical preliminaries that are necessary for the solutions of the problems. It places emphasis on both theoretical and implementation methods. In addition, it lays out the nomenclature and guidelines used throughout all following chapters.

Table 1.1: *Objective-Contribution and Chapter-Contribution Traceability Matrix*

	Contribution 1	Contribution 2	Contribution 3
Objective 1	✓	✓	
Objective 2	✓	✓	
Objective 3	✓		
Objective 4	✓	✓	
Objective 5		✓	
Objective 6			✓
Chapter 2	✓	✓	✓
Chapter 3	✓		
Chapter 4		✓	
Chapter 5			✓
Chapter 6	✓	✓	✓

Chapter 3 presents the developed high-degree automation method. It proposes a reconfiguration strategy that is both human-centered and compliant with an SVO2 concept. In Chapter 4 the automation method that would satisfy an SVO1 concept is shown. In addition, explanations as to how the methodology can serve as a fallback are provided. Lastly, this chapter demonstrates how and under which conditions both high- and low-degree of automation methods can satisfy the requirements, placed by the regulatory organs.

In Chapter 5 the method of modeling the high-level requirements of a flight control system that are derived by the aircraft Concept of Operations is explained. It further demonstrates how and to what extent the solutions can be kept agnostic to the system architecture and how a functional decomposition and allocation is achieved. In that chapter, the implementation of the methods of Chapter 3 and Chapter 4 is provided. The main body of the thesis is concluded with Chapter 6 where simulation results using the product of Chapter 5 are presented and elaborated upon. The results serve as validation of the methods of Chapters 3, 4 and 5. The lower portion of Table 1.1 provides a reference where different contribution aspects are addressed. Finally, the thesis is concluded with Chapter 7 where a summary and reflection on the research process and recommendations for future work are provided.

Chapter 2

Theoretical Background

The introduction of automation functions requires understanding of the underlying system properties and characteristics. The control concepts and operational procedures need to be accounted for. In addition, the automation design needs to be rooted in the common practices and guidelines, established in the field. This is necessary in order to ensure that all known aspects, potential pitfalls and hazards are addressed appropriately.

This chapter provides this preliminary information and lays down the foundation for all contributions within the thesis. It explains the theory behind the applied methods and their motivation. It analyses the aerodynamic and structural characteristics of lift-to-cruise eVTOL aircraft.

This chapter is composed as follows. Section 2.1 introduces and explains key terms that are consistently used throughout the thesis. They describe how systems and functions operate and how different types of operation are classified with regards to their properties. Section 2.2 provides the theoretical background on the design of automation functions. It explains the types of interactions the automation module has with the surrounding systems. In addition, it exposes all potential hazards that an introduction of automation has on the system safety and the aircraft operation. Principles and guidance on how to address and avoid potential automation mishaps are summarized. The section furthermore lists the design methods that are used in the automation functions of the further chapters and shows how they can be realized in the modeling environment that is used in the thesis.

In order to automate a system adequately, its inherent properties need to be understood as they impact the automation design. Section 2.3 elaborates on these aspects. It shows how different components and their failure modes may negatively affect the system response. Methods to mitigate the negative influences are discussed. This lays down the foundation of requirements that are set on the automation modules in the later chapters. The properties that need to be known and accounted for are not limited to the physical design of the system but also on the control concept. Section 2.4 provides an overview of the design decisions made in terms of Simplified Vehicle Operations that the automation module needs to interact with. These are the control algorithms on one hand and the pilot input elements on the other hand. Lastly, Section 2.5 summarizes the relevant requirements

that are introduced from the regulatory organs. They need to be accounted for and hence impose additional requirements on the automation functions and their fit in the operational procedures.

2.1 Common Terminology

This section introduces a set of common terms that are used throughout the thesis. Based on the method of operations, it is possible to classify both systems and functions. Their classification types are listed and explained in Sections 2.1.1 and 2.1.2 respectively.

2.1.1 Characterization of System Design

Among others, functions and systems can be grouped based on their operation under the presence of failures. This thesis distinguishes between two main classes -“fail-open” and “fail-safe” [58]. Under certain malfunctions, the former cease to fulfill their intended function. In aviation, a fail-open system and a single point of failure system are often used interchangeably, i.e. in aviation a fail-open system is one that experiences a total loss in the presence of one fault. The latter type - “fail-safe“ - is typically used in safety-critical applications where the loss of the system is attributable to casualties. Therefore, in an event of a fault this type of system enters a predefined state of operation that does not severely impact the overall system performance.

The class of fail-safe operation is further broken down based on the consequences the failure has on the subsequent operation. For this thesis, the following types are relevant and their definitions stem from [58–60]:

- **Fail-Passive:** A critical fault causes the system to revert to a state that is deemed safe. Usually this state is chosen such that the impact to the surrounding systems is low.
- **Fail-Active:** The operation of the application is continued despite the occurrence of a critical fault. However, the system performance is reduced.
- **Fail-Operational:** The operation of the application is continued with no noticeable performance changes despite any fault occurrence.

2.1.2 Characterization of Functional Operation

During run-time functions have different states or “modes” of operation. This section breaks down the types of operation functions can have. This thesis follows the conventions as found in [61] and [62]. The states that are relevant for this thesis are as follows:

- **Unavailable:** If a function is unavailable then the conditions, under which proper execution of the function can be guaranteed, are not met. These could be loss of sensor information, incorrect envelope and others. For example, the function “terrain following” is unavailable without height above ground information. The function “spoiler deployment” is unavailable during cruise flight.
- **Available:** An available function is one, which is not unavailable.
- **Engaged:** The current function is available and is being executed. As a consequence, its output is affecting the system-behavior. For example, “spoiler deployment engaged” implies that currently the spoiler is utilized for braking or roll control. Sometimes, the term “active” instead of “engaged” is used. In this thesis “engaged” and “active” are used interchangeably.
- **Armed:** An armed function is one, which is available and will become engaged if predefined conditions are met. This trigger may be automatic, manual or both. While armed, the function does not yet affect the system-behavior. Usually if functions are meant to be engaged in a sequence, the next function planned to be engaged is the one that gets armed. In the example used, during an automatic landing sequence the “spoiler deployment” function may become armed shortly before touchdown. Manual input below the predefined safe speed would engage the function.
- **Disarmed:** A disarmed function is available, but is neither armed nor engaged. It does not affect the system-behavior.

It must be noted that a function does not need to be armed but may also be disarmed prior to becoming engaged. Furthermore, after disengagement that is not due to unavailability, an engaged function may become armed or disarmed depending on the application.

Another commonly used term in the aviation industry is “mode”. In [63] the term mode is defined as a “set of related features and functional capabilities of a product”. More specifically, this thesis refers to mode as a combination of engaged functions that produce a specific type of in-output response, i.e. an “operational mode”. This definition is consistent with the notions, found in [64].

2.2 The Automation of a Flight Control System

In the context of an aircraft, the automation is responsible for coordinating the efforts of the systems in the vehicle in order to alleviate the need for crew input in certain operational activity and thereby increase the overall system safety [65]. The responsibilities of a flight control system automation include the management of the control concept, it interacts with the surrounding systems to facilitate the achievement of a task and it manages the

reconfiguration state and more. The mechanics of the processes the automation addresses are different in terms of underlying physical effects. As a consequence, the underlying automation functions vary vastly in terms of design. Despite the differences, however, the fundamental concepts and considerations of the automation function design can follow the same established guidelines and practices.

This section introduces the theoretical basis of the automation functions, presented in this thesis. It is structured as follows. In an attempt to reduce the pilot workload, new potential hazards and problems arise by the use of automation. Following this, Section 2.2.1 provides the common challenges associated with the introduction of automation in the aircraft operation. From those challenges, design principles are derived. They guarantee that the automation design will account for potential shortcomings. They are summarized in Section 2.2.2. Section 2.2.3 presents the design methodology of an automation function. An automation function is constructed from different design patterns. The section provides design constructs and demonstrates how they can be formally defined and parameterized. Lastly, in Section 2.2.4 an overview of the development environment that is used for this thesis and the implementation rules that the solutions need to follow are provided. A simple example is used to illustrate the implementation methods.

2.2.1 Automation Aspects and Challenges

This section serves as an overview of the aspects the automation module must consider and the challenges it must address to ensure its proper execution. Section 2.2.1.1 covers the design aspects and Section 2.2.1.2 provides a list of challenges and hazards that arise due to the automation of tasks within the aircraft operation.

2.2.1.1 Automation Design Aspects

This section summarizes all aspects related to the run-time execution behavior of the FCS that impact the design of automation functions. An overview of these topics can be found in Figure 2.1.

One automation task is reacting to the operator input via the Human-Machine-Interface (HMI) (Interaction Concept). The automation is responsible for processing the crew requests, communicated to the system via control inceptors and discrete inputs in the cockpit. The subsequent actions are a function of the chosen automation strategy. Depending on the mission segment, the function availability and state of configuration, these inputs may trigger a sequence of automation tasks. Otherwise, a request may be discarded if it occurs in an inappropriate flight or configuration state. Whether an automation task is initiated or not must be fed back to the operator along with the state of the automation via the cockpit indications in order to ensure pilot awareness. Therefore, a requirement that is allocated to the automation is adequate information supply to the crew.

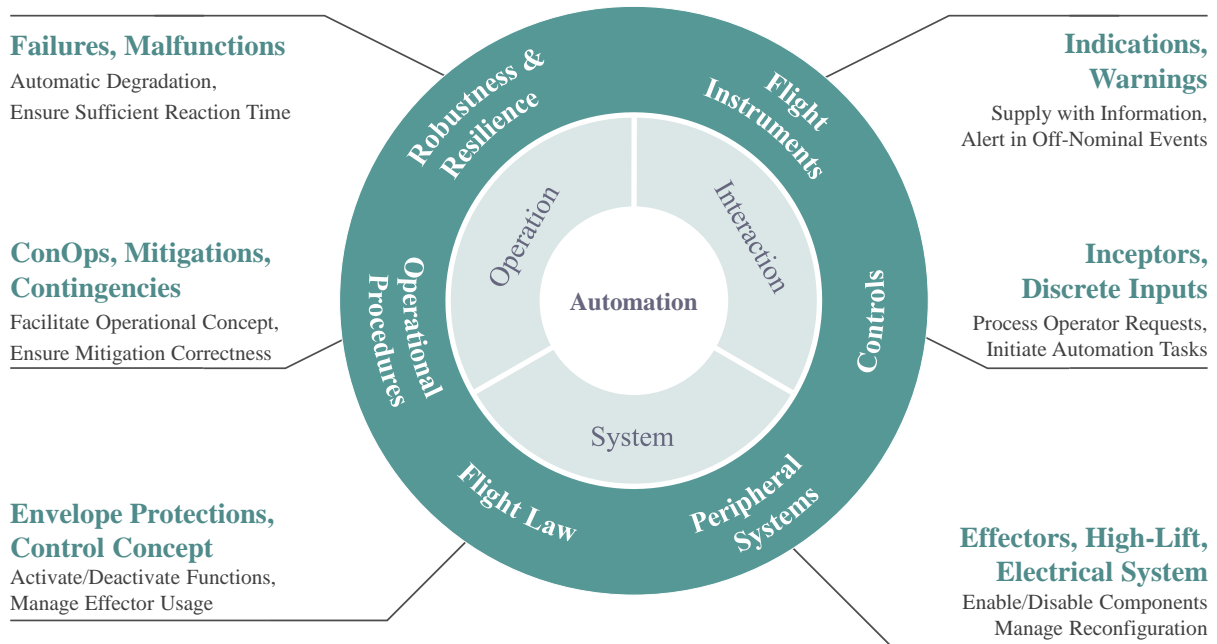


Figure 2.1: *Automation Design Considerations and Aspects*

If an automatic task is initiated, the automation module is responsible to orchestrate all involved subsystems in order to achieve the specific goal (System Considerations). This includes the deactivation and activation of components (e.g. sensors, motors controllers, etc.) and functions (e.g. envelope protections). In addition, the automation may manage the state of configuration. It feeds the control algorithms with the required data so as to facilitate the proper execution of the control concept (e.g. allowed effector usage). If necessary, the module also ensures that the sequences of events occur in a specified order.

Consequently, the automation module is involved in the implementation of the operational concept of the aircraft. It ensures that both nominal and off-nominal procedures can be performed as defined in the Concept of Operations. In the cases of component malfunctions, the automation must react accordingly by executing an automatic function degradation, thereby allowing for fail-safe system behavior and ensuring continued safe flight.

2.2.1.2 Automation Challenges

When providing automation functions, the goal is alleviating the crew workload. In theory the complexity of operating the vehicle should be reduced by the presence of automated tasks. However, practice has shown that depending on the design of the automation module, a shift in the complexity could be observed instead [66]. In such a scenario, the operator is left managing the automation, which may prove more demanding than the task it is meant to automate. This is the case if the management of the automation is

not intuitive. This section provides an overview of the major fallacies associated with automation function design that may negatively impact the aircraft handling instead of improving it.

Unless fully automated, a system relies on human operator input and intervention. The involvement depends on the degree of automation. Given that fully automatic operation “may be an Utopian idea” [67], the role of the human operator cannot be neglected in the automation design. The more advanced the control system, the more critical is the role of the human operator [68].

Whenever a fallback principle is applied, the responsibility of the crew shifts from a supervisor to an operator of the system. With increasing levels of fallback, the crew takes control of an increasing number of processes and tasks. For this purpose, Bainbridge [68] identifies several relevant operator qualities. Those are directly correlated to challenges in the automation design, found in [66]. The findings of both sources - [68] and [66] are summarized below.

Challenges in Operating the System

Depending on the automation design and application, Sarter et al. [66] claim that without proper addressing, “[the workload is] unevenly distributed, not reduced” [66]. The rationale behind this is that often the automation is incapable of capturing all required data for reaching a feasible decision and therefore relies on the operator for additional input. In these instances this leads to the problem that a workload reduction due to the process automation is preceded by a spike of workload in order to initiate that automated process.

Regardless of the operator workload distribution, the type of required operator experience has to be different than for a system without automatic functions [66]. The crew is required to understand the behavior of the automation, its capabilities, limits and available functions. This is necessary in order to supervise and control the system effectively, assess the correctness of its operation and understand the automation process. Therefore, the author recognizes “the need for new approaches to training” [66].

Although the type of training may not be crucial in nominal operation, in [68] the author argues that in the instance of a fallback the assumption is that the system behavior is abnormal. This implies that contingency actions are necessary. This is exacerbated when the automation does not succeed in its attempts to counteract a failure. In such occurrences this could mean that an envelope is exceeded [66]. If the operator is not familiar with the system, then the operator might be unable to determine whether the adverse behavior is caused by the system itself or due to possible inadequate input. Ironically, the latter is caused by the pilot in an attempt to stabilize the system. Therefore, the operator’s experience with the machine and its automation are the decisive factor. If the automation is not designed properly, the level of training with the automation may have to be higher than average and not lower. This contradicts the concept of Urban Air Mobility.

Challenges in Understanding the System Behavior

In order for the operator of a system to be aware of the correctness of the automation functions, adequate information supply is required. According to the Bainbridge [68], “[...] an operator will only be able to generate successful new strategies for new situations if he has adequate knowledge of the process” [68]. To come up with an effective action plan in an adverse situation requires situational awareness, which is only gained through understanding of the system feedback. A lack of mode awareness in critical situations is a great hazard.

The actions of the operator are directly dependent on the current state of the process. In short, over time the operator builds up knowledge of the procedure flow and can mentally prepare for upcoming state changes or possible adverse situations. Even before taking over system control, the operator has to already be aware of what the effective action plans are. This is necessary in order to assure a fast reaction time and adequate response. Therefore, sometimes it is better to execute processes in fixed sequences despite a loss of efficiency.

“New opportunities for new kinds of error” [66] are associated with lack of mode awareness. Those errors occur whenever the crew is not aware that the operational mode has changed to one where the control concept is significantly different [69]. The mental perception of how the system is supposed to behave does not align with the current behavior. This creates a hazard even though the crew input is correct for the perceived mode and the automation is reacting correctly to the inputs supplied.

Challenges in the Interaction between Human and System

An inappropriate operational concept leads to problems in later operation. This, on one hand, can create inadequate handling but, on the other hand, influences the perception of the operator about the robustness of the system.

The robustness of the automation functions impact how the operator would perceive and interact with the system. If not resilient, a system quickly gets branded as unreliable. Consequently, the monitoring effort of the crew increases in anticipation of a possible malfunction of the automation. This increase in alertness may even persist in phases flight where nothing substantial occurs, leading to fatigue.

Bainbridge [68] argues that often the system designer’s perception of the human operator is negative. Automation is applied when the operator is “inefficient” [68] at performing a task. The argumentation is that the crew cannot deal with the high amount of stimuli and hence is unable to control the system as good as the automation would. However, then the operator cannot be expected to monitor the automation or understand its state based on those same stimuli.

If not fully automated, then the operator is expected to execute tasks that were impossible to automate due to safety restrictions or high complexity. Sometimes the tasks the operator is left with are arbitrary and they have inadequate support from the machine

to execute them properly. A similar effect occurs when automating tasks “for the sake of automation”, not taking into account how the operator expects the tasks to be executed, leading to confusion.

This section underlined common automation design errors that lead to inadequate handling of the system. In the next section, principles used to classify and develop automation functions in order to mitigate such hazards are provided.

2.2.2 Automation Design Principles

Automated systems are commonly classified in terms of their level of automation. Depending on the industry and used references, different levels and classification metrics are utilized. Publicly available sources that provide level assignment include [70], [29] and [71]. The properties the classifications have in common is the extent the human is involved in the operation, their responsibility and command authority. The highest level is therefore one, in which the crew has no designated responsibility because the whole operation is covered by the automation. Respectively, the lowest level includes no automation and the operator has full authority over the system.

When relating the levels of automation to the types of Simplified Vehicle Operations of Section 1.2.2, then SVO3 is a concept of the highest automation level because it assumes fully automatic flight. The remaining two phases of Simplified Vehicle Operations correspond to levels, where non-negligible pilot involvement is necessary in the operation of the eVTOL, making all potential pitfalls of Section 2.2.1.2 applicable.

The purpose of automation functions in SVO1 and SVO2 is to automate parts or complete procedures and tasks within the system. However, these functions need to communicate and follow instructions from the pilot. More importantly, the operator is seen as the “last line of defense” and plays a key role in the mission. An important concept in the type of automation design has emerged, referred to as “human-centered” [29]. This concept emphasizes that the automation must coordinate its responsibilities with the crew and must “[enable] a more cooperative human-machine relationship in the control [...]” [29].

In [29], Billings derives qualitative properties of automation, with which the behavior of the system can be evaluated from the perspective of the human operator. They can be found in [29] and are summarized below:

- **Complexity:** The interaction and response possibilities of the systems are not readily available, known or understandable for the operator. A negative example the author provides is the operation of early flight management systems. Certain input of the crew used to cause a disengagement of modes. Because of the system complexity, the disengagement of the mentioned modes was not known by the crew

prior to the input. This had an adverse aircraft level response especially if this disengagement is not desired by the crew. In order to avoid such unwanted system responses, human-centered automation needs to operate in a simple manner.

- **Brittleness:** In the presence of abnormal events, the automation “does not have desired behavior at or close to some margin of its operating envelope” [29]. The automation therefore operates robustly within its design conditions and in the presence of known adverse scenarios. However, it breaks down completely should something abnormal outside the specification occur. An example the author provides is the functioning of the early traffic collision avoidance system designs, where in certain situations it was unable to compute the appropriate avoidance commands. Such examples are the reason a human-centered automation needs to be built resilient.
- **Opacity:** Even though the its operation may be correct, the automation is opaque if the crew is unaware of the current system actions, their reason and the automation intent. If the system complexity or the lack of operator training are not the issue, then inadequate or complete lack of feedback to the operator is the cause of automation opacity. The feedback therefore needs to be intuitive, concise and clear. With increasing automation complexity this may not necessarily be trivial, but the human-centered automation must be transparent to the user.
- **Literalism:** The automation functions in conventional aviation are deterministic systems, i.e. they will calculate the same outcome given the same input trajectories. Therefore, the system reaction is a consequence of its specification. The problem arises when a system response is not considered in the specification but is necessary. In contrast to brittleness, this may even occur within the operating envelope. A flexible system should allow for manual intervention in such scenarios so as to mitigate potential hazards due to unaccounted effects within the aircraft operation envelope.

An aircraft automation must avoid exhibiting the above-mentioned characteristics. This is not trivial if no guidance material is available. Sufficient operator training with the automated system certainly can improve the person’s perception and awareness of the automation even if the design exhibits unfavorable characteristics. Billings [72] suggests several properties that the developer must take into account during the automation design so as to mitigate operational mishaps and improve the system properties. They are summarized below.

- **“Responsibility and command authority” [72]:** Whenever command authority is taken away from the system operator, then in certain situations the full performance of the aircraft cannot be used by the pilot. Therefore, control authority should

only be limited if absolutely necessary. This should be known by the crew and - if necessary - provisions should be included, in which pilot authority can be gained back.

- **“Operators must be involved” [72]:** Even if systems are moving towards higher automation and therefore less necessary operator actions, the pilot should never have a pure monitoring role. Some level of human involvement is necessary in order to ensure mode awareness, otherwise in an event of intervention, a “change from passive monitor to active problem-solver can be abrupt and difficult.” [72]
- **“Operators must be informed” [72]:** When developing the feedback to the crew, the designer of the system automation must assume the “pilot’s [...] role and way of thinking” [72] in order to ensure intuitive information supply in operation. The automation data supply therefore must be timely, concise and unambiguous. The amount of information must be sufficient to reach an informed decision but not be overwhelming.
- **“Humans must be able to monitor the automation” [72]:** The reason behind an automation process must be clear to the crew. The workload of the operator should not be increased and only abnormal scenarios and malfunctions need to be indicated. However, at the same time the operator must know whenever a commanded action is executed or not.
- **“Automation must therefore be predictable” [72]:** During each phase of an automated process, the operator must know what the involved modules and components are and what their malfunction causes in terms of high-level system behavior. This is necessary in the events that they need to take command over the process. In preparation for this situation, the pilot needs to know what the sequence of automation events are. “This, of course, requires that the pilot have an accurate mental model of how the automation is expected to behave.” [72]
- **“Automation must monitor the human” [72]:** Even with low involvement, operator fatigue occurs. This may lead to false or potentially hazardous actions. Similar to human monitoring of the automation, the automation must monitor the operator actions and supply warnings if events - especially abnormal ones - are not acknowledged.
- **“Communication of intent” [72]:** The automation concept should be designed in such a way that the operator actions point to clear instructions to the automation module and vice-versa. This is especially true for abnormal events where the amount of information supplied between the involved parties (operator, automation, system components) is increased.

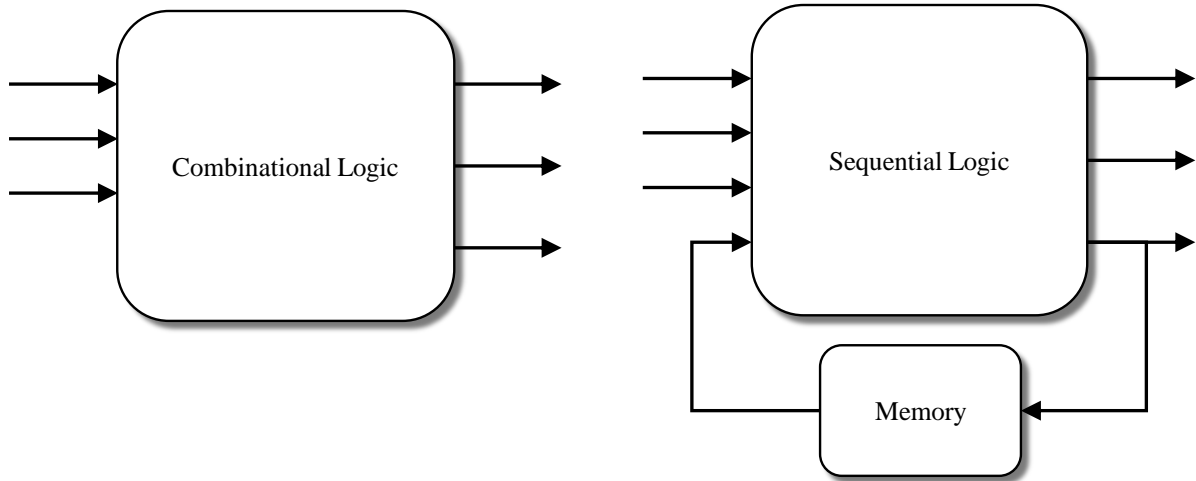


Figure 2.2: *Graphical Representation of the Combinational Logic (Left) and the Sequential Logic (Right)*

This concludes the theoretical principles of automation function design. The following section introduces the mathematical preliminaries and the design methods, with which a human-centered automation module can be created.

2.2.3 Automation Design Methods

This section provides the design methods used for the creation of automation functions. It furthermore derives and explains the mathematical notations, used throughout the thesis. As seen in Figure 2.1 of Section 2.2.1.1, the automation module of a FCS is an aggregate of many tasks. Each task is composed of multiple logical decisions. Therefore, an automation module is an integration of a considerable number of logical operations.

The theory of computation defines two major models of computations that are relevant for this thesis - the “combinational” and “sequential” logic [73, 74]. Combinational logic is a method of computation, in which the current outcome of the logic is solely dependent on the current inputs it is fed. Therefore, the decision-making process reaches the same output regardless of the input time history. On the other hand, sequential logic relies on so-called “memory” for its conclusion. Sequential logic is still deterministic but the output is dependent on the time history of the inputs. For the sake of clarity, a graphical representation of the two models of computation is provided in Figure 2.2.

This section is structured as follows. Sections 2.2.3.1 and 2.2.3.2 introduces the basic concepts and notions, necessary for understanding the design methods, utilized in the thesis. Section 2.2.3.3 provides an overview of the combinational logic methods. Section 2.2.3.4 does the same for sequential logic.

2.2.3.1 Sets and Functions

All quantities the automation modules uses to reach a decision are members of certain sets. Apart from the well-known sets used for numbers, such as the set of natural numbers \mathbb{N} , integers \mathbb{Z} , real numbers \mathbb{R} , here the boolean set \mathbb{B} is introduced. This set is defined as

$$\mathbb{B} = \{true, false\}. \quad (2.1)$$

As seen later, the automation function design relies heavily on operation in the boolean domain. In graphics *true* and *false* is substituted with “1” and “0” respectively to conform with the common conventions of publicly available literature.

For the sake of completeness, the notation of functions is explained. A function executes a predefined operation, taking members of one set and computing an outcome in a defined set. The set of input members is referred to as domain and the latter is defined as the range. Range and domain can, but need not be the same set. For example, in the binary function

$$f(a, b) = c \quad (2.2)$$

the inputs are $a \in \mathbb{R}$ and $b \in \mathbb{Z}$, whereas $c \in \mathbb{B}$. In this case, the range of f is notated as $\mathbb{R} \times \mathbb{Z}$ and the domain is \mathbb{B} . Therefore, another notation of f is

$$f : \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{B}. \quad (2.3)$$

A function used throughout the thesis is the indicator function. It is notated with χ and is a unary function with a range in the boolean domain. It produces *true* whenever the passed input is within a set, which is specified in the index of the indicator function. For example, let A be a subset of a larger set, notated with U . Then, if $x \in U$, the indicator function output is

$$\chi_A(x) = \begin{cases} true & \text{if } x \in A \\ false & \text{otherwise.} \end{cases} \quad (2.4)$$

2.2.3.2 Relational Operators

Relational Operators are binary functions that stem from computer science. In this thesis they are most commonly used to classify certain input data for later use in the automation functions. For example, Relational Operators are used to determine whether an airspeed is high or low enough for a given decision. The Relational Operators are summarized in Table 2.1 in order to establish the used notation. Within this thesis, “*equals*” function is notated as “*==*”. This is necessary in order to differentiate between equality and assignment. The latter is denoted with “*=*”.

Table 2.1: *List of Relational Operators*

Name	Notation	Output
Equals	$a == b$	<i>true</i> if a equals b , <i>false</i> otherwise.
Not Equals	$a \neq b$	<i>true</i> if a does not equal b , <i>false</i> otherwise.
Greater	$a > b$	<i>true</i> if a is greater than b , <i>false</i> otherwise.
Less	$a < b$	<i>true</i> if a is less than b , <i>false</i> otherwise.
Greater or equals	$a \geq b$	<i>true</i> if a is greater than or equals b , <i>false</i> otherwise.
Less or equals	$a \leq b$	<i>true</i> if a is less than or equals b , <i>false</i> otherwise.

2.2.3.3 Combinational Logic

The methods used in this thesis that belong to the category of combinational logic are Relational Operators, boolean algebra (or boolean functions) and Truth Tables. Boolean functions can be used stand-alone for certain decision-making. They are also useful as atomic elements, necessary for an increasingly complex decision-making process. The Truth Table is a useful tool to utilize multiple relational and boolean expressions. The used methods in the above-mentioned order are explored in this section.

Boolean Algebra

Boolean algebra uses k -ary¹ functions that are in the boolean domain. Such constructs are also referred to as boolean functions. In this thesis, boolean algebra is used for evaluating conditions in combination with the outcomes of relational operations. The expressions are either used directly for certain decisions or taken in combination in more complex processes. Exhaustive information on boolean algebra is provided in [75]. The relevant information is summarized below.

Boolean algebra is composed of three basic operators - the logical “and”, “or” and “not” [74]. Their computation is summarized in Table 2.2, in which A and B are the inputs to the operators. All other “derived boolean operations” [74], are a resultant of these three operations.

All boolean functions are built on the basic operators of Table 2.2. As an example of a boolean function, the decision-making process of a pilot is modeled. The presented logic must evaluate whether the operator is cleared for landing under visual flight rules. In the constructed example, information can arrive from three sources. Landing clearance can be granted either via radio link or via green flashes directed towards the aircraft. Let $radio \in \mathbb{B}$ and $flash \in \mathbb{B}$ express whether the pilot has registered these signals respectively. In addition, pyrolytic lights could be directed toward the aircraft, signaling that landing

¹ k -ary means that the function has a number of inputs, equal to k .

Table 2.2: *Boolean Algebra: Basic Operators*

A	B	AND $A \wedge B$	OR $A \vee B$	NOT $\neg A$
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	

Table 2.3: *Example of an Exhaustive Truth Table*

<i>i</i>	pyro	radio	flash	f
1	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
2	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
3	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
4	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>
5	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
6	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
7	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>
8	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>

Table 2.4: *Example of a Simplified Truth Table*

<i>i</i>	pyro	radio	flash	f
1	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
2	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
3	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
4	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>
5 – 8	<i>true</i>	–	–	<i>false</i>

should not be executed, regardless of previous instructions. $pyro \in \mathbb{B}$ is used to express whether this is the case. Therefore the decision-making process of the pilot whether the aircraft is cleared for landing can be expressed as $f : \mathbb{B} \times \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ and namely

$$f(\text{radio}, \text{flash}, \text{pyro}) = \neg \text{pyro} \wedge (\text{radio} \vee \text{flash}). \tag{2.5}$$

Truth Tables

Though Equation 2.5 may be manageable, with increasing number of inputs and decision complexity, certain boolean functions may become difficult to comprehend. Truth tables are combinational logic constructs that are useful in order to gain understanding as to the designed decision-making process.

A Truth Table for the above-constructed example can be found in Table 2.3. The Truth Table explores the outcome for every combination of inputs and therefore has a size of $2^m \times (m + r)$ where $m \in \mathbb{N}$ is the number of inputs and $r \in \mathbb{N}$ is the number of outputs. To manage with the increasing size of the table, simplifications are permissible.

For example, notice that in Table 2.3 for *pyro* being *true* (rows five to eight), the outcome is always *false* regardless of the remainder of inputs. This can therefore be summarized as depicted in Table 2.4, so long as it is guaranteed that the table is fully determined.

The way the outcome of the Truth Table can be converted to a boolean function is by understanding that each line of the table is a boolean equation by itself, composed of logical “and” and “not”. For example, one combination that leads to a positive outcome is

$$f_2(\text{radio}, \text{flash}, \text{pyro}) = \neg \text{pyro} \wedge \neg \text{radio} \wedge \text{flash}, \quad (2.6)$$

where the index 2 refers to the particular row of the Truth Table. Based on this, the outcome of function f can be expressed as

$$f(\text{radio}, \text{flash}, \text{pyro}) = f_2 \vee f_3 \vee f_4. \quad (2.7)$$

Here the inputs of the individual function components f_2 , f_3 , and f_4 are omitted for the sake of readability.

2.2.3.4 Sequential Logic

In sequential logic, certain information from previous computations is stored in a “memory” function as previously depicted in Figure 2.2. Thus, the current outcome of a sequential logic is dependent on the past input history that has driven the contents of the memory function to that particular state.

This thesis relies heavily on the usage of sequential logic for the development of automation functions. For example, this logic is often used for the creation of counters that track the duration of a certain condition. However, sequential logic is primarily used for determining the mode of operation.

This section provides an overview of the used constructs within the domain of sequential logic. Several examples are provided - the Latch, the Edge Detector and the Confirmation Counter. They are used to illustrate the notation principles and the key characteristics of sequential logic. Finally, the Finite-State Machines used in this thesis are presented. Finite-State Machines are the key design method used for the proposed automation functions of this thesis.

Latch

The Latch is widely used as an example to explain sequential logic due to its simplicity. This construct is used to store information and is enabled by the memory function. This information storage is also referred to as “state”. In this thesis, the state stored from the previous function call shall have the superscript $'$. The stored state of a Latch for example would be $Latch' \in \mathbb{B}$. In order to fully define a state, its initial value needs to explicitly be mentioned. In this thesis, the initial states of the memory are always denoted with the index 0. For the Latch, usually

Table 2.5: *Truth Table of a Latch if the Memory is Treated as an Input*

Latch'	a	b	Latch
<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

$$Latch_0 = false. \quad (2.8)$$

The Latch uses two different conditions to change its state. It changes from *false* to *true* based on the first condition and from *true* to *false* based on the second. Different decision possibilities are available in the cases where both conditions are in effect. In this example the Latch shall not experience a state change, i.e. if both conditions are *true*, then the state shall retain its previous value. Thus, if $a \in \mathbb{B}$ and $b \in \mathbb{B}$ are the two respective conditions, the operation of a Latch is formulated as

$$Latch(a, b) = \begin{cases} true & \text{if } a \wedge \neg b, \\ false & \text{if } \neg a \wedge b, \\ Latch' & \text{otherwise.} \end{cases} \quad (2.9)$$

The stored state *Latch'* is a function of the previous a and b values. This notation is omitted in this thesis for the sake of readability. The current value of the function is passed on to the memory for storage. In the next iteration, that value becomes *Latch'*.

Note that with the exception of the memory function, sequential logic behaves exactly as combinational logic and in fact uses the same constructs. If the stored values are regarded as inputs to the function, then their equivalence can be shown. As an example, the Latch operation can be depicted as a Truth Table as visible in Table 2.5.

Confirmation

Confirmation Counters are another function that is built with sequential logic. They indicate whether a condition has been fulfilled for a predefined number of iterations and in this thesis are used to evaluate whether certain flight conditions are attained. The state of the system is the Counter value and is therefore an integer with an initial state of

$$Counter_0 = 0. \quad (2.10)$$

Suppose the condition requiring confirmation is $a \in \mathbb{B}$. Then the Counter value is a function based on that condition and namely

$$Counter(a) = \begin{cases} Counter' + 1 & \text{if } a, \\ 0 & \text{otherwise.} \end{cases} \quad (2.11)$$

As long as the Counter value is above a certain value - in this case *threshold* - then the condition is confirmed. The Confirmation function can be fully defined as

$$Confirm(a, threshold) = Counter(a) > threshold. \quad (2.12)$$

Edge Detectors

Edge Detectors are useful for acknowledging changes in a signal. Often they are used in combination with Confirmation Counters. An example where an Edge Detector would be used is for actions taken due to button presses that enable or disable a function. The action must be performed only on a change of the input that expresses the button action. Otherwise, the function would be enabled and disabled continuously.

The Edge Detector memorizes the input of the system. The state of the Edge Detector system is therefore the previous input value. The initial state s of an edge detector is chosen as

$$s_0 = a, \quad (2.13)$$

where a is the input of the edge detector function.

Depending on the way the Edge Detector operates, three types can be distinguished. The first detects any change of the input, the second detects rising edges (i.e. the input going from *false* to *true*) and the third detects falling edges (i.e. the input going from *true* to *false*). In order, the functions can be formulated as

$$\begin{aligned} Edge(a) &= a \neq s' \\ EdgeIncrease(a) &= a \wedge \neg s' \\ EdgeDecrease(a) &= \neg a \wedge s' \end{aligned} \quad (2.14)$$

Finite-State Machines

Finite-State Automata are sequential machines that are used to model system behavior (prediction) or specify system behavior (synthesis) [76]. In the case of the development of automation functions, the latter is of interest. These constructs are used for higher-level decision logic, such as mode selection, and are therefore at the core of the artifacts of this thesis.

Finite-State Machines are grouped in accordance with their functional scope. Exhaustive analysis on the applicability of the different types is available in [77]. According to the results of the analysis, Mealy State Machines are the preferred option, primarily due to

their “faster reaction to inputs” [77]. Therefore, only this type of automaton is elaborated upon here. The notation principles stem from [76, 78, 79] and are adapted to fit in the notation of the remainder of the thesis.

A Mealy Machine M is fully specified with the tuple

$$M = (S, s_0, U, Y, \delta, h). \quad (2.15)$$

In addition, the State Machine state is notated with s .

In Equation 2.15, S is a finite set of states, hence the origin of the name of the logic. s can assume a member of this set, i.e. $s \in S$. s_0 is the initial state or “starting state” [78]. U and Y are finite sets referred to as the “input alphabet” of the input u and “output alphabet” of the output y . They specify the values the State Machine inputs and outputs can assume. $\delta : S \times U \rightarrow S$ is the “transition function” and $h : S \times U \rightarrow Y$ is the “output function” of the automaton.

The system description allows for multiple inputs and outputs. In addition, multiple State Machines can be summarized. If the system is composed of n State Machines, then each automaton has its own state set. The state set S of the system is consequently defined as

$$S = S_1 \otimes S_2 \otimes \dots S_n, \quad (2.16)$$

where $i \in [1 \dots n]$ is the corresponding State Machine and S_i is its state set. In addition, suppose the automaton has m inputs and r outputs. The input and output tuples are expressed in the same manner and contained in U and Y of Equation 2.15. The interested reader is advised to [76] for a thorough derivation.

In the case of a FCS, the logic is clocked [75], i.e. the State Machine performs its actions in predefined cycles. The mechanics of the logic are as follows. Based on the input string, transitions take place via the transition function tuple. The Machine may go from one state to another or remain in the current state. For instance,

$$\delta(s_1, true) = s_2 \quad (2.17)$$

implies that the State Machine transitions from state s_1 to state s_2 if it receives an input of *true*. Similarly, the automaton remaining in the state s_1 if receiving *false* is denoted as

$$\delta(s_1, false) = s_1. \quad (2.18)$$

Deterministic State Machines specify transition conditions for all combinations of input tuples [78]. The output functions are organized in a similar fashion. Whenever a transition has taken place, the output function computes as per

$$h(s_1, u) = y. \quad (2.19)$$

One advantage of the usage of State Machines is their graphical representation. In increasingly complex system automation, this view provides a good overview of the function and eases its design. The Latch mechanics already presented in Equation 2.9 can be expressed as a State Machine as well. Figure 2.3 is its graphical representation.

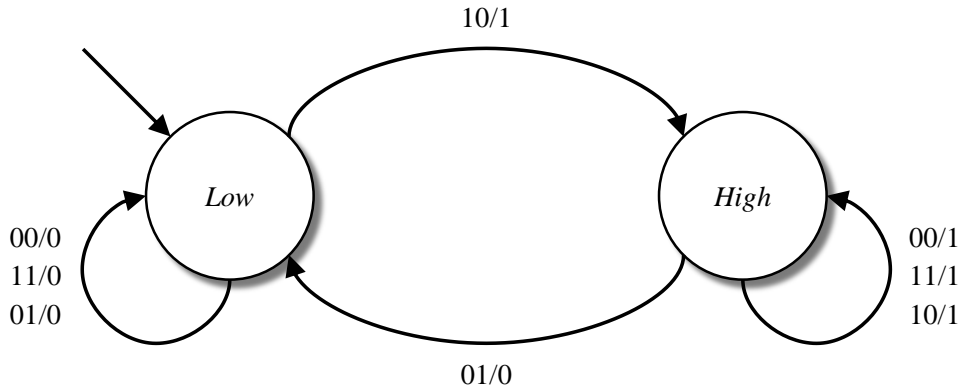


Figure 2.3: *A Latch as a Mealy Machine*

In a schematic of a State Machine, the nodes summarize all states the system (in this case the Latch) can assume. The names of the Latch states are chosen for ease of readability. In the provided example, the state *Low* is the initial state. The starting state is depicted with the edge without origin. The remaining edges together with the numbering next to the edges all illustrate the transition and output functions. The edge depicts the origin and destination states. The set of inputs that lead to that transition are listed on the left of the slash. On the right of the slash, the output value is shown. In the example, the top middle edge denotes that the system transitions from *Low* to *High* if *a* is *true* and *b* is *false*. Should this occur, the output is set to *true*. This corresponds to the transition and output function of

$$\delta(\text{Low}, \{\text{true}, \text{false}\}) = \text{High} \quad (2.20)$$

and

$$h(\text{Low}, \{\text{true}, \text{false}\}) = \text{true} \quad (2.21)$$

respectively.

2.2.3.5 Functional Allocation

The methods used to construct automation functions were explained in the previous sections. This section deals with the architectural composition of the automation. It explains where the different constructs of the previous section are used in an integrated manner in order to achieve a behavioral specification.

The flow of an automation function begins with the processing and evaluation of input data. Based on the performed evaluations, decisions are met within the system with regards to flight state, state of configuration, activation of contingencies, etc. Lastly, based on these decisions, actions are performed by automation, control concept and peripheral systems. The intent is communicated to the operator. This way of structuring is endorsed in [80] and is also adopted here.

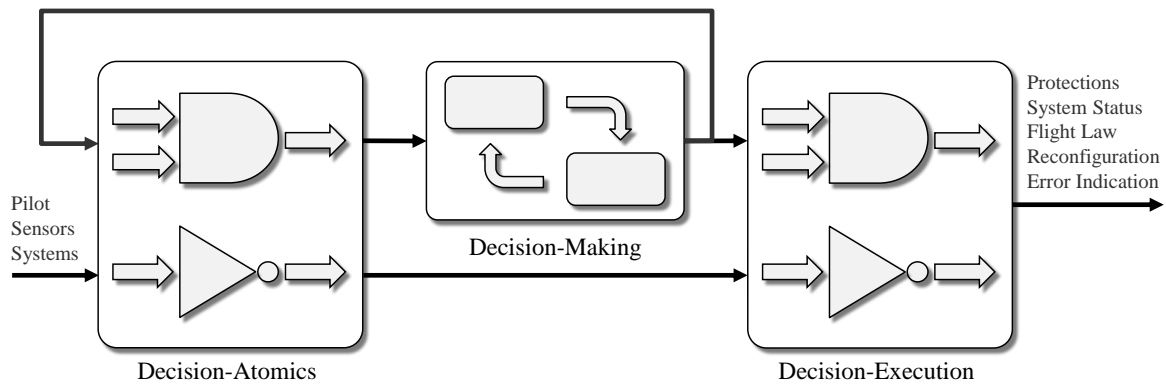


Figure 2.4: *Automation Functional Layout*

Three distinct functional elements are introduced, each of which handle the above-mentioned tasks - the Decision-Atomics, Decision-Making and Decision-Execution. The relationship between them can be seen in Figure 2.4. These elements are composed of the already introduced methods in Sections 2.2.3.2, 2.2.3.3 and 2.2.3.4. Each functional element is elaborated upon below.

The operations within the Decision-Atomics involve mostly combinational logic. They process all physical quantities entering the automation using Relational Operators. Boolean algebra is subsequently applied to the input information in order to create conditions for later use by the remaining functional elements. In addition to combinational logic, the Decision-Atomics may contain Confirmation Counters or Edge Detectors. The rationale for including these sequential logic elements are their simplicity and low dependency on other automation modules.

Thus, the Decision-Making module accepts solely the outputs of the Decision-Atomics. It uses the already synthesized information, keeping the decision-making process manageable by reducing the input space. This module contains the core of the automation and is composed of sequential logic, the majority of which are State Machines. Interactions between the State Machines are permitted only through the Decision-Atomics. This division of functions - preparation of data (Decision-Atomics) and evaluation of data (Decision-Making) - allows for modular design and decoupling of State Machines and therefore separate verification and validation.

The output of the Decision-Making are only the states of the constructs used. Even though complex output functions of the State Machines could be utilized, this is avoided. Instead, these computations are performed in the Decision-Execution. It takes the outcome of the Decision-Making and the Decision-Atomics and generates the data that needs to be distributed to all surrounding systems. This division maintains a degree of independence between automation design and peripheral systems. If necessary, alterations of the output behavior can be performed without manipulating the input evaluation and decision-making.

2.2.4 Automation Implementation Methods

The previous section introduced the theoretical constructs and their relation within an automation function. This section provides an overview of the implementation environment with which the automation module and the underlying methods are implemented. It is structured as follows.

Section 2.2.4.1 introduces the environment itself and all available tools and modeling practices that need to be taken into account. In order to illustrate the implementation patterns, an example is provided in Section 2.2.4.2. The desired response is elaborated in detail there. The specification in Section 2.2.4.2 is implemented in the next sections. In particular, Sections 2.2.4.3, 2.2.4.4 and 2.2.4.5 demonstrate how a Decision-Atomics, Decision-Making and Decision-Execution modules can be created to satisfy the automation behavior respectively.

2.2.4.1 Modeling Environment and Guidelines

At the TUM Institute of Flight System Dynamics the development of flight control algorithms is model-based and is performed in MATLAB/Simulink. Therefore, this software is utilized in this thesis. This environment supports the specification of high-integrity systems and applications intended for embedded systems. Guidelines on the former are available in [81]. In [82] a workflow together with permissible Simulink constructs are provided. They ensure a deterministic code generation process.

In terms of development processes and methods, a modeling guideline has been developed at TUM-FSD. For the creation of State Machines in particular, in [83] a method is published which fully complies to the above-mentioned guidelines. This thesis leans on the methods found in [83]. Using the example provided in the next sections, the relevant aspects of these sources are mentioned.

2.2.4.2 Example: Landing Gear Automation

The deployment automation of a landing gear is used as an example in order to illustrate the implementation patterns of the methods used in this thesis. This example is highly-simplified and the resulting implementation is not intended for use in real applications. Resilience against failures and other adverse scenarios are not under consideration. The example is instead tailored so that a large number of the methods of Section 2.2.3 can be utilized and is solely used for explanation purposes. This section provides a specification of the intended behavior, whereas the upcoming sections explain how this specification is achieved by design of the Decision-Atomics, Decision-Making and Decision-Execution.

The landing gear shall deploy and retract manually on operator input. This is only permitted when deployment or retraction is available. For the sake of keeping the example simple, those two inputs are assumed to be mutually exclusive, i.e. it is impossible to demand a simultaneous deployment and retraction command. Thus, the scenario of conflicting inputs is not in scope.

The retraction availability shall be established via weight on wheels sensors and the radar altimeter feedback. Retraction cannot be started if on ground or at low altitudes. The deployment availability shall be estimated solely on the airspeed - at too high dynamic pressures the deployment cannot be started to avoid structural damage.

In addition, the system shall include automatic retraction and deployment safety functions. The retraction shall be triggered if a particular airspeed is exceeded in order to protect the structure. The automatic deployment shall occur below certain velocities and heights.

The automation function shall communicate to the landing gear system whether to retract or deploy, which in this example is assumed to follow the instructions. In addition, the automation shall supply indication items that inform the operator what the current landing gear commands are. In addition, the pilot shall be made aware whether the deployment or retraction of the landing gear is unavailable due to the current flight condition via the indications.

2.2.4.3 Decision-Atomics

The implementation of the Decision-Atomics can be seen in Figure 2.5. Relational Operators where the input is compared to predefined parameters are done using “Compare to Constant” Simulink blocks. Otherwise, “Relational Operators” Simulink blocks are used to compare two signals against each other. The latter is not depicted in the figure. Boolean algebra is performed with “Logical Operator” blocks. In the example, the airborne state is determined with two out of three weight on wheels sensors not registering ground contact.

In addition, the naming of the resulting boolean signals must follow the convention of TUM-FSD as found in [77]:

- “*_flg*” is used to signify a boolean that is triggered if a property is fulfilled or not
- “*_cfg*” is used for edges
- “*_rfg*” is used for rising edges
- “*_ffg*” is used for falling edges

2.2.4.4 Decision-Making

The Decision-Making in this particular example contains only one State Machine. In MATLAB/Simulink, this is created via the Stateflow toolbox’s “Chart” block. It is illustrated in Figure 2.6. The underlying design patterns and development guidelines stem

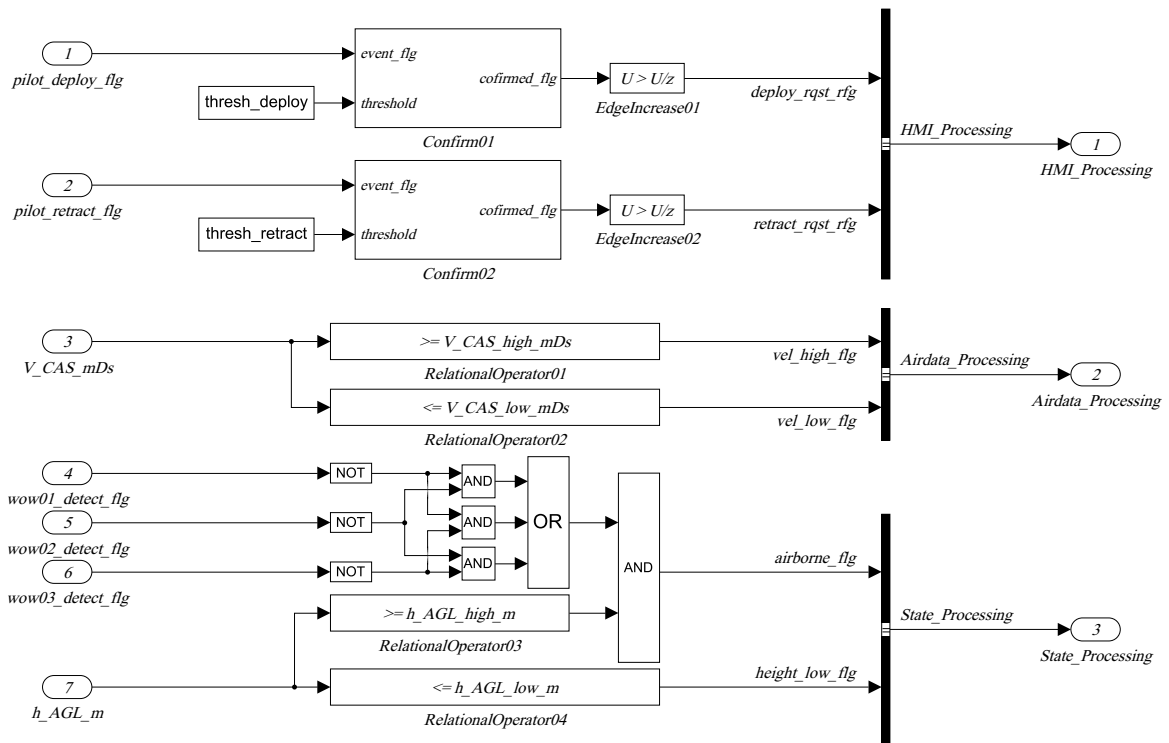


Figure 2.5: *Decision-Atomics - Implementation Example*

from [83]. In [77, 83], the author presents a method of creating high-degree of automation modules that contain nested State Machines (sub-states). The methods are both compliant with the guidelines of TUM-FSD but introduce a structure where the automation can be created in a modular manner to alleviate and manage complexity. Implementation examples are available in [77, 83, 84]. The important aspects are summarized here.

Finite-State Machine states are established with Simulink Enumerated types. This is in accordance with [77]. In this example the states belong to the set $S_{LG} = \{Deploy, Retract\}$ as seen in Figure 2.6 where the State Machine implementation is visible. The following implementation rules apply:

- At the entry point, the output - the state of the State Machine - needs to be assigned.
- Recall that for deterministic State Machines, transition conditions for all combinations of input tuples need to be specified. In Stateflow charts this is not necessary, as it is managed by the modeling environment [77].
- Transition conditions need to be expressed only on horizontal edges [77]. Those are indicated within square brackets as seen in Figure 2.6. The link between transition conditions and transition functions is established in the paragraphs below.
- Output actions need to be expressed only on vertical edges [77]. Those are indicated within curly brackets as seen in Figure 2.6.

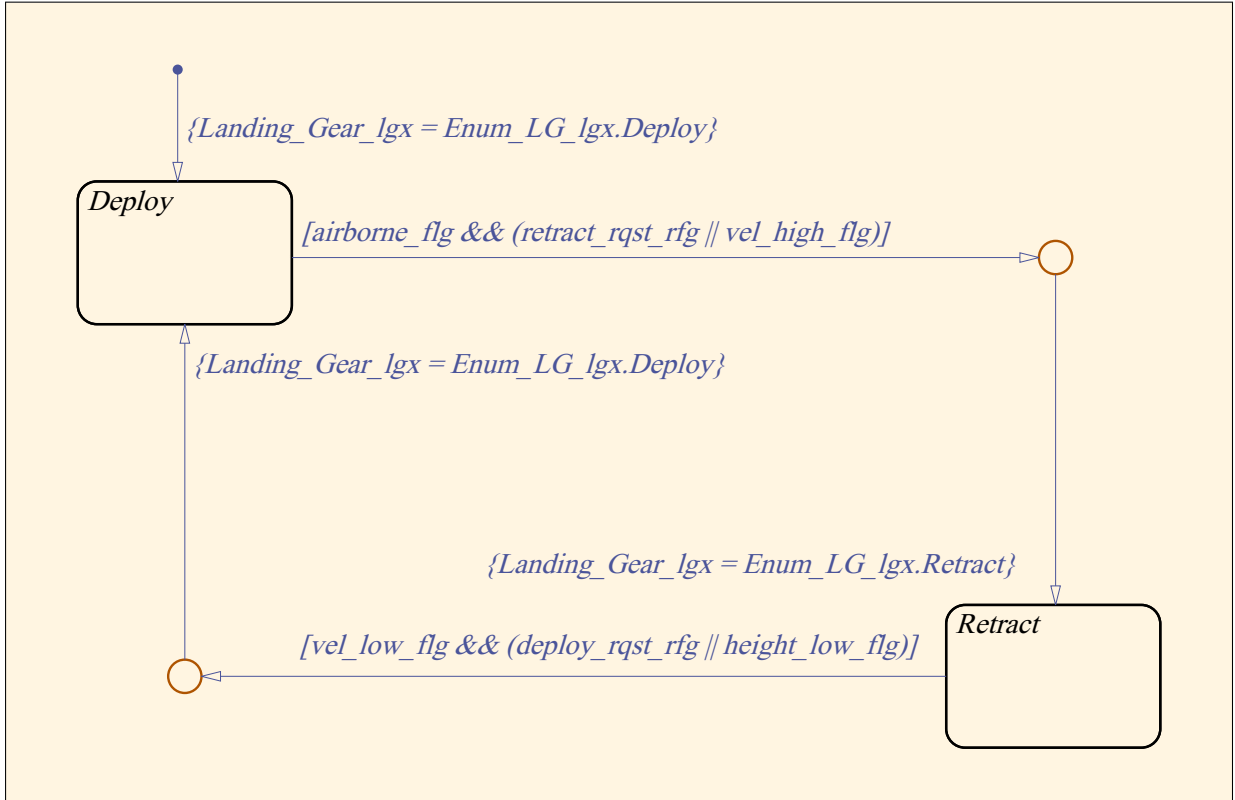


Figure 2.6: *Decision-Making - Implementation Example*

- If the automation includes a State Machine, composed of sub-State Machines, then each level shall be contained in a switch-case which is preceded by the higher-level state. Refer to [83] and [77] for more information.

The implementation seen in Figure 2.6 fulfills the following specification. Firstly, the state of the automaton is specified as $s_{LG} \in S_{LG}$. S_{LG} was already defined in this section. Furthermore, $s_{LG_0} = Deploy$ is the starting state. All inputs of the State Machine belong to the boolean domain and are visible in Figure 2.6. They are the outputs of the Decision-Atomics of Figure 2.5. The output is $Landing_Gear_lgx \in S_{LG}$ and represents the current state of the Mealy Machine. The transition function that allows a transition from state *Deploy* to *Retract* is then expressed as

$$\delta(Deploy, u_{LG_1}) = Retract. \quad (2.22)$$

In the equation u_{LG_1} is any member of the set of input tuples $T_{LG_1} \subset U$, for which

$$t_{LG_1} = airborne_flg \wedge (retract_rqst_rfg \vee vel_high_flg) \quad (2.23)$$

is *true*. t_{LG_1} is referred to as the “transition condition” and is visible in Figure 2.6. From here it follows that

$$\delta(Deploy, u_{LG_1}^c) = Deploy, \quad (2.24)$$

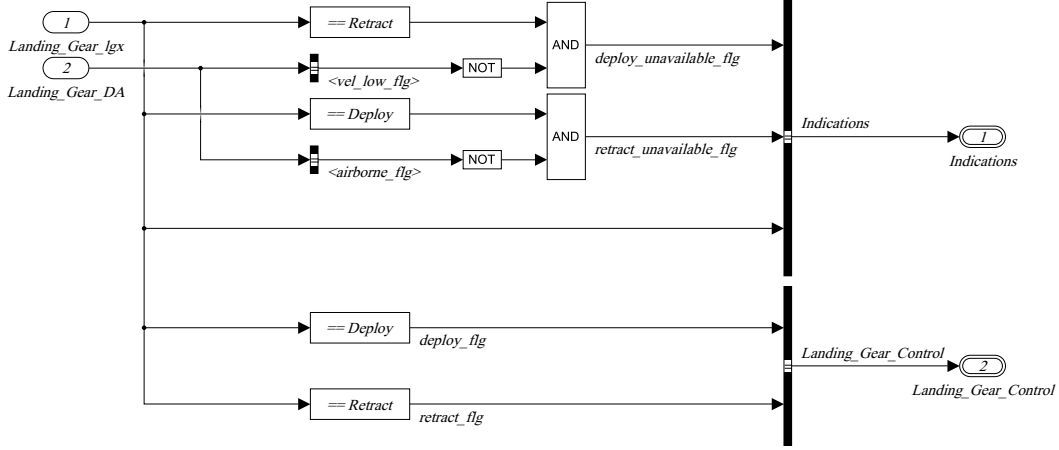


Figure 2.7: *Decision-Execution - Implementation Example*

where $u_{LG_1}^c$ is a member of the complement set $T_{LG_1}^c$. This implies that the state is retained for all input combinations that cause $\neg t_{LG_1}$ to be *true*. For the sake of readability, the State Machines in this thesis are described with the transition conditions instead of listing all input combinations that satisfy this condition.

Analogously, it can be stated that $u_{LG_2} \in T_{LG_2}$ are all inputs, for which the transition condition

$$t_{LG_2} = vel_low_flg \wedge (deploy_rqst_rfg \vee height_low_flg) \quad (2.25)$$

is *true*, then

$$\begin{aligned} \delta(Retract, u_{LG_2}) &= Deploy, \text{ and} \\ \delta(Retract, u_{LG_2}^c) &= Retract. \end{aligned} \quad (2.26)$$

t_{LG_2} is also visible in Figure 2.6 This fully specifies all transition functions. Because solely the state is fed out, for every input combination. The output functions can be written as

$$h(s, u) = \delta(s, u). \quad (2.27)$$

For the sake of completeness, in Appendix A the relationships between transition conditions, functions and sets are explained in more detail.

2.2.4.5 Decision-Execution

Figure 2.7 illustrates how a Decision-Execution utilizes the output of the Decision-Atomics and the decision making to supply the two necessary systems for this example. The constructs utilized are the same as in the Decision-Atomics system. It is important to note that each component that the automation module is required to communicate with receives all necessary data via a dedicated data structure. This omits the possibility of signals misinterpretation.

This concludes the theoretical and mathematical preliminaries on the automation function design. As illustrated in Section 2.2.1, understanding of the hazards and characteristics of the underlying system is of key importance when substituting the operator tasks with automated processes. These aspects are covered in the next sections of this chapter.

2.3 Structural and Performance Considerations of eVTOL Lift-to-Cruise Aircraft

The design of every aircraft is optimized in accordance to its mission profile. It involves the choice and placement of components in order to satisfy performance metrics derived from that profile. The resulting airframe has specific structural and dynamical characteristics that need to be considered in the subsequent software design phase.

This section analyzes and breaks down the main system components of eVTOL lift-to-cruise aircraft that directly influence the automation design. Each section first derives the relevant underlying physical phenomena and proceeds to explain their impact on the aircraft level. If necessary, this is done for relevant failure modes as well. Section 2.3.1 examines the distributed propulsion system and how its operation needs to be managed by the automation. The same is done for the high-lift system in Section 2.3.2. Lastly, Section 2.3.3 explains what properties of the traction system need to be accounted for in the automation function design.

2.3.1 The Properties of Distributed Hover Propulsion

The distributed propulsion of a lift-to-cruise aircraft is a system that is composed of multiple propulsion units. In order to understand the role the whole system plays in the automation design considerations, firstly the individual unit is examined in Section 2.3.1.1. Thereupon, the system as a whole is studied in Section 2.3.1.2.

2.3.1.1 Individual Distributed Propulsion Unit

In this section, the properties of the distributed propulsion units are examined. In MOC SC-VTOL [10], EASA refers to the effectors, responsible for vertical thrust as Lift/Thrust Unit (LTU). This terminology is utilized in this thesis as well. Understanding certain key properties of the common LTUs used on the lift-to-cruise aircraft is of importance in this thesis as it has implications on the automation function design.

According to [85], for lift-to-cruise typically “Multirotor-style” LTU constellations are used. These types of units are composed of a rotor, inverter and propeller. The state of technology on LTUs for lift-to-cruise aircraft incorporates a brushless direct current motor as the electric rotor due to its advantageous dynamic characteristics, high speed

range and endurance [86]. The rotor, together with the inverter, are responsible for the torque generation, used for speed and position control. This is established by commutation, i.e. sequential current supply of the different motor coils that are grouped in so-called motor phases. For this, knowledge of the rotor position is necessary in order to shape the magnetic field properly. The choice of control strategy is dependent on the usage of an external position measurement.

Without an external measurement, the control is referred to as “sensorless” [87]. The knowledge of the rotor position is established by Counter-Electromotive Force (Back-EMF) [88]. Back-EMF originates from the rotor rotation and therefore at low rotational speeds or at standstill, the intensity of the Back-EMF is either too small or non-existent to be detected by the electronics. Due to the lack of the rotor position knowledge during rotor standstill, the movement of the component is typically initiated by injecting predefined sequences [87–89]. In addition, at low revolution speeds, the dynamics are much different with relation to the remainder of the rotational speed envelope [90]. Given external position measurement, such shortcomings can be omitted at the expense of mechanical complexity and cost [88].

In contrast to helicopter applications where the propeller blades are attached to the shaft with hinges and their pitch is controlled while the rotation speed is held constant [91], the propellers of the currently envisioned LTUs for lift-to-cruise aircraft are mounted rigidly and the pitch is constant [85]. This is because the propeller itself is comparatively small (i.e. has lower inertia) and the above-mentioned electric rotor dynamic response and large speed range allow for Revolutions per Minute (RPM) control. This omits a large portion of the mechanical complexity, attributable to the variable pitch control. In this thesis, the term “idle RPM” is used often. This term refers to the LTU RPM, above which the unit response is deterministic, i.e. the operation of the LTU is outside the engagement region with low rates of revolution where a potential turn-on sequence may be injected.

According to [92, 93], for multicopter aircraft, the propeller thrust and torque for rigidly mounted propellers is modeled as

$$F = k_F \omega^2 \quad (2.28)$$

$$T = k_T \omega^2 \quad (2.29)$$

respectively, where k_F and k_T are propeller thrust and moment coefficients that include the air density and the propeller surface and chord length. ω is the revolution rates of the propeller. This relationship is useful to understand that the force and torque production are linearly dependent on the square of the rotor RPM. However, this modeling is only valid for operation near hover, thus it makes the assumption that the freestream is negligible.

For lift-to-cruise aircraft that can reach significantly higher airspeeds, the freestream needs to be considered as well. Such a scenario is depicted in Figure 2.8. According to [94], for freestream velocities solely parallel to the rotor disk, the averaged out force and

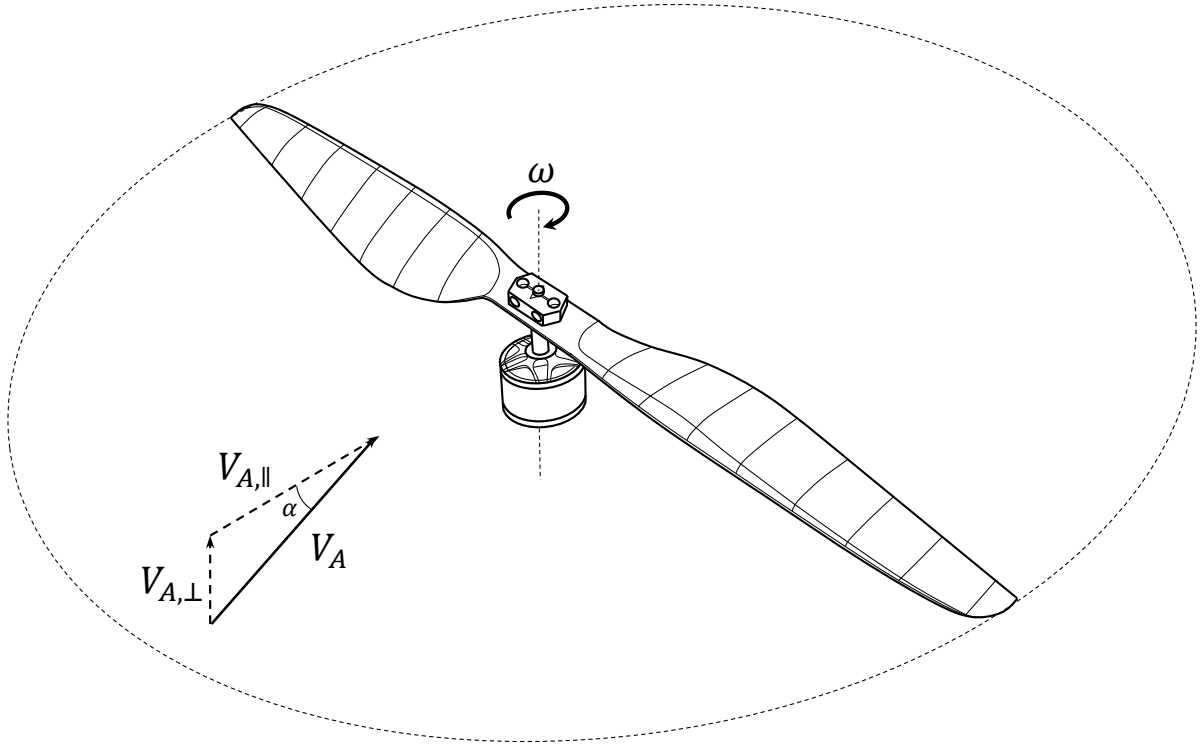


Figure 2.8: *A Lift/Thrust Unit in an Arbitrary Freestream*

torque of the LTU are

$$F = \frac{1}{2} \rho c C_L \left(\frac{2R^3}{3} |\omega|^2 + |V_{A,||}|^2 R \right) \quad (2.30)$$

$$T = \frac{1}{4} \rho c C_D \left(R^4 |\omega|^2 + |V_{A,||}|^2 R^2 \right). \quad (2.31)$$

In the equations, ρ denotes the air density, C_L and C_D are the lift and torque coefficients of the propeller, c is the chord length, R is the radius of the propeller and $V_{A,||}$ is the freestream velocity. The nomenclature in the equations is adapted from [94] to fit the convention, used at TUM-FSD. The above-mentioned equations are used in order to illustrate the underlying physical effect and the impact they may have on the LTU. For constant densities and no freestream, the coefficients in Equations 2.28 and 2.29 can be clearly linked to the parameters in Equations 2.30 and 2.31.

Another effect resulting from the freestream is that the leading propeller blade has an increase in the incoming dynamic pressure and the retreating blade has a decreased one. In other words, the freestream velocity $V_{A,||}$ causes a moment. The velocity component is depicted in Figure 2.8. The result is an oscillatory bending moment profile, according to [94] with an average load of

$$\tau = \frac{1}{2} \rho c C_L R^3 |\omega| |V_{A,||}|. \quad (2.32)$$

The nomenclature in the equation is adapted from [94] to fit the conventions, used at TUM-FSD. The above-mentioned equation is used in order to illustrate the underlying physical effects and the impact they may have on the LTU. The interested reader is advised to [94], where in addition the oscillatory load profile is simulated for a small propeller model.

Solely axial inflows $V_{A,\perp}$ change the effective angle of attack of the propeller blades which in turn leads to thrust and torque effectiveness changes that are constant over the propeller revolution [94]. Derivations and experimental results found in [95] and [96] conclude that under non-axial inflow conditions both effects from above are observed.

The important observation is that the load oscillation scales with the power of three with regards to the propeller radius and linearly with the freestream velocity. For large LTUs that operate at high dynamic pressures this may lead to significant oscillatory load.

Knowing the above mentioned properties of the electric drive and the propeller aerodynamics, failure modes of the LTU operation that impact the automation function design are identified.

1. **Total loss:** Regardless of the mechanical and electronic design, a total failure cannot be excluded. If this does not involve breakage of the propeller, but loss of electric power, then in addition windmilling can occur after the total loss. This is due to the torque that is induced by the propeller inflow and the lack of counteracting moment from the electric drive.
2. **Partial Loss of Torque:** Certain electromechanical faults can cause improper or complete lack of current supply to a set of motor coils. As a consequence, the strength of the magnet field is compromised and therefore the maximum available motor torque is reduced.
3. **Stuck at Value:** Depending on the software design of the motor controller, in the case of communication loss, the LTU may retain the last received RPM command. This failure mode is less likely. Often, the motor is disengaged if no new command message is received within a predefined time frame. This method is often preferred as it guarantees a certain degree of determinism. For the sake of completeness, the stuck at value mode cannot be excluded.
4. **Failure of In-Flight Engagement:** This failure is driven by the vastly different load conditions than typical on-ground engagement. The in-flight activation occurs at high aerodynamic pressures (above the aircraft stall speed) and non-negligible inclination angles. As a consequence of the difference in load profile, the starting of the motor may malfunction. The likelihood is dependent on the motor control strategy used.²

²In an industry project at TUM-FSD, this failure mode was experienced during reconfiguration to powered-lift mode in flight tests.

2.3.1.2 The distributed propulsion system

The distributed propulsion system of a lift-to-cruise aircraft is composed of multiple LTUs. They are aligned such that the generated thrust is parallel to the aircraft body z-Axis, though sometimes slight inclinations are foreseen in order to maximize yaw authority. The force generated from the propulsion system as a whole is the sum of all generated propeller forces. Those forces produce roll and pitch moments due to the distance of the LTUs from the center of gravity. Yaw authority is ensured by arranging the propeller spin directions accordingly [93]. The demonstrator aircraft presented in Section 1.1.4 has a distributed propulsion system that is composed of eight LTUs.

Due to the ineffectiveness of control surfaces at low dynamic pressures, a lift-to-cruise aircraft relies solely on the hover propulsion system for adequate control authority. As a consequence, the system as a whole requires to be fail-active even though the individual LTUs may be fail-open or fail-passive. For this reason the hover propulsion system is realized redundant and is therefore over-actuated. Failures in one or more LTUs will result in loss of attainable moment set and control performance loss [97][98] but controllability is ensured.

When not operational, at high dynamic pressures the propellers of the propulsion system are aligned in direction of either the aircraft body x-Axis or - if available - with the inflow direction. This minimizes the overall aircraft drag.

When operational, at increasing dynamic pressures and changing angles of attack, the load explained in Section 2.3.1.1 is transferred to the motor shaft and thus to the surface it is attached to. As a consequence, the beams the LTUs are attached to are subject to material fatigue [99]. This leads to the introduction of the so-called “lift-system operation never exceed speed” V_{LSNE} . This is defined as the airspeed of the vehicle, above which thrust generation from the propulsion system may cause structural damage to the airframe.

2.3.2 The Properties of the High-Lift System

The high-lift system is a mechanism that is attached to the aircraft primary lifting surface. Typically, it is composed of flaps and sometimes slats. An overview of possible configurations is provided in [100]. Though the designs of high-lift systems vary significantly, the purpose of deployment is to change the aerodynamic characteristics of the wing and namely to increase the maximum lift coefficient $C_{L,max}$ of the aircraft [100].

Effectively, the increase in maximum lift coefficient lowers the stall speed. Having a reduced stall speed is necessary “if the natural value of $C_{L,max}$ for an airplane is not high enough for safe takeoff and landing” [100]. Wingborne take-off or landing is not necessarily the envisioned use-case for UAM lift-to-cruise VTOL. In prototype stages, however, mitigation strategies may be pursued. In such events, it may be required for the test aircraft to land in wingborne flight. More importantly, a lower stall speed may be required in order to execute the reconfiguration from powered-lift to wingborne flight and

back at lower dynamic pressures. The latter argument is the reason for the incorporation of a high-lift system in the demonstrator aircraft of this thesis, introduced previously in Section 1.1.4. The system is composed of a plain flap on each primary lift surface. In this thesis, the terms “High-Lift System” and “Flaps” are used interchangeably.

When a high-lift system is incorporated into the design, then the aircraft state of configuration is broken down into two classes based on the current deployment of the flaps. If fully retracted, then the configuration is “clean”, otherwise it is referred to as “dirty” [101]. The aircraft stall speed for the clean configuration is denoted as V_{STALL} for the dirty configuration with fully deployed high-lift system as $V_{STALL_{FE}}$.

Utilization of a high-lift system has disadvantages and carries challenges. Apart from the added complexity, the deployment of the system significantly increases the aircraft drag. In clean configurations the structural limit speed is denoted as V_{NE} so as to guarantee consistency with the definitions of the Code of Federal Regulations (CFR), and namely in 14 CFR 1.2 “VNE” [102]. In contrast, for dirty configurations the allowed maximum airspeed is lower than V_{NE} . The reason for this is that deflected flaps are prone to structural damage due to the aerodynamic loads they encounter. 14 CFR 1.2 “VFE” [102] defines this speed as the “maximum flap extended speed”. Within the thesis, this speed is denoted as V_{FE} . V_{FE} is a function of the current flap setting [101].

2.3.3 The Properties of the Traction System

In this section the rudimentary characteristics of the traction system that play a role in the design of automation functions are expanded upon. The findings of this section are used later in this thesis. For lift-to-cruise aircraft, the traction system is responsible for generating the forward thrust necessary for achieving and maintaining wingborne flight. Due to the airspeed range that UAM aircraft are envisioned to operate, the majority of currently developed lift-to-cruise eVTOL aircraft have a propeller-drive traction system and so does the demonstrator configuration for this thesis.

One typical off-nominal scenario in UAM use-cases is the avoidance of obstacles in ground proximity. The capability of avoiding an obstacle longitudinally is a geometric problem, where in order to gain the most clearance, the flight path angle needs to be maximized. For wingborne flight this occurs in the so-called “speed for best climb” V_X [102]. For propeller-driven aircraft, the calculation of V_X can for example be found in [101]. The speed for best climb is important for the automation functions because reconfiguring to wingborne flight prior to attaining V_X may excessively limit the aircraft’s capability of obstacle clearance.

In order to increase the availability wingborne flight, typically the traction system is composed of multiple propulsion units. In this thesis, they are referred to as Traction Thrust Unit (TTU). The traction system of the aircraft presented in Section 1.1.4 is

composed of two units, mounted symmetrically on each end of the elevator. Because a failure of one propulsion unit cannot be excluded, the following properties need to be considered.

A TTU does not only produce forward thrust, but also a yaw moment due to the lateral offset of that force from center of gravity. Nominally, the presence of two traction units on each side of the aircraft cancels out the moments.

However, when one of traction unit fails, then the yaw moment of the opposite unit needs to be counteracted by other means in order to sustain steady-state flight. For wingborne flight, this is achieved with the rudder. However, the effectiveness of the control surface scales linearly with the aerodynamic pressure. Therefore, below a given airspeed, the moment cannot be counteracted by the rudder alone.

Motivated by this characteristics, the code of federal regulations defines the so-called “minimum control speed with the critical engine inoperative” in 14 CFR 1.2 “VMC”. In 14 CFR 25.149(b) it is elaborated that for conventional fixed-wing aircraft “[V_{MC}] is the calibrated airspeed at which, when the critical engine is suddenly made inoperative, it is possible to maintain control of the airplane with that engine still inoperative and maintain straight flight with an angle of bank of not more than 5 degrees” [103].

In climb, this effect is exacerbated because it requires more force production from the traction units and hence more yaw moments are produced. Similarly to V_X , in 14 CFR 25.111 [104] and 14 CFR 25.121(b) [105] the velocity of V_2 is introduced. It is defined as the velocity, at which minimum climb gradient has to be maintained with an engine inoperative. The value of the gradient depends on the number of engines. For the calculation of V_2 , the interested reader is advised to [106].

At speeds lower than V_{MC} or V_2 , in lift-to-cruise aircraft in addition to the rudder, the control authority is maintained with the hover propulsion system. For this reason, the magnitudes of these velocities need to be considered in the automation function design of the reconfiguration to wingborne flight.

The structural topology of an eVTOL aircraft and the utilized physical systems impose hard requirements on the automation specification. One other origin of considerations are the design choices in terms of control concept and operator interaction via the Human-Machine-Interface. These topics are covered in the following section.

2.4 Aspects of the eVTOL Aircraft Operation and Control Design

As seen in 2.2.1, the control concept and the interaction with the operator play a key role in the automation function operation. The integration of control concept and automation needs to be coherent, seamless and intuitive. On the other hand, the need of a human-centered approach to the design of automation modules requires knowledge and consideration of the crew input items. As a consequence, these aspects need to

be analyzed thoroughly because they impose design decisions on the automation. This section summarizes the important topics of control design and operation that impact the automation module.

From **Contribution 1** and **Contribution 2** it is implied that two Simplified Vehicle Operations control concepts exist that are operated on the vehicle presented in Section 1.1.4. The latter control strategy implements an SVO1 whereas the former implements an SVO2. The levels of Simplified Vehicle Operations were presented in Section 1.2.2. For recollection, SVO1 is the fallback of SVO2.

The automation functions of this thesis operate with the two concepts. Therefore the SVO2 and SVO1 control strategies are presented in Section 2.4.1. The summary focuses on the characteristics of their interaction with the automation.

As seen in the analysis of Section 2.4.1, the notion of eVTOL aircraft-specific flight phases persists in both of the concepts. In Section 2.4.2 those relevant flight phases are formally defined. Lastly, the operator input items are presented in Section 2.4.3. In particular, the division of the input items with relation to the eVTOL aircraft flight phases is broken down.

2.4.1 Simplified Vehicle Operations

In this section, the two aircraft control concepts are presented. Firstly, the Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics is presented in Section 2.4.1.1 and is followed by its Fallback concept in Section 2.4.1.2. Available literature on the topic include [8] and [107] respectively. Both sections focus on the properties of those control algorithms, the pilot authority, the control allocation and possible envelope protections. Finally, the sensor information dependency for both control concepts is summarized in Section 2.4.1.3.

2.4.1.1 Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics

The topic of control of aircraft, capable of reconfiguration from powered-lift to wingborne flight, has been a topic at the TUM Institute of Flight System Dynamics for more than five years. A control concept for highly-automated flight for such a configuration is published with [108–112]. The application is an unmanned aircraft with a take-off mass of approximately five kilograms. The control concept is tailored to operators with little to no flying experience. As a consequence, the operation includes input elements that remain consistent in their pilot interpretation throughout the whole flight envelope. The control algorithm involves no manual input for reconfiguration and is described as “unified” in [110].

Since then this approach has evolved and was tailored for manned flight. Protections were introduced, such that safety of flight was increased. This concept is referred to as Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics (FSD-SVO). In this section the relevant characteristics of the FSD-SVO are analyzed and the information it requires to adequately function within the context of the eVTOL operation.

FSD-SVO presents an intuitive control strategy that directly addresses the workload reduction by well defined control variables that on high-level are consistent in every flight phase. Relevant publication on the concept can be found in [8, 34, 113]. Here the important control elements along with their dependency and requirements with respect to the automation functions are analyzed.

Error Controller and Command Variables

The error controller of the FSD-SVO differentiates between three main operating modes - *HV*, *TR* and *WB*. In this thesis they are also referred to as “hover”, “transition” and “wingborne”. The former refers to an envelope, in which the aerodynamic effects from the lifting surfaces are negligible, whereas in the latter the airspeed is sufficient to sustain flight solely using aerodynamic lift. The “transition” mode bridges the two modes.

Even though the notion of these flight phases is mentioned, the underlying algorithm is a unified structure that is utilized throughout the whole aircraft flight envelope. The phases mentioned in [8], notated also as “Behavioral Modes” are rather necessary from the perspective of the tracked variables and the interpretation of the control inceptors. Those are blended so there is no discrete switch in the inceptor interpretation or the generated pseudocontrols.

The command variables blending over the airspeed and modes are summarized in Figure 2.9. The flight phases are visible along the x-Axis. The different command channels are along the y-Axis. The main observation is that the different command channels always generate commands that have the same high-level response throughout the extended flight envelope.

The heading is tracked as a pure yaw motion in hover and over the airspeed is blended to a bank to turn during the transition phase. For hover, the kinematic speed is tracked. This allows for high-precision maneuvering with respect to the ground, crucial for precision landing. This phase allows for a maximum longitudinal and lateral kinematic velocity command, indicated with V_{HOVER} . In the transition mode, the command variables are blended such that at the end of the blending solely the aerodynamic speed is tracked.

For correct operation, the law expects information with regards to the currently applicable flight phase. This is critical not only to assign the correct command variables but also to specify the ranges of the permissible command variables. For example, in hover


$\mathbf{V}_{Cy,d}$ $\rightarrow V_{Cy,c}$	$\beta_{\text{kin},d}$ $\rightarrow V_{By,c} = \sin(\beta_{\text{kin},d})V_{\text{kin}}$	$f_{By,d}$ $\rightarrow f_{By,c}$			
$\mathbf{V}_{Cx,d}$ $\rightarrow V_{Cx,c}$	$\mathbf{V}_{\text{CAS},d} + \alpha_{\text{kin},d}$ $\rightarrow V_{\text{CAS},c} + \alpha_{\text{kin},c}$	$\mathbf{V}_{\text{CAS},d}$ $\rightarrow V_{\text{CAS},c}$			
$\dot{\psi}_d$ $\rightarrow \dot{\psi}_c$	$\dot{\psi}_d$ $\rightarrow \phi_c = \sin^{-1} \left(\frac{\tan(\theta) \dot{\psi}_d V_{Bz}}{\sqrt{g^2 + \dot{\psi}_d^2 V_{Bx}^2}} \right) - x$ $x = \sin^{-1} \left(\frac{-\dot{\psi}_d V_{Bx}}{\sqrt{g^2 + \dot{\psi}_d^2 V_{Bx}^2}} \right)$	$\dot{\psi}_d$ $\rightarrow \phi_c = \sin^{-1} \left(\frac{\tan(\theta) \dot{\psi}_d V_{Bz}}{\sqrt{g^2 + \dot{\psi}_d^2 V_{Bx}^2}} \right) - x$ $x = \sin^{-1} \left(\frac{-\dot{\psi}_d V_{Bx}}{\sqrt{g^2 + \dot{\psi}_d^2 V_{Bx}^2}} \right)$			
\dot{h}_d $\rightarrow \dot{h}_c$ (Powered Lift)	\dot{h}_d $\rightarrow \dot{h}_c$ (Powered Lift)	\dot{h}_d $\rightarrow \begin{cases} \dot{h}_c & \text{Small Amplitudes} \\ V_{\text{CAS},c} & \text{High Amplitudes at Maximum Thrust} \end{cases}$ (Aerodynamic Lift)			
Hover	V_{hover}	Transition	V_{stall}	Wingborne	

Figure 2.9: *FSD-SVO Command Variables in the different flight phases. The Image Stems From [8].*

the vertical velocity is limited so as to avoid the possibility of the powered-lift system running into its own vortices. Similarly, the load factors are scheduled throughout the envelope.

Logically, the automation of the transitions from powered-lift to wingborne flight and backwards involve an acceleration forwards and backwards respectively. This is facilitated by the law itself, however, the automation is the instance that executes the reconfiguration. Therefore, the forward velocity command channel in particular needs to be observed for the correct automation design. The interpretation of the control inceptor that corresponds to that channel needs to be harmonized among automation and law. With regards to the control laws, two positions are distinguished.

The first position on the velocity command channel is the one where the command, corresponding to V_{HOVER} , is requested in order to distinguish between the two flight phase requests - hover and transition. The same argumentation is followed when wanting to distinguish between the transition and wingborne phase of the law. In particular, the so-called “safe” speeds are introduced in the law. They are defined as

$$V_{SAFE} = V_X \text{ and} \quad (2.33)$$

$$V_{OEI} = V_2 \quad (2.34)$$

for the clean configuration and have the index “*FE*” for the configuration with deployed flaps. The velocities the safe speeds are computed with were previously introduced in Sections 2.3.3 and 2.3.2. The importance of the two safe speeds of Equation 2.34 and V_{HOVER} becomes apparent later on in Chapter 4.

Control Allocation

The task of the control allocation within the control concept is to calculate effector commands in order to achieve the pseudocontrols, required by the error controller. The allocation strategy, pursued for the FSD-SVO, follows the methods published in [112]. The control allocation is robust with regards to an effector failures, tolerating up to one failure in the hover propulsion and actuation.

The distribution of the control allocation follows the same behavioral modes as the law. In the hover mode, the pseudocontrols are fully generated with the hover propulsion system, whereas in the transition phase the control surfaces gain prioritization with increasing airspeed. In the wingborne phase, the hover propulsion system is gradually driven down and solely the control surfaces are used.

As a consequence, the control allocation needs to receive the mode information from the automation functions that implement the automatic transition from powered-lift to wingborne flight and back. In addition, the automation imposes a requirement on the control allocation for engagement or disengagement of the powered-lift system. The importance of this is to guarantee the availability of powered-lift flight as is explained later in Chapter 3.

Lastly, although the control allocation does not provide commands to the high-lift system, the state of the flaps must be provided to the allocation by the automation. The reason for this is that knowledge of the aerodynamic effects facilitate the computation of feasible effector commands.

Protections

As per the definition of SVO2 found in Section 1.2.2, the control concept of FSD-SVO includes mechanisms in order to limit the operator authority in the cases where structural limits are exceeded or potentially unstable flight envelopes are entered.

FSD-SVO includes an angle of attack protection in the wingborne phase, which ensures that flow separation does not occur. At low horizontal kinematic speeds, it does not allow high sink rates so as to ensure that the no propeller enters the vortices, generated by the hover propulsion system. Structural limits in terms of maximum load factor are maintained as well. All above mentioned protections are activated via the behavioral mode.

In addition to these protections, the SVO concepts includes underspeed and overspeed protections. They can be enabled and disabled by the automation function. In addition the values of both protections can be set by the automation. These protections are of particular importance, as described later on in Chapter 3.

2.4.1.2 Fallback Control Concept

The Fallback implements an SVO1 control concept in the events where a takeover is necessary. By the definitions already introduced in Section 1.2.2, the operator must have significantly higher control authority in order to ensure safe flight. Therefore, the Fallback concept includes significantly less automation. The design of the concept is explained in detail in [107]. Here the important elements along with their dependency and requirements with respect to the automation functions are analyzed.

Law and Flight Modes

The Fallback concept implements two separate control elements, responsible for the two distinct configuration states - powered-lift and fixed-wing flight. The powered-lift mode includes two distinct operating modes - hover and transition. Fixed-wing mode is referred to as “wingborne” in [107].

The command variables of the Fallback principle can be found in Table 2.6. When compared to the FSD-SVO of Figure 2.9, it is evident that again four command channels are utilized. The interpretation of the channels is similar and can be seen as a degradation of the FSD-SVO.³

³As seen in [107], the law includes multiple different degradation stages. They are not in the scope of the thesis.

Table 2.6: *Fallback Command Variables in the Different Flight Phases*

HV	TR	WB
Bank Angle	-	-
Traction Thrust, Pitch Angle	Traction Thrust Pitch angle	Traction Thrust
Yaw Rate	Bank Angle	Bank Angle
Body Normal Vertical Acceleration	Body Normal Vertical Acceleration	Pitch Angle

The bank angle in hover causes a lateral acceleration that builds up to a lateral velocity - the variable that is commanded in the FSD-SVO. It can be noticed that in the transition and wingborne phase this channel has no meaning, whereas in the FSD-SVO it remains utilized. As seen later, in the transition phase and wingborne phase the command channel itself no longer exists.

Instead of velocities, the second command channel in the Fallback control concept specifies a traction thrust demand. This eventually leads to a steady-state velocity. The control magnitude is in the responsibility of the pilot.

The channel that uses the yaw rate in hover is reconfigured to “bank to turn” for all other flight phases. Lastly, the Fallback law does not control the vertical speed, but instead tracks a vertical acceleration. The difference with regards to the FSD-SVO is that for both the channels the pilot is responsible for maintaining the desired course rate and vertical speed.

The law of the Fallback principle requires the information with regards to the control elements that should be engaged in order to provide adequate the command variable mappings as per Table 2.6.

Control Allocation

Comparing the control allocations of FSD-SVO and its Fallback, the significant difference is that the Fallback allocation utilizes the control surfaces at all times - even in the hover phase where they have limited efficiency. The automation is required to supply the allocation with information about malfunctioned LTUs or control surfaces.

In powered-lift flight, the hover-propulsion system is utilized in order to ensure command variable tracking. In the wingborne mode, those motors are driven down to a halt. Similarly to the FSD-SVO, the automation imposes a requirement on the control allocation for engagement of the powered-lift system in the wingborne flight. This ensures a smooth switch to powered-lift flight. This is explained later in Chapter 4.

Table 2.7: *Overview of the Law Sensor Dependency*

Sensor Data	FSD-SVO	Fallback
Body Rates	✓	✓
Body Accelerations	✓	✓
Attitude	✓	✓
Airspeed	✓	(✓)
Angle of Attack	✓	
Kinematic Speed	✓	

2.4.1.3 Signal Dependencies

This section provides an overview of the necessary information of both control concepts. Firstly, the required data from the automation that was elaborated upon in Sections 2.4.1.1 and 2.4.1.2 is summarized. Next, the control concept dependency on external sensor information is depicted.

In terms of data from the automation module, both FSD-SVO and Fallback require knowledge of the active flight mode. Furthermore, they need knowledge whether an effector has malfunctioned. An input to engage the LTUs prior to entering powered-lift flight is required. In addition to this, for the FSD-SVO concept the state of the high-lift needs to be communicated. Whether an under- and overspeed protections needs to be engaged must be communicated along with the exact values of those protections.

The required external data of the two control concepts can be seen in Table 2.7. It is visible that the Fallback concept requires less sensor data sources. Furthermore, with exception to the airspeed, all necessary information the Fallback requires is inertial, i.e. it is of high-integrity and availability. The airspeed is not used for control, but only for logical decisions. In the events that it is not available, other means can be provided as seen in Chapter 4. Therefore, the concept is utilized whenever kinematic or aerodynamic information is lost so as to ensure continued safe flight.

2.4.2 Flight Phases of Lift-To-Cruise VTOL Aircraft

The clear demand of lift-to-cruise aircraft is to enter and leave wingborne flight, while airborne requires procedures that did not exist prior to the emergence of such topologies. These prescribe the changes in mode of operation from hover to wingborne flight and vice-versa. In Section 2.4.1 it was visible that the tracked variables change in dependence of that mode. In [114] we refer to the changes in operation as “transition” and “retransition”. For the sake of completeness, the two are defined here as well. They are relevant to understand when the individual software functions of the control concepts previously introduced in Sections 2.4.1.1 and 2.4.1.2 are engaged.

Transition

The transition is the process of entering wingborne flight from powered-lift flight. The start of the transition phase is initiated by the operator with a predefined set of commands. The transition starts when the aircraft reaches a predefined kinematic speed. The transition phase is completed when the gained airspeed is enough so that the required lift for horizontal flight can be sustained solely with the aerodynamic surfaces. In addition, the transition phase is completed when the hover propulsion system is fully disengaged.

Retransition

The retransition is the process of entering powered-lift flight from wingborne flight. The start of the retransition phase is initiated by the operator with a predefined set of commands. The retransition starts when the propulsion system begins its engagement. The retransition is completed when the hover propulsion system is engaged and the aircraft has decelerated to a predefined kinematic speed.

Given these definitions, the aircraft airspeed envelope is assigned to specific regions. These can be found in Figure 2.10. There, the controller modes are allocated to the appropriate airspeed ranges. For the sake of completeness the range allocation is performed for both configuration with and without a high-lift system. The transition phase initiation and the retransition phase end are denoted with the airspeed V_{HOVER} . This velocity was defined in Section 2.4.1.1. All other definition originate from Section 2.3.

If the hover propulsion system were to be disengaged prior to V_{STALL} , then horizontal flight cannot be guaranteed long-term. Similar to this, the engagement of the Lift/Thrust Units can only be performed below V_{LSNE} in order to guarantee the airframe's structural integrity. The region in Figure 2.10 depicted with red is the overlap of TR and WB and signifies the theoretically permissible hover propulsion activation and disengagement region without robustness or performance considerations. Furthermore, one can notice that in this airspeed range, the choice of Flight Phase - TR or WB is solely dependent on the state of the propulsion system. From these considerations, another observation can be made. In the figure the benefit of utilizing a high-lift system can be observed. The flaps increase the theoretically permissible activation and deactivation region and in fact allow for earlier activation with relation to the airspeed.

The location of the three flight phases is dependent on the specific airframe and its aerodynamic characteristics. The pilot inputs that enable the changes to other regions are derived from the operational procedures and are therefore a result of design decisions. Those are explained in detail in further chapters of this thesis.

2.4.3 Aircraft Control Inceptors, Discrete Inputs and Indications

The previous sections discussed the aspects of the aircraft operation with respect to the utilized control concepts. Equally important to the human-centered automation is the interaction concept as summarized in Section 2.2.1.1. The interaction between operator,

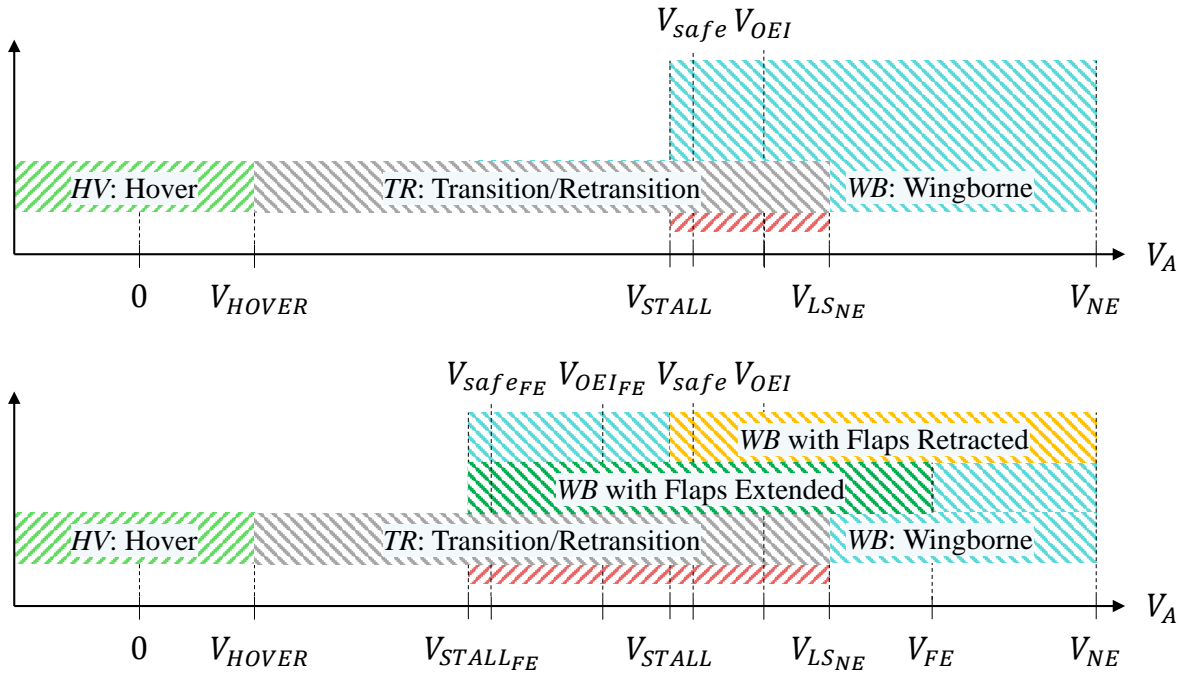


Figure 2.10: *Lift-to-Cruise Aircraft Flight Phase Allocation with Relation to the Speed. The Region Sizes are Chosen for Better Visibility and Need Not be to Scale. **Top:** The Region Allocation for Constellations Without High-Lift Systems. **Bottom:** The Region Allocation for Constellations with High-Lift Systems.*

controller and automation needs to be intuitive, easy to understand and transparent. In this section, the details of the Human-Machine-Interface that impact the automation design are summarized. The section is split into two parts. Firstly, the pilot input items are presented in Section 2.4.3.1. They lay the foundation of the automation information processing and interpretation of the operator demands and intents. Next, the feedback of the automation to the crew is discussed. The indication items are presented in Section 2.4.3.2.

2.4.3.1 Control Inceptors and Input Items

The control inceptors onboard an aircraft are the primary source of pilot input, necessary to control the vehicle. In the case of a fly-by-wire system, they are directly attached to the flight control system and the input the operator is translated into effector commands by the engaged law. The inceptor position sets the law’s control objective. Relevant industry standards that prescribe requirements on the design of aircraft control inceptors include [115–118]. The input items and control inceptors of the eVTOL aircraft of Section 1.1.4, the relevant force gradients and damping forces are not in the scope of this thesis.

The movement ranges of the control inceptors of the vehicle are depicted in Figure 2.11. On the right, the so-called “climb stick” is visualized. The left/right and up/down movement of that stick are responsible for the vertical and directional control of the

aircraft. The exact command variables depend on the currently engaged law presented in Sections 2.4.1.1 and 2.4.1.2. For both laws, the two movement directions - left/right and up/down - are allocated to channels three and four respectively. The channel allocation was presented with Figure 2.9 and Table 2.6 respectively.

For the the transition and retransition automation, the inceptor depicted on the left-hand side of Figure 2.11 is of particular interest. This input item is the so-called “throttle stick”. Detailed information on the design of the throttle stick is available in [119]. The left/right movement is responsible for channel one of the two laws, presented in Sections 2.4.1.1 and 2.4.1.2, whereas the up/down movement is responsible for channel two, i.e. the longitudinal control. The latter also represents the currently required flight phase and is the reason for the non-trivial geometric design of the inceptor ranges as visible in Figure 2.11. The properties of the design are elaborated upon in detail below.

The throttle stick is divided into three parts as visible in the figure. First part, denoted in green is intended for the hover phase. There, the full range of the lateral control can be utilized. The second part, marked in gray, is allocated for the transition phase. The geometrical restrictions of the lateral control become evident in this region. This is the reason that the Fallback control concept does not need authority in this phases as discussed in Section 2.4.1.2. Lastly, the movement range, depicted in blue is allocated for the wingborne phase. This color-coding is consistent with Figure 2.10, where the flight phases were introduced. The division of the stick is important for the automation design. The automation together with the control concepts need to guarantee that these flight phases are reached.

In addition, the throttle stick utilizes tactile cues. They are important for the operation for maintaining situational awareness. The tactile cues provide mechanical feedback of the operator’s intent. Firstly, there is a so-called “detent” between the hover and transition throttle regions. It is depicted with a red dashed line in Figure 2.11. At the location of the detent, a perceivable higher force is exerted. Thereby, the operator is able to remain in the individual regions without significant effort. In addition, the pilot gets immediate feedback when a boundary has been crossed.

Between the wingborne and transition regions, the tactile cue is referred to as “gate”. This is a lateral movement corridor, the middle of which is the border of the two regions. In addition, the two entry and exit points of the gate have mechanical barriers that could prohibit the movement of the inceptor. They can either not permit entering or exiting the gate. The mechanical barriers can be opened via a dedicated discrete input item. In addition, the barrier can be opened by the automation module, which enables the implementation of the procedures, presented later in Chapter 4. In Figure 2.11, an opened barrier can be seen at the upper entry/exit point of the gate. It is marked with a red-white pattern. A closed barrier is visible on the bottom entry/exit point of the gate, denoted with a thick red line. The choice of barrier position in the figure is chosen arbitrarily for visualization purposes of both barrier states.

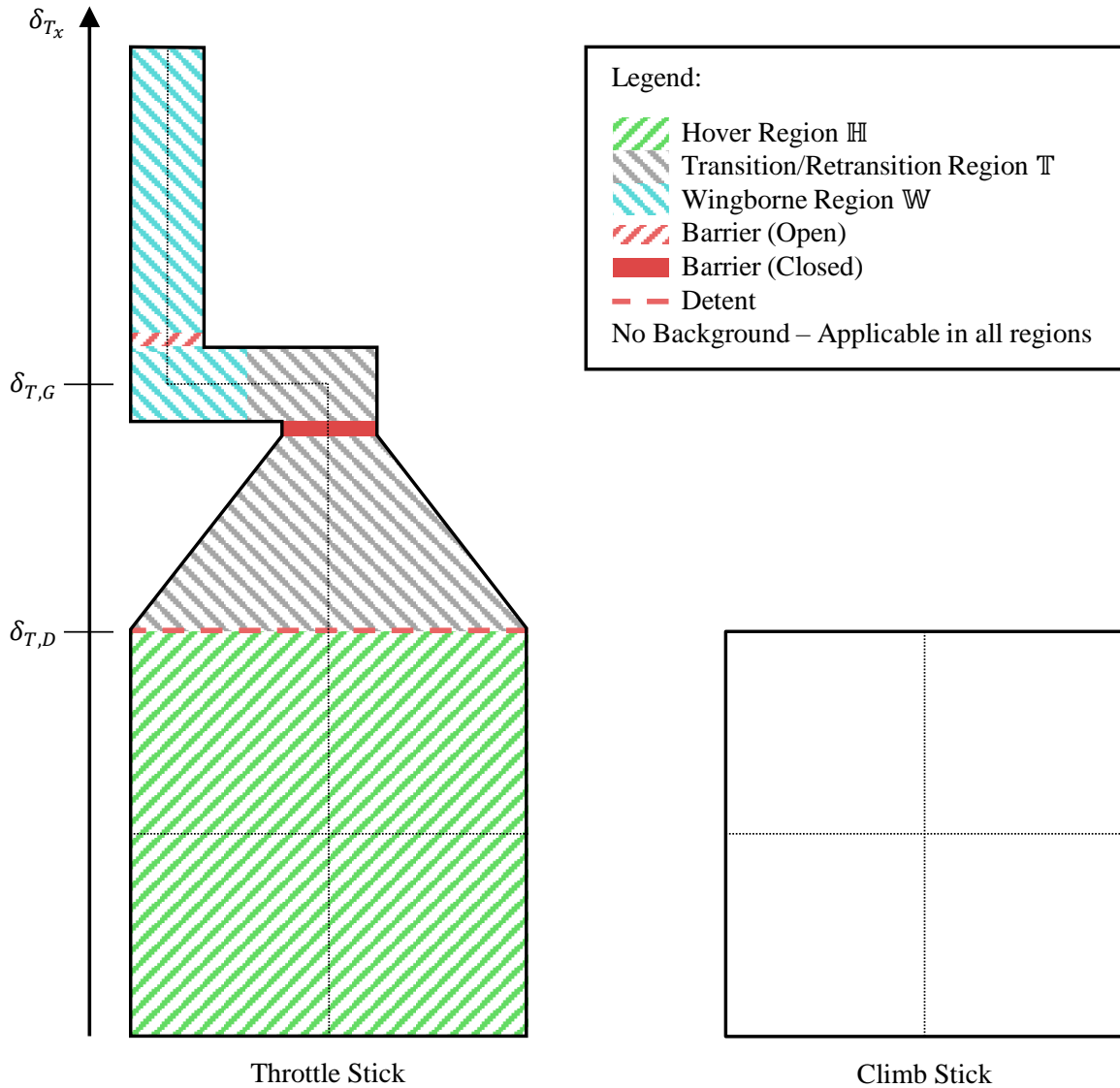


Figure 2.11: *The Pilot Control Inceptors. **Left:** Throttle Stick. The Up/Down Movement is Responsible for Longitudinal Control. The Left/Right Movement in the Hover and Transition regions is Responsible for Lateral Control. **Right:** Climb Stick. The Up/Down Movement is Responsible for Vertical Control. The Left/Right Movement is Responsible for Directional Control.*

In this thesis, the following convention shall be used in order to indicate the position of the inceptor with relation to the regions, depicted in Figure 2.11. The input item, responsible for the opening the barriers shall be denoted with $OPEN_{GATE}$

The vertical position of the inceptor is denoted with δ_T . Whether the throttle is in the hover, transition or wingborne region, is the case when

$$\begin{aligned} \delta_T &\in \mathbb{H}, \\ \delta_T &\in \mathbb{T} \text{ or} \\ \delta_T &\in \mathbb{W} \end{aligned} \tag{2.35}$$

holds respectively. In addition, the position of the throttle in the gate and at the detent are denoted with $\delta_{T,G}$ and $\delta_{T,D}$ respectively. They are visible in Figure 2.11. With the former value and the regions from above, it is possible to distinguish whether the throttle position is on the left or right side in the gate. The two regions - “in gate left” and “in gate right” can be formulated as

$$\begin{aligned} \mathbb{L} &:= \{\delta_T \in \mathbb{W} | \delta_{T_x} = \delta_{T,G}\} \text{ and} \\ \mathbb{R} &:= \{\delta_T \in \mathbb{T} | \delta_{T_x} = \delta_{T,G}\} \end{aligned} \tag{2.36}$$

respectively.

Lastly, provisions for two discrete input items are foreseen. Those are utilized by the low-degree of automation system, discussed in Chapter 4, and are required for the movement of the high-lift system. For this reason, they are denoted with $extend_{rqst} \in \mathbb{B}$ and $retract_{rqst} \in \mathbb{B}$.

2.4.3.2 Indications

The indication items include a warnings and caution system, commonly found onboard aircraft [120, 121]. The warnings and cautions can be dismissed manually if they are not persistent. In addition, the indication item that is responsible for the information supply to the crew with regards to the status of transition and retransition automation is published with [9]. The item itself is seen in Figure 2.12. The human factors considerations with regards to manned flight that deal with the layout of the indication item are not in the scope of this thesis.

With regards to the indication item, the central task of the automation is to provide the necessary information in order to facilitate the appropriate situational awareness. For this reason, the indication item is elaborated upon below.

In the figure, the three divisions signify the three flight phases of Section 2.4.2 - hover, transition/retransition and wingborne respectively. Based on the automation progress and status, the three divisions assume the colors, depicted in the figure. The color coding interpretation is described in detail in [9] but for the sake of completeness is also summarized here:

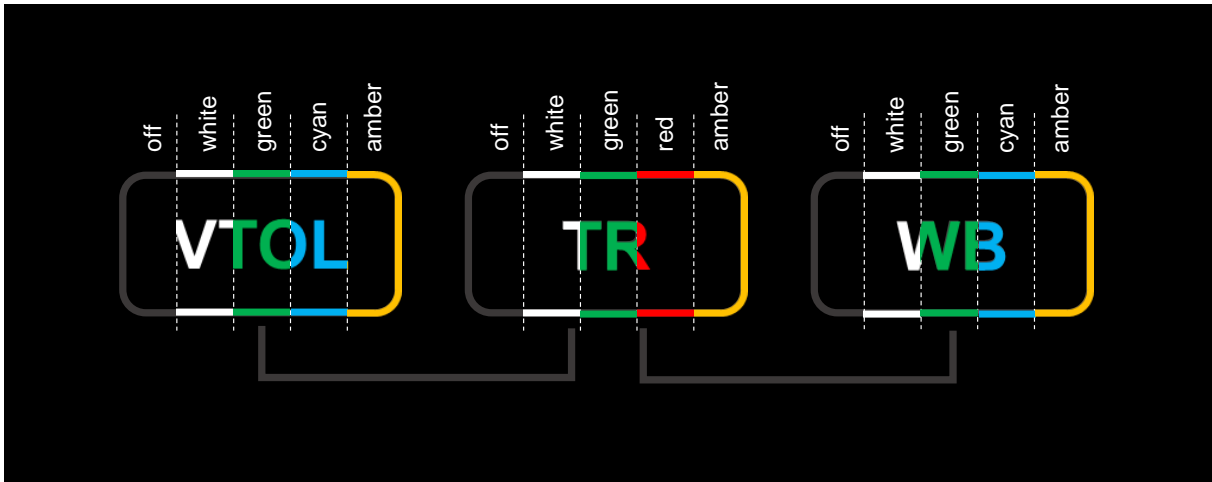


Figure 2.12: *Transition and Retransition Procedure Indication Item. The Image Stems From [9].*

- **Off:** Fully disengaged. The mode of operation is neither requested, nor in the process of disarming.
- **White:** The mode of operation is requested or in the process of arming.
- **Green:** The mode of operation is engaged and no degradation is in effect.
- **Cyan:** The mode of operation is engaged. However, it is in a degraded state.
- **Amber:** A pilot action is necessary in order to engage the mode of operation.
- **Red:** The mode of operation is unavailable.

Please note that the last color coding would be applicable if for example the kinematic velocity information should fail during wingborne flight. In these instances, permitting the entry into the transition phase would not be allowed. The conditions for this unavailability are a direct function of the flight law's robustness and sensor information requirements in the flight phases and therefore application specific.

2.5 Regulatory Efforts and Standards

The rapid development of the manned eVTOL sector has not been left unnoticed by the certification authorities. In an effort to establish a harmonized certification process and common practices in the early stages of the market development, the European Aviation Safety Agency (EASA) has produced the Special Condition for Vertical Take-Off and Landing Aircraft (SC-VTOL) [35] and Means of Compliance with the Special Condition VTOL (MOC SC-VTOL) [10, 36, 37]. SAE International has also addressed the topic of lift-to-cruise configurations in the context of flight control function development of

manned military aircraft. Their efforts have been published in the revised version of Vehicle Management Systems - Flight Control Function, Design, Installation and Test of Piloted Military Aircraft, General Specification (AS94900A) [122].

This section summarizes relevant extracts from regulatory requirements that directly impact the automation design. The section is organized as follows. Firstly, the direct requirements on the automation of the transition and retransition is summarized. This is performed in Section 2.5.1. In addition, requirements on the transition and retransition are imposed indirectly. These depend on the exact execution of those processes with relation to the flight phase. In order to fit the procedures and the underlying automation into these flight phases, they must be understood fully. Therefore, the next sections summarize the requirements that arise during the vertical take-off and landing, the departure and the approach. They are discussed in Sections 2.5.2, 2.5.4 and 2.5.3.

2.5.1 Requirements on the Transition and Retransition

For lift-to-cruise configurations, AS94900A distinguishes between two flight envelopes, referred to as “hover” and “forward flight”. They do not explicitly define the regions, but the document implicates that the difference of the envelopes is the forward speed and the utilization of the hover propulsion system.

For the transition process, AS94900A states that altitude shall be maintained and that the aircraft shall not lose altitude. Furthermore, requirement for heading tracking are set. For steady-state level flight with airspeed below fifty knots, the deviation shall be less than one degree and less than 0.5 degrees for higher speeds. In transients, this requirement is relaxed to five degrees. As for the retransition, AS94900A states that no negative forward kinematic speed is permissible.

The MOC SC-VTOL sets requirements on eVTOL aircraft, regardless of their topology. By definition eVTOL aircraft for UAM may not be configurations, capable of both powered-lift and wingborne flight and therefore the MOC SC-VTOL does not impose any requirements on the transition and retransition or any other derived procedures. It does, however, lay down clear envelopes and conditions that need to be fulfilled in different flight segments within a mission. By implication, one can derive in which segment the transition and retransition can take place and what properties the system needs to fulfill during those processes. Those topics from the MOC SC-VTOL are briefly summarized. They are used to fit the solutions in the standard in Chapter 4.

One central term in the MOC SC-VTOL is the Critical Failure of Performance (CFP). In [10], the CFP is defined as the set of probabilistically permissible failures within the system, with which a performance parameter is degraded most. Since the performance parameters change from flight phase to flight phase as is seen later in this section, by implication the CFP is flight phase specific. Additionally, the MOC SC-VTOL requires explicit definitions of procedures in the cases where there are different modes of operation or degradation.

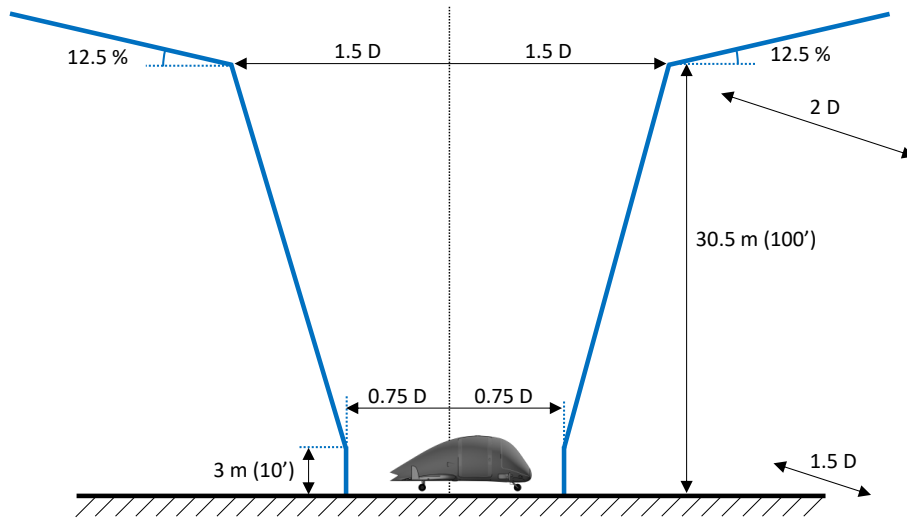


Figure 2.13: *The Take-Off and Landing Reference Volume as Found in [10]*

2.5.2 Vertical Take-off and Landing

The MOC SC-VTOL prescribes the so-called “reference volume” [10]. During the vertical take-off and landing, the aircraft should fully remain within the confines of this volume. For the sake of completeness, the reference volume is depicted in Figure 2.13. In particular, the “high hover height h_2 ” [10] is of importance. This height is the basis for the specifications of the departure and approach, which need to be created with relation to h_2 - the so-called “virtual elevated vertiport” [10].

2.5.3 Departure

In order to obtain certification credit, the applicant must demonstrate that the aircraft can perform the take-off trajectories defined in MOC VTOL.2115. Three different scenarios are defined which result from three different use-cases:

- Conventional Take-Off: Assumes a take-off from a vertiport on the ground with no obstructions.
- Elevated Conventional Take-Off: Similar to the Conventional Take-Off in that it also assumes that there are no obstructions. However, in this trajectory the use-case is a take-off from an elevated surface.
- Vertical Take-Off: A use-case that is supposed to satisfy the future UAM demand. In this scenario, the assumed take-off is in an environment, where obstructions are possible.

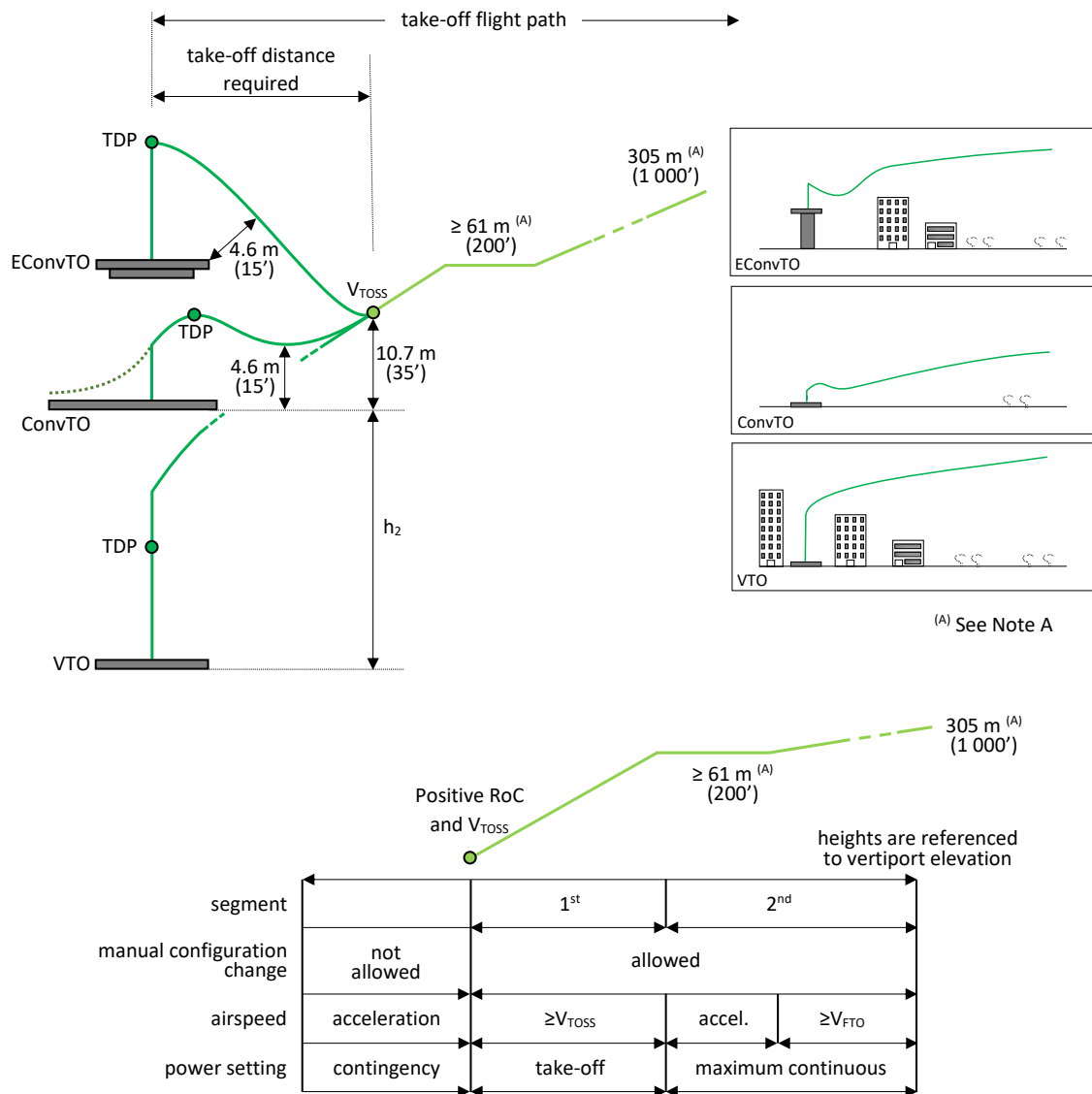


Figure 2.14: The Take-off Trajectories as Found in [10]

The take-off trajectories are visualized in Figure 2.14. The first prominent point in the take-off trajectory is the Take-Off Decision Point (TDP). The decision where to place the TDP is up to the applicant. A rejected take-off can be performed before reaching the TDP. Afterwards, it shall be possible to execute the take-off trajectory even in the presence of a CFP. Naturally, the decision of where to place the TDP depends on the vehicle performance characteristics.

Assuming no rejected take-off is executed, upon reaching the TDP the operator is required to reach a velocity, referred to as Take-Off Safety Speed (TOSS). This speed is denoted in this thesis with V_{TOSS} . During the acceleration to V_{TOSS} , the proximity from ground obstacles (including the elevated vertiport height) shall never be less than 15 feet in any vertiport placement and the height above the elevated vertiport shall not exceed 35 feet for a Conventional Take-Off. Furthermore, the SC-VTOL prohibits “manual configuration changes” in this phase of the take-off. No manual configuration changes also implies that even in the event of a CFP, the operator shall be capable of controlling the aircraft solely using the primary control inceptors. This information is visible also in Figure 2.14.

The subsequent phases of the take-off element are involved with a climb to a total of one thousand feet. The climb is split into two distinct segments. In the first one, the aircraft is required to climb to two hundred feet with a speed of no less than V_{TOSS} . During this segment, the rate of climb cannot be less than 4.5%. As visible from Figure 2.13, this climb gradient can be up to 12.5%.

In the second segment, another acceleration phase takes place, in which the aircraft shall reach a velocity, referred to as the Final Take-Off Speed (FTO). In this thesis this speed is expressed with V_{FTO} . Upon reaching this speed the aircraft shall be capable of climbing with a gradient of no less than 2.5%. The height of one thousand feet shall be reached at an aircraft configuration, referred to as “final take-off configuration” [10]. Automatic reconfiguration changes are permitted.

2.5.4 Approach

MOC VTOL.2130 of the MOC SC-VTOL specifies the performance that has to be demonstrated during the landing phase. The document does not prescribe a rate of descent but rather allocates this decision to the applicant. The Landing Decision Point (LDP) [10] is introduced and is defined as the last point along the approach trajectory, at which the landing can be rejected and a go-around can be initiated. This point has to be reached with a speed of at most V_{REF} - the so-called “Landing Reference Speed” [10]. Furthermore, if the speed of the aircraft happens to be less than V_{REF} during the go-around, no reconfiguration should be necessary until regaining the speed. This velocity is furthermore linked to performance requirements in terms of flight path angle

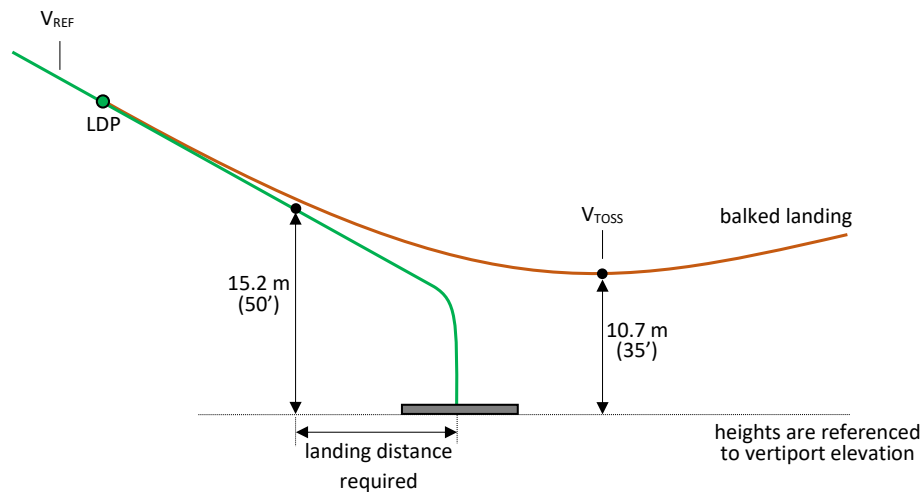


Figure 2.15: *The Approach Trajectory Including a Rejected Landing as Found in [10].*

and maneuverability. During the rejected landing, the velocity V_{TOSS} is to be regained, upon which the requirements of the take-off apply. Both scenarios as depicted in Figure 2.15.

This concludes the second chapter of this thesis, where the theoretical and mathematical preliminaries were presented. They are necessary in order to understand the solutions, presented in the subsequent chapters of this thesis. In Section 2.2, the theory behind the correct design and implementation of automation functions was presented. Afterwards, the characteristics of the system, subject to automation, were studied. These included physical effects and limitations that are a consequence of the airframe, found in Section 2.3. Subsequently, design decisions from the interaction and control concept were elaborated upon, because they impact the automation mechanics. Those were summarized in 2.4. Lastly, in Section 2.5 the performance requirements from the regulatory side were analyzed. In the next chapter, a high-degree of transition and retransition automation is presented. The solutions of the next chapter realize **Contribution 1**.

Chapter 3

High-Degree of Automation Transition and Retransition

In [123] the authors make the claim that a majority of the accidents in aviation are linked at least in part to human mistakes. The data in [123] points out that more than seventy percent of incidents have had some sort of crew involvement - be it direct or indirect. Therefore, the notion of a system that is capable of executing high-level crew commands on its own appears very attractive in order to minimize mishaps.

The reason a properly designed automated system could reduce the probability of human-made errors is that such a system intrinsically reduces the human involvement. Not requiring operator input for every single decision-making process leaves less possibility for erroneous input. These decisions are instead delegated and automated by the design. The high-degree of automation inherently requires more software capabilities. They in turn can be used to enforce and maintain an envelope, in which the aircraft exhibits desirable properties, adequate handling qualities, maintains structural limits, etc. Using automation, the target properties can thus be fulfilled regardless of the operator input in order to keep the system safe. If necessary, they can even be enforced for conflicting operator commands.

For this reason, in this chapter a high-degree automation strategy for reconfiguration from powered-lift to wingborne flight and back for lift-to-cruise VTOL aircraft is proposed. It addresses all challenges that inherently arise with regards to high automation concepts. In terms of operation, the solution ensures that the concept is intuitive for the crew. Having always the situational awareness in hindsight, the design is human-centered and also includes the information supply to the flight deck indications. Furthermore, faults and failures in the aircraft components are addressed and considered. The resulting procedures and automation are associated with **Contribution 1** of this thesis.

The chapter is structured as follows. Firstly, Section 3.1 analyses the different challenges that the automation concept needs to consider and directly address. It discusses the safety-driven properties that need to be enforced, the different operational aspects that the design needs to comply with and the type of information that the operator needs

to be supplied with. The section sets the requirements that the derived procedures and automation need to fulfill and thereby lays the foundation for the whole subsequently presented work of this chapter.

Afterwards, in Section 3.2 all functions necessary in order to perform a transition and retransition are qualitatively summarized. This is necessary to derive the fundamental software architecture of the high-degree of automation flight control algorithms. For the sake of completeness, those functions are then allocated to the software components. The properties that those functions need to exhibit to enable the suggested procedure are specified. A basis for the systems the automation interacts with is established. All subsequent ideas and concepts of this chapter are derived with the surrounding systems in hindsight.

With the information of the above-mentioned section, a process flow during a transition to wingborne flight and a retransition to powered-lift flight is formed. Contingency processes in abnormal events are derived. In both normal and contingency procedures the necessary operator actions are tracked, demonstrating minimum crew involvement in the reconfiguration. This is performed in Section 3.3. The section proceeds to analyze and prove on a high-level that the software mechanisms maintain the set out safety properties as designed.

The core of the chapter is contained in Section 3.4, where the design of the automation functions that facilitate the procedures of Section 3.3 is presented. Section 3.4 introduces the required State Machine and how the different logic of the components is intertwined in order to completely automate the transition and retransition. At the same time, it enforces a safe system state and enables the contingency measures. The section discusses what operator inputs are required for the operation and how they are processed. This is done for all sensor sources. The State Machine mechanics are presented and how their output affects the surrounding elements is discussed. This includes the informational supply to the operator, to the aircraft effectors and to the surrounding software components that are part of the control algorithm. In the section the utilization of a high-lift system within the transition and retransition automation is shown.

In Section 3.5 the characteristics of the methods presented in Section 3.4 are examined in depth and compliance of the design from Section 3.4 to the procedures of Section 3.3 and the safety objectives of Section 3.1 is demonstrated. The section is supplemented with an analysis of the automated system response for the different failure modes. The chapter is concluded with Section 3.6 where the achieved contributions are summarized.

3.1 Problem Description

In this section the aspects that impact the developed concept are broken down and clear objectives that the proposal has to fulfill are defined. The FSD-SVO concept presented in Section 2.4.1.1 offers an intuitive concept to control a VTOL aircraft. It, however,

requires information from an automation module. With the goal to make use of the FSD-SVO for the the automation of the transition and retransition, the requirements that arise with relation to its integration are initially observed. This is performed in Section 3.1.1. Additional concepts that are not considered in the FSD-SVO are addressed by the automation in this chapter.

Sections 2.3 and 2.4 discussed the common malfunctions that are known to occur during a mission with a VTOL. In order to maintain a safe system state, in Section 3.1.2 the potential hazards during operation that have a direct impact on the later derived procedures are discussed. The section derives safety-driven requirements that are allocated to the automation concept.

Finally, in order to design an adequate automation strategy, not only a robust design that accounts for the possible faults is needed. In addition, the operator involvement needs to be considered. No process that is human-centered may be deemed satisfactory if the user has no perception of the state of the function. This hinders the knowledge of available input options and mitigation strategies. Therefore, the pilot involvement in the context of the nominal process flow and in the cases of abnormal events needs to be accounted for. In Section 3.1.3 defines the goals with regard to the operator awareness during the reconfiguration from powered-lift to wingborne flight and back.

3.1.1 Coherence and Supplementation of the Simplified Vehicle Operations Concept

The Simplified Vehicle Operations provide a concept that aims to minimize the necessary operator skill while maintaining the required operational safety. This for example is the justification for the FSD-SVO concept discussed in Section 2.4.1.1.

As seen in Section 2.4.1, the FSD-SVO addresses the operator control inceptor interpretation in the different flight phases, referred to there as “behavioral modes”. The unified structure’s control variable generation from the inceptors and the inceptors’ interpretations are blended throughout the flight envelope. However, FSD-SVO does not mention how the changes of the flight phases should occur. These changes are the backbone of the command variable blending.

For example, FSD-SVO mentions that during powered-lift system deactivation the awareness of the operator should be maintained. It, however, does not mention under which conditions this is to occur. It does not discuss the information origin or what it needs to contain. The same applies for the retransition. Instead, it mentions that this is to be managed by the automation or “moding” module. This automation module is in the scope of this chapter.

From here the first objective of this chapter is specified. One of the goals is to derive a transition and retransition procedure and an underlying automation module that executes them. This automation concept must be seamlessly integrated in the FSD-SVO concept.

It may not alter or negatively influence the control concepts of FSD-SVO, but should complement the method and manage the sequences and switches between the required behavioral modes.

In addition, FSD-SVO does not address the operation of a high-lift system. In this chapter it is demonstrated how such a system is managed in the context of Simplified Vehicle Operations. The operation must be fully automatic and thereby consistent with the notion of maximum operator workload reduction.

3.1.2 Increasing the Robustness

Although rare, component faults occur during flight. An intuitive operational concept is brittle if it does not account for failures and subsequently provide an adequate and equally as intuitive mitigation strategy.

FSD-SVO addresses the nominal operation. Though it manages to compensate common failure scenarios and omit immediate hazards, the subsequent impact in the switching between flight phases needs addressing. This is a further issue solved in this chapter. The proposed strategy must constantly enforce safety-constraints, driven by the reconfiguration state and failure scenarios. The constraints do not influence the nominal behavior, but increase the operational safety. How this is done is seen in Table 3.1.

Table 3.1 shows the severity of different failures for the functions within the high-degree of automation control laws, responsible for the powered-lift and high-lift system scheduling in the different flight phases. It must be noted that both the high-degree of automation functions and possible hazards are much more than the ones listed in Table 3.1. The table is not exhaustive and does not claim completeness. It serves an illustrative purpose to justify the methods presented in this chapter.

The first three rows of the table discuss the severity of the failure of LTUs. If a failure occurs at low aerodynamic pressure, the effect would be catastrophic, as stability cannot be maintained. At the same time at high dynamic pressures - i.e. in wingborne flight - the hover propulsion system is not required and therefore such an event does not have immediate adverse safety effect. This by definition is a dormant error that manifests in a catastrophic event if this hazard remains unaddressed in the retransition and if the aircraft decelerates to low airspeed. More precisely, the mitigation of limiting the deceleration of the aircraft such that stall cannot occur, reduces the severity significantly. The subsequent operator workload is increased due to the necessity to perform a wingborne landing. This is something that must be facilitated by the automation.

The next two rows of Table 3.1 tackle the issue of an inadvertent activation of a propulsion unit. It can easily be determined that in such cases it is necessary to decelerate the aircraft below or not permit the aircraft to accelerate above the structural velocity limit V_{LSNE} . V_{LSNE} was introduced in Section 2.3.1.2. The same argument is made in the last row of the table, only with the structural limit of an extended high-lift system. There again it is necessary to reduce and forbid exceeding given dynamic pressures.

Table 3.1: Sections of the High-Degree of Automation Control Concept FHA that Implicates Requirements on the Automation

Function	Flight Phase	Failure Condition	Failure Effect	Severity	Proposed Mitigation	Resulting Severity
Provide Powered-Lift	Hover	Failure of a critical number of Propulsion Units* below the stall speed	Crash at low speeds and altitudes	CAT	At sufficient altitudes the termination system can be engaged. Otherwise none.	CAT
	Wingborne	Failure of a critical number of Propulsion Units*	None	No Effect		No Effect
	Retransition	Failure of a critical number of Propulsion Units* above the stall speed	Further deceleration will lead to a crash at low altitudes	CAT	Prohibit deceleration of the aircraft to speeds below V_{STALL} .	MAJ
Provide High-Lift	Wingborne	Inadvertent powered lift production above an airspeed of V_{LSNE}	Structural Damage	CAT	Cut power supply to powered-lift units after executing the transition	MAJ
	Transition	Erroneous non-zero powered lift while accelerating above V_{LSNE}	Structural Damage	CAT	Prohibit acceleration of the aircraft to speeds higher than V_{LSNE} when powered lift not fully disengaged	MAJ
	Wingborne	Failure to fully retract while accelerating beyond an airspeed of V_{FE}	Structural Damage	CAT	Prohibit acceleration of the aircraft to speeds higher than V_{FE}	MIN

*Critical number of propulsion units - Number at and above which control authority solely with the powered-lift cannot be ensured.

The automation design addresses the hazards described above and implements the mitigation strategies of Table 3.1, increasing the overall robustness of the control concept.

3.1.3 Operator Support

Even if designed with robustness in mind, an automation strategy is weak if the operator has no knowledge of the incidents that occur or what the subsequent activities of the automation are. If the awareness of the pilot is not adequate, it may be unclear what the actions at disposal are. Consequently, an incorrect action may be taken. This is the next challenge to address.

The proposed strategy is human-centered and intuitive for the operator. It is by design simplistic which aids for adequate mode awareness. The indications are tailored to the underlying automation design. Transparency is ensured by the data, supplied from the automation to the indication items. Mode confusion is omitted by the design of the automation that takes into account the control inceptors and has a limited state-space.

In adverse conditions, the automation provides information to aid the operator if actions are necessary. With the robust-design properties of Section 3.1.2 these actions are not time-critical and further reduce the workload.

With the goals summarized, a solution that satisfies them is presented. The next section presents the functional elements of high-degree of automation system that can solve the above-mentioned challenges.

3.2 Process Breakdown

When performing the transition, the aircraft needs to accelerate to a velocity, commanded by the pilot, which is in the wingborne region. The regions were defined in Section 2.4.2. After a specific airspeed, the hover vertical propulsion units need to shut down. From Section 2.4.2 it is known that this needs to occur at speeds above the stall speed.

A similar rationale is made with relation to the retransition, where when reaching an airspeed while decelerating, the LTUs are to be engaged. This needs to occur at speeds, that are smaller than V_{LSNE} to avoid structural damage.

Therefore, when observing the two processes without the presence of faults, the expectation is that the complexity may be manageable. However, from Table 3.1 it is visible that the driving factor with relation to the software complexity is the necessity to enforce a safe system state at all times and guarantee the operator awareness in the cases where abnormal events occur and action is necessary.

This section summarizes the high-level properties the system needs to satisfy in order to facilitate this high degree of automation while addressing possible malfunctions. The underlying objective for these functions is to avoid system brittleness and at the same time ensure an increased transparency.

The necessary functionality of the high-degree of automation control is listed. Afterwards in Section 3.2.1 the captured functions are allocated to software components. Thereby a software architecture that satisfies the set objectives is derived. This is the basis for Section 3.3 where the mechanics of the transition and retransition processes and the scheduling of the derived functions are explained.

Firstly, to ensure a high level of automation, the following system property is derived.

R1 The system shall provide stability throughout the whole aircraft envelope and track high-order control variables.

Those variables change based on the current position in the flight envelope and the current aircraft configuration state. The above mentioned property is self-explanatory but necessary to satisfy the demand of lower operator qualification as dictated. It is addressed in the FSD-SVO concept that was introduced in Section 2.4.1.1. These properties need to be valid in the event of single failures.

In terms of safety, the following conditions need to be satisfied. Firstly, it needs to be ensured that the speed V_{LSNE} cannot be exceeded in the case where LTUs are not yet disengaged or fail to come to a halt in order to avoid structural damage. The failure modes, because of which disengagement is impossible were discussed in Section 2.3.1. Additionally, a lower velocity needs to be specified, below which disengagement of the motors cannot occur. From these two conditions the following properties are defined.

R2 The system shall implement error detection functions.

R3 The system shall implement envelope protections.

R3.1 The system shall implement overspeed protections.

R3.2 The system shall implement underspeed protections.

R4 The values of the overspeed and underspeed protections shall be dependent on the current aircraft configuration state.

In order to further minimize system opacity and ensure mode awareness, the introduction of HMI functions is necessary.

R5 The operator shall be able to unambiguously specify their desired mode of operation.

R6 The operator shall be made aware of the current state in the process via indications.

Lastly, the proper execution of the transition and retransition needs to be facilitated. In the case of abnormal scenarios, mitigation strategies have to be enforced. The following function is derived.

R7 The system shall implement an automation module that schedules the actions of the different software modules within the FCS.

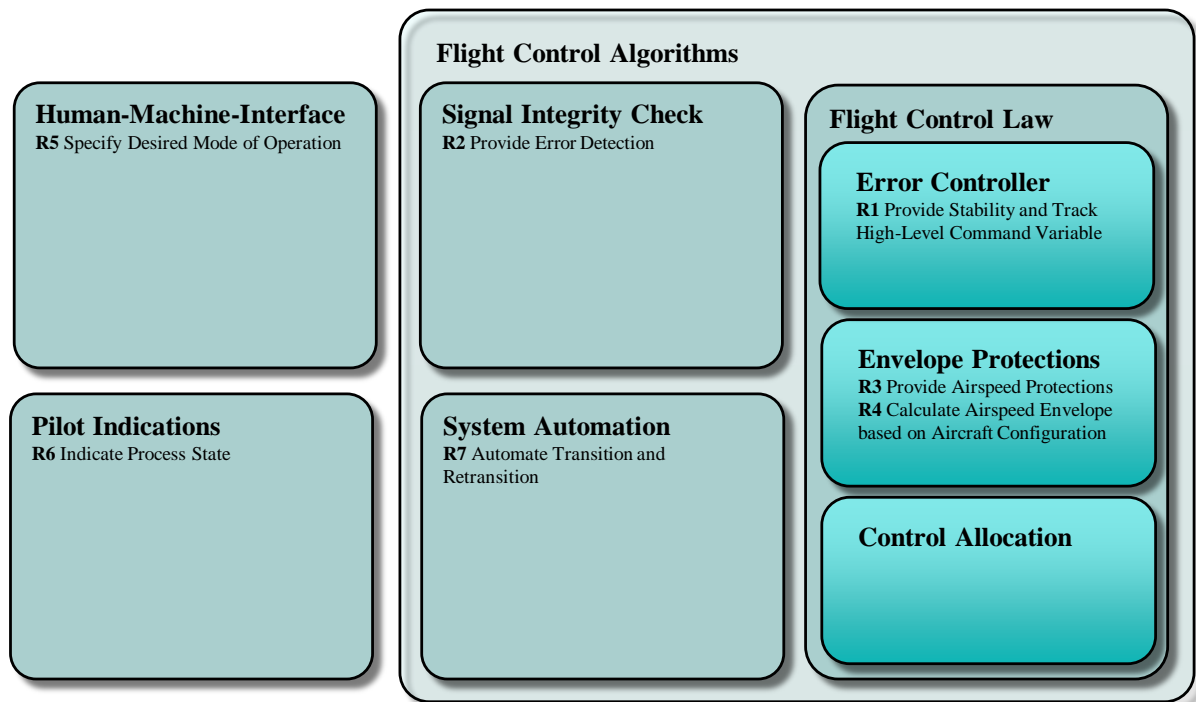


Figure 3.1: *High-Degree of Automation Software Modules, involved in the Transition and Retransition. The list is not exhaustive with regards to actual implementation.*

3.2.1 Software Architecture and Function Assignment

In this section the software architecture of a FCS that facilitates the desired transition and retransition characteristics is introduced. Then, the properties of Section 3.2 to the software components are allocated. The functional allocation to modules of the system architecture is depicted in Figure 3.1.

The Human-Machine-Interface is responsible for processing the operator inputs in terms of control inceptor deflections and discrete inputs, such as buttons and switches. Therefore **R5** is allocated to the HMI. The design of the interface items has to be such that the operator can unambiguously set the required mode of operation - wingborne or hover. This function of the software architecture is further responsible to transmitting all relevant data to the Flight Control Algorithms.

The Flight Control Algorithms block from Figure 3.1 is the container, used to summarize all functionality that is executed on a flight control computer. The individual functions are elaborated upon in the order of causality.

The Signal Integrity Check executes continuous monitoring that tracks the health of all incoming signals necessary for closed-loop control and automation. The Signal Integrity Check conditions the signals using different filtering methods and applies voting mechanisms for the cases of redundant signal sources. It proceeds to forward all relevant peripheral malfunctions that are registered. It therefore has to satisfy **R2**. As seen later, the automation design is robust against an undetected erroneous input of certain

information sources but not all. A summary of the signals, where resilience cannot be ensured is summarized in Section 3.5.2.2. Therefore, for such instances this failure mode needs to be mitigated by utilizing sensor redundancy.

The System Automation processes the current aircraft configuration. Depending on configuration, the sensor and operator input along with the possible malfunctions, it makes deterministic decisions on the appropriate actions and pieces together all other software functions. Hence, it is capable of orchestrating the whole system to produce desirable transition and retransition sequences, fulfilling **R7**.

The Flight Control Law function is responsible for calculating effector commands such that higher-order objectives are achieved. The command variables change over the envelope and the choice of the variables is managed by the System Automation. The Flight Control Law is divided into individual tasks that must be touched upon here. It is the implementation of the FSD-SVO concept that was already summarized in Section 2.4.1.1.

The System Automation forwards to the Error Controller what the current flight state is. The latter then maps the operator input from the Human-Machine-Interface to the command variables that fit that envelope and guarantees that the handling qualities are maintained, satisfying **R1**.

Similarly, the Envelope Protections receive the state of the aircraft configuration from the System Automation. This aircraft configuration determines the values of the Airspeed Protections, which are enforced by the function. Therefore, **R3** and **R4** are maintained by this software module.

Although no property is assigned to the control allocation as seen in Figure 3.1, this module plays an important part in the overall strategy. The control allocation takes the demands of the Error Controller in terms of required forces and moments and distributes them to the effectors. However, it also accounts for commands that stem from the System Automation, thereby executing processes such as the turn-on or the shutdown of the hover propulsion units. It translates the requirements of the System Automation that implement **R7** to hover propulsion commands.

The Pilot Indications receive the information from the System Automation with regard to the state of the aircraft configuration and status of the procedures. By the use of appropriate display items, they fulfill **R6**. The Pilot Indications complete the interaction concept, supplying the operator with the necessary information to make qualified judgments and gain mode and situational awareness.

3.3 Transition and Retransition Functional Flow

In the previous sections, the necessary properties to facilitate the transition and retransition were identified. They were assigned to software systems within the architecture. In this section, the sequences of pilot and system actions that represent the automated transition and retransition are presented.

The descriptions and explanations below provide a high-level overview. They are used to derive the actual implementation and the exact mechanics of each step in the sequence. The sub-functions are elaborated upon in Section 3.4.

It must be noted that the execution flows depicted in this section are used for illustrative purposes in order to elaborate upon the sequence of events that occur during the automated transition and retransition. As seen later in Section 3.4, the deployed automation module is designed to account for malfunctions of components and deviations by the crew from the procedures below.

3.3.1 Normal Transition

The execution flow of the procedure is depicted in Figure 3.2. The figure considers the actions of the software or crew. The actions of the latter are depicted in gray. In the following sections, the identifiers of the procedural steps conform to the following convention. Each step begins with either a “t” or “r”, indicating whether this step belongs to the transition or retransition procedures respectively. If present, the next character “m” signifies that this step in the procedure is part of the mitigation strategy. This is followed by the numbering in order of the causal chain of events. If the step is part of the expected flow, then the identifier ends. Alternatively, if this step is off-nominal - such as the start of the mitigation strategy or the mitigation strategy options themselves, then the identifier is supplemented with additional characters (e.g. “a” or “b”) to indicate this.

As mentioned in Section 2.4.2, the transition starts with the operator request. This is done in step t1). Over the Human-Machine-Interface, the operator’s input changes, communicating to the software that wingborne flight state is desired. This is done by moving the throttle lever in the division dedicated for wingborne flight and remaining at its lower-most section. The divisions of the control inceptor were depicted previously in Section 2.4.3.1. The processing method is explained in more detail later in Section 3.4.1.1.

From the definition of Section 2.4.2, it further follows that the starting point of the procedure is in a flight region, where the propulsion system is utilized for control. Therefore, for now it is assumed that the upper airspeed limit is prior to the start of the process is V_{LSNE} . Later in Section 3.3.2 it is demonstrated that this assumption holds.

As argued in Section 3.2.1, this input initiates a forward acceleration of the aircraft, which is driven by the law. Once the transition speed has been reached, the conditions from Section 2.4.2 for the transition initiation apply. Therefore, in t2) the Pilot indications

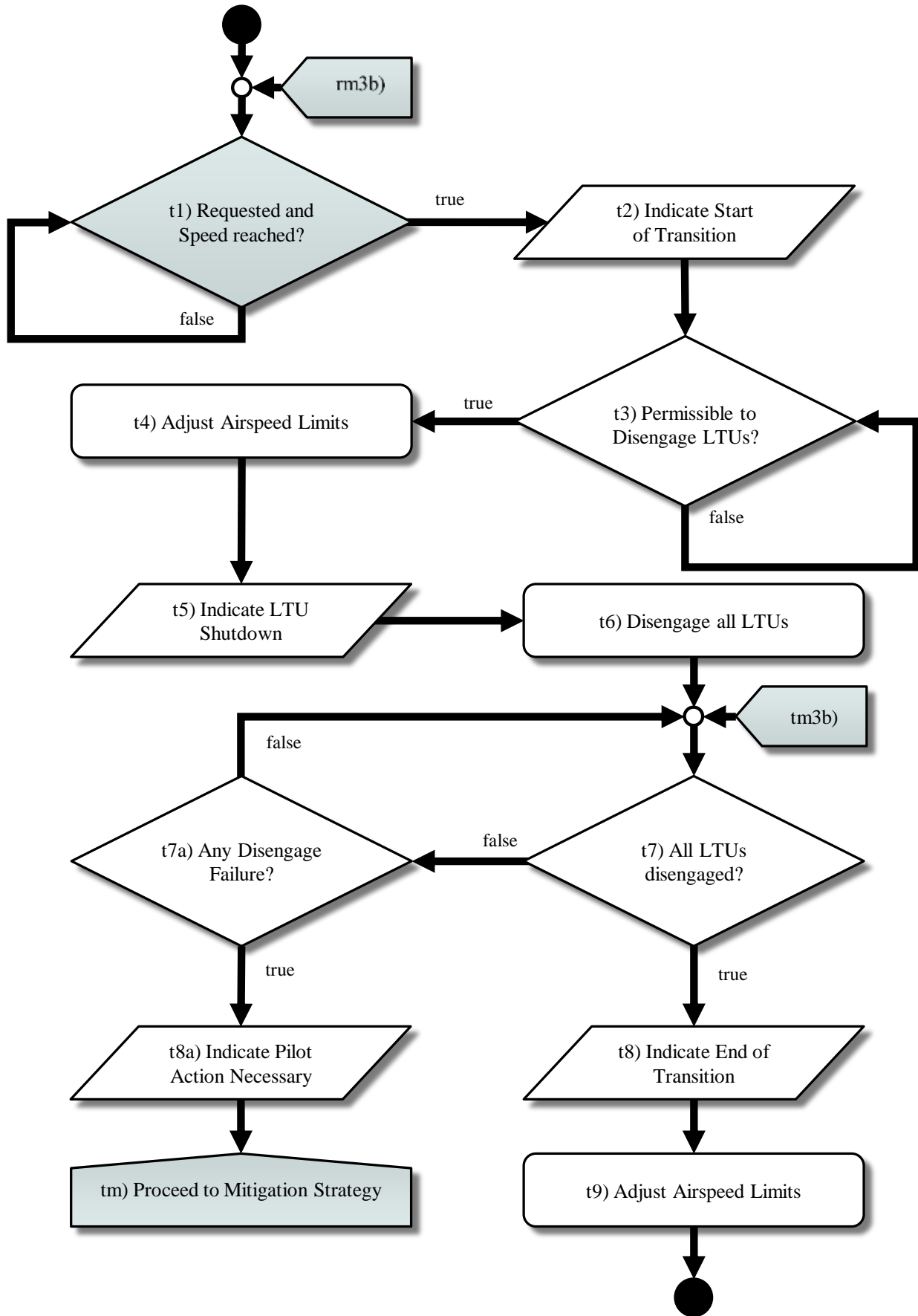


Figure 3.2: Transition Process Flowchart

change accordingly to notify the operator of the start of the transition. This is necessary to facilitate better awareness of the upcoming change of flight state and possible necessity of abnormal procedure execution.

The check from t3) continuously monitors whether the conditions to initiate a motor disengagement are met as the aircraft accelerates forward. With the increase of airspeed, the thrust from the powered-lift units that is necessary to sustain the flight is gradually taken over from the aerodynamic surfaces. When check t3) is evaluated to pass, the motor disengagement process is triggered. This evaluation is based on several criteria and the check itself is explained in Section 3.4.1.2.

In t4) the airspeed limits are readjusted due to the upcoming shutdown. In this step, the lower limit is chosen such that stall cannot occur. The choice of speed is explained in Section 3.4.4.2. Afterwards in t5) the operator is informed with means of indication that the shutdown is starting.

To recapitulate - the envelope protections has an airspeed limit of V_{LSNE} from the start of the procedure, mitigating structural damage. The lower airspeed limit set in step t4) and mitigates stall. By procedure design a safe system state during the motor disengagement is enforced.

The System Automation proceeds to initiate the shutdown sequence in t6). Over the control allocation, it enforces a ramp down of the motors. In [114] we noticed that due to the propeller inflow conditions and interactions the efficiency of the LTUs is different. Since the thrust is quadratic with relation to the revolution rate, having the gradient the same during shutdown produces a moment, equivalent to a disturbance. Therefore, the gradient of the ramp of all motors need not be the same. The design of the command downward ramp is not within the scope of this thesis.

The System Automation proceeds in t7) to continuously check whether the powered-lift system has indeed been disengaged. This is done by means of feedback from the propulsion system and explained in Section 3.4.1.1.

If the check from t7) passes, then the operator is notified of the conclusion of the transition procedure and that wingborne flight is entered in t8). Since the aircraft is fully wingborne, in step t9) the upper airspeed limit is readjusted to enable the entry to higher airspeed, which is explained later in Section 3.4.4.2. This marks the end of the transition procedure.

In step t9) the reason for the checks in t7) becomes apparent. t7) ensures that the conditions for wingborne flight are fully met. If step t7) were to be omitted, then a failure of the disengagement and the subsequent release of the airspeed in t9) results in the possible exceedance of the structural limit speed with operative propulsion system V_{LSNE} . By the means of t7) the severity of the findings of Table 3.1 are mitigated.

The check for off-nominal events t7a) runs together with t7) and evaluates whether a failure condition is in effect. This is registered by the LTU feedback, Signal Integrity checking but also functionally by means of a timeout. This follows the computations,

introduced later in Section 3.4.1.1. Provided a failure is in effect, then this abnormality requires an appropriate mitigation. The operator is informed of their necessary involvement in step t8a) over the Indications. This is handled in tm), which is explained in Section 3.3.3. Depending on the action performed, the transition may be reentered via tm3b), as explained in that section.

The complete automation of the transition in the nominal case is evident in Figure 3.2 - it is visible that from the moment of initiation onward, no action from the operator is required. The actions of t2), t5) and t8a) are necessary to prepare and ensure the awareness of the operator in the cases where tm) - the mitigation - is in effect. If the mentioned failure scenario of Table 3.1 is not applicable for a given configuration, then these actions may be omitted. This failure scenario can be mitigated by Lift/Thrust Unit design.

It must be noted that the transition procedure may be aborted by the operator prior to the *true* evaluation of check t3) by withdrawing the transition request that is one of the conditions of step t1). Since the withdrawal is equivalent to a retransition request, after step t3) this action would trigger a retransition process. For the sake of readability, in Figure 3.2 it is assumed that the crew does not deviate from the procedure. The appropriate means to account for such deviations are considered and the decision-making process of the software in these instances become visible with the introduction of the system automation in Section 3.4 and the logic analysis in Section 3.5.2 later on in this chapter.

3.3.2 Normal Retransition

The execution flow of the procedure is depicted in Figure 3.3. The figure summarizes the actions of the software and crew during the retransition. The figure follows the same color and naming patterns as Figure 3.2.

According to the definition of the retransition of Section 2.4.2, the state prior to triggering the process is the wingborne limit. From Section 3.3.1 it follows that the lower airspeed limit at this point is set such that at the very least stall cannot occur. The exact values are explained later in Section 3.4.4.2.

Similar to the normal transition process from Section 3.3.1, in the check from r1) it is evaluated whether the retransition is requested. This is handled by the system automation. The request inherently creates a deceleration command, which is tracked by the error controller.

At some point during the deceleration the necessary speed for the motors enabling has been reached. As a consequence, the system automation evaluates check r1) as passed and subsequently sends a command to the airspeed protections. This command sets the upper limit to V_{LSNE} , ensuring that the structural integrity is maintained in the upcoming turn on process. This is depicted in step r2).

To ensure automation transparency, the imminent change of aircraft configuration is communicated to the operator in r3). The automation proceeds to communicate to the control allocation to execute the turn on process of the LTUs in r4). This is done as a ramp up to idle RPM. Step r4) formally marks the start of the retransition process according to the definition, introduced in Section 2.4.2.

As the sequence is being executed, a check is performed by the System Automation whether the LTUs are engaged, depicted in step r5). Notice that during this time the lower airspeed limit prohibits stalling the aircraft, whereas the upper limit ensures the structural integrity of the aircraft, thereby enforcing a safe system state by the design of the procedure.

Once the check of r5) evaluates a successful engagement, the operator is notified in step r6). This ensures the crew awareness that the next mode of operation - powered-lift flight - has been entered.

For recollection, in Section 3.3.1 it was mentioned that prior to entering wingborne flight, the lower airspeed limit is set, such that stalling the aircraft is not possible. Since in r5) it is ensured that the LTUs may be fully utilized, in r7) the lower limit is released. This is done by the system automation by the corresponding message to the airspeed protections.

As the aircraft continues to decelerate to lower air- and kinematic speeds, check r8) monitors whether the conditions are met to signify the end of the retransition phase. When this occurs, in step r9) the operator is notified that the hover region is reached and thus end of the retransition is indicated. Note that that the assumption made in Section 3.3.1 is confirmed since the last upper airspeed limit set is that of V_{LSNE} .

During the check of motor engagement r4), a check whether the engagement fails is executed as well. This is noted with r5a). If process fails, then in order to mitigate the event of Table 3.1, adjustment of the airspeed in r7) and subsequent deceleration is initially not permitted. Instead, r6a) is triggered, which raises the awareness of the crew that a mitigation strategy needs to be executed. This is handled in rm), which is explained in Section 3.4.1.2.

In Figure 3.3 it is visible that in the nominal case the transition is fully automated. The reason why in r8) actions from the crew are expected, is that they may require to remain the Transition/Retransition region for an extended amount of time. In this case it is not feasible to enforce the inherently low ground speed of the hover region. Furthermore, by means of r3) and r6a) the crew's mode awareness is ensured in the cases where the mitigation strategies of rm) are in effect. The mitigation and reentry in the retransition over rm3) are explained in Section 3.3.4

Furthermore, the retransition may be aborted at any time during the process, which is omitted in Figure 3.3 for the sake of readability. Should this occur after check r1) is evaluated to be true, then the transition process is triggered. The mechanics of this process are visible in Section 3.5.

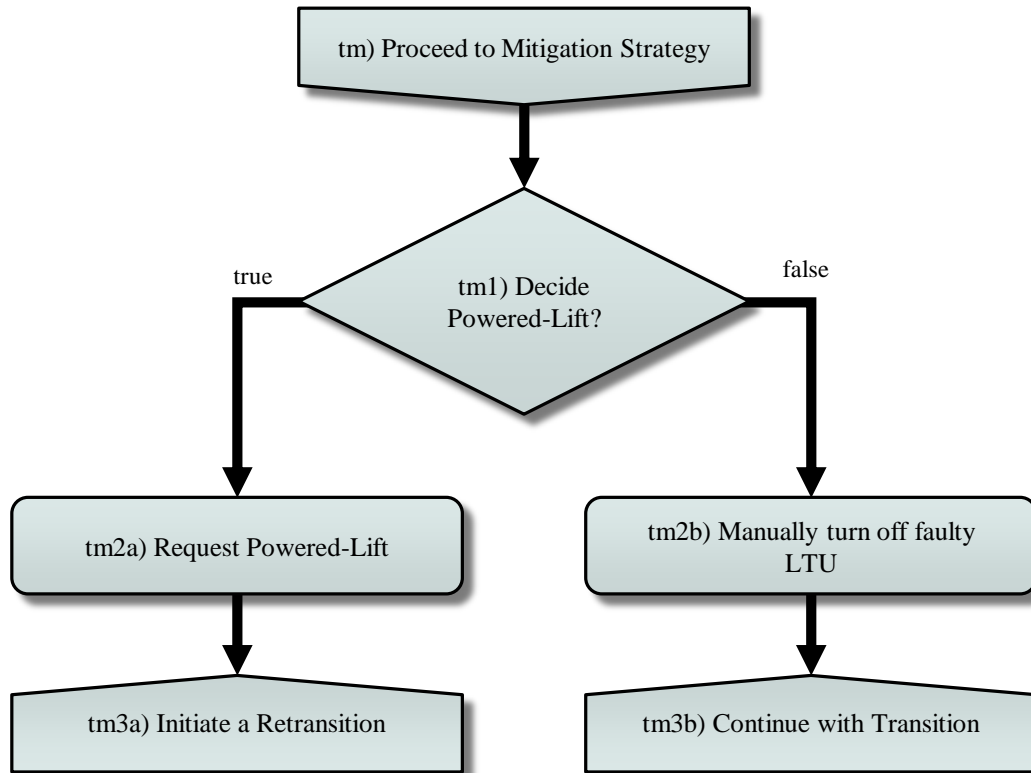


Figure 3.4: *Transition Abnormal Procedure Flowchart*

3.3.3 Transition Mitigations

In Section 3.3.1 it was mentioned that a failure to disengage the propulsion system triggers a mitigation strategy that requires the involvement of the operator. This Section examines namely this mechanism. The functional flow of this process is described in Figure 3.4.

For recollection, this abnormal procedure is initiated when a reconfiguration is taking place. Because a disengagement of an LTU and thus the reconfiguration fails to succeed, the wingborne region cannot be entered at this point.

It is mission specific what the next line of action is and therefore the decision falls onto the crew. For instance, when performing the transition in the vicinity of the take-off point, it may be more desirable to abort the mission, re-enter hover and perform a landing. If this is the case, then tm1a) is evaluated as true and the re-engagement of the LTUs is required. This is equivalent to triggering the retransition process. Therefore, in this case, the process from Figure 3.3 is called, the mechanics of which were already explained in Section 3.3.2.

On the other hand, if prior to landing covering greater distances is required, then going in the wingborne region can be done by manually turning off the failed motor which is visible in Figure 3.4 from step tm2b) onward. Subsequently, the check t7) of Figure 3.2 can be run again and if the manual disengagement was successful, then the transition proceeds as depicted in the Figures 3.4 and 3.2.

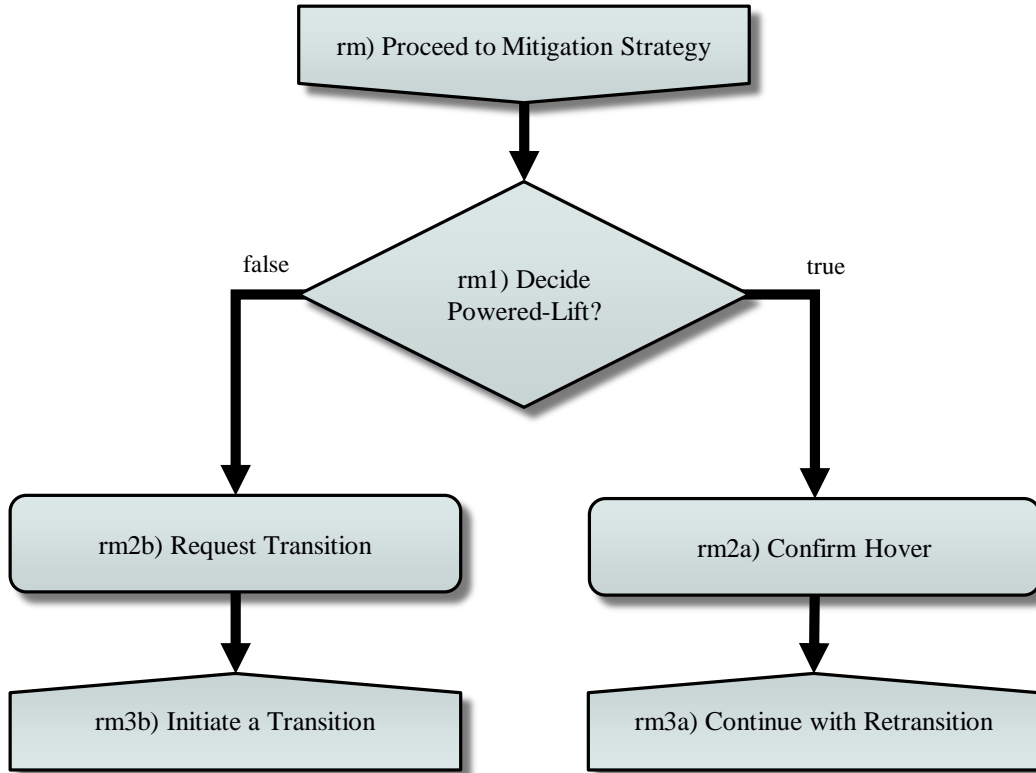


Figure 3.5: *Retransition Abnormal Procedure Flowchart*

It must be noted that prior to the initiation of these sequences, the aircraft is in a safe state - the possibility of stall or structural damage is mitigated by the airspeed protections. Therefore, the aircraft is in a stable flight condition. The decision-making process of the crew is therefore not time-critical.

Furthermore, when entering the abnormal procedure the aircraft is in the low-end of the wingborne region with an upper airspeed limit of V_{LSNE} . Assuming that a reliable manual deactivation of all LTUs is not possible by system design, then the abnormal procedure to wingborne flight is not feasible anymore as exceeding V_{LSNE} with a running LTU may be catastrophic. Then the only choice left is to enter hover flight. However, since the aircraft is in a stable flight state, the crew may decide to stay in this state for a prolonged duration of time prior to entering hover flight. Moreover, even in the absence of such a mitigation strategy, in Section 3.5 it is visible that this has no implication on the automation design whatsoever.

3.3.4 Retransition Mitigations

This section proceeds to explain the retransition abnormal procedures if rm) occurs in Figure 3.3. The retransition process flow is explained in Figure 3.5. Similarly to Section 3.3.4, the crew's decision deals with the desired flight mode which depends on the mission parameters.

From the energy considerations, it is most likely that the retransition is attempted in approach, i.e. in the vicinity of the landing port. In the cases where re-engagement of a powered-lift unit is not possible, then a go-around may be appropriate. This is highly likely going to be performed in wingborne flight. In these instances, the crew would opt for rm2b) and since the powered-lift is partially engaged as per r5) of Figure 3.3, then going into wingborne flight is done via the transition procedure of Section 3.3.1.

Important to notice in Figure 3.5 is that hover flight can still be entered via rm2a) with the appropriate crew input. In the off-nominal procedures of Section 3.3.3, the wingborne flight is limited until confirmed LTU disengagement. Here in contrast the limitation of entering a lower airspeed can be best interpreted as a warning that is removed upon the confirmation of rm2a). The reason is that many VTOL aircraft configurations are incapable of wingborne landing. Therefore, availability of the hover flight function is actively pursued in the cases of false positives.

As seen before in Table 3.1, such configurations mentioned above arguably have a fail-operational or fail-active powered-lift system, in which cases hover flight is always available. In the worst case, this flight phase can be entered with reduced performance. Even in these instances it makes sense to require confirmation from the crew instead of just entering powered-lift flight immediately. Firstly, this raises the crew awareness that if entered, powered-lift flight will be with decreased performance due to the detected error. Secondly, remaining in this flight state prior to confirmation allows for the possibility of a go-around and a reattempt of full engagement of the LTUs. In each of these cases the system transparency is enforced.

Lastly, if hover flight and a wingborne landing are both impossible, then aborting the mission via a flight termination system would be in effect. Flying to a safe zone may be required. In these cases again such checks are required, which additionally requires the system not to enter powered-lift flight “blindly”.

In all mentioned scenarios it is visible that due to the design of the procedure, the aircraft is in a safe flight state, which is guaranteed by appropriate scheduling of the airspeed limits. The limits are equivalent to the ones in Section 3.3.3. Therefore, the decision-making process is again rendered non time-critical.

3.3.5 Procedure Summary

In Section 3.3.1 the normal transition procedure was derived and supplemented with the process in the cases of an off-nominal event in Section 3.3.3. The same was performed for the retransition in Sections 3.3.2 and 3.3.4 respectively.

For the sake of readability, the processes are broken down in four figures for each of the sections above. In Appendix B the interested reader can find the full integrated process flow of the procedures.

The procedures derived in this thesis are fully automated in the cases where no failures are registered. They furthermore fit well with the existing Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics because the operator involvement is kept to its minimum.

In the cases where a failure occurs, the design of the procedures allows for a simplified decision-making process. The reason is that the automation is designed transparent and the choices at disposal for the decision-making process are limited and tied to intuitive physical reconfigurations.

Opacity is mitigated in the following way. The pilot's intended region of operation is clearly commanded via the inceptor and the pilot is supplied with feedback that this is correctly registered. Awareness is further raised by providing the operator with the progress of the process. This process is broken down and centered around the activation and deactivation of the hover propulsion system.

In the abnormal scenarios, the operator is supplied with the information that their desired region is currently unavailable. The unavailability conditions are simplistic and therefore manageable to track. They consist of the failure to engage or disengage the propulsion system in retransition or transition respectively.

In these abnormal events, the choices are to revert to the previous operational mode or to enforce the desired region. In both cases this is completed with a clear set of actions.

Even if the underlying troubleshooting and the decision-making process of the operator prove complex, the system is in a safe state at all times. Therefore, the activities are not time-critical. This alleviates the pilot workload.

In the following sections the suggested software implementation of the procedures is presented. All the process steps and how they fit in the software design are broken down and analyzed.

3.4 Automation Design

In the previous section the transition and retransition process on an aircraft level was described, taking the actions of the crew and software into account. In this section the focus is on a design solution that fulfills the requirements that are resultant from the process. This section is organized as follows.

Initially, the automation strategy is presented in Section 3.4.1 by introducing the State Machine, its input alphabet and transition functions. This State Machine does not account for the usage of a high-lift system.

If flaps are utilized, then the operation impacts the mechanics of the State Machine. Those impacts are examined and the functionality to account for the high-lift system are expanded in Sections 3.4.2 and 3.4.3. The former tackles the operation of the high-lift system only. The latter discusses the supplementation of the design in Section 3.4.1 to account for the reconfiguration state of the high-lift system.

In all three sections mentioned above, the underlying purpose of the information that is passed to the surrounding software modules is described, which is followed with information of how the data is processed.

Afterwards, the logics of the surrounding modules is presented in Section 3.4.4. The interactions of the automation with the law and control allocation are explained in Section 3.4.4.1. The information supply to FSD-SVO that enables the control concept is explained. In Sections 3.4.4.2 the automation management of airspeed limits is presented. They are necessary to enforce safety constraints. The scheduling of the high-lift system is explained next in Section 3.4.4.3. Finally, in Section 3.4.4.4 the logical decision that is necessary to supply the operator with adequate awareness is presented. With this the State Machine is fully explained.

3.4.1 Automation Strategy Without a High-Lift System

In this section the choice of automation abstraction is discussed. The System Automation proposed here is centered around the hover propulsion system behavior. Three considerations motivate this choice.

Firstly, such an abstraction layer directly corresponds to how the LTUs are to be utilized by the law, keeping the interface between law and automation lean and unambiguous. Secondly, the centering around the LTU mode of operation creates an information supply to the operator that is on a physically-intuitive and therefore transparent level. Lastly, it should be noted that according to Section 3.1.2 during the reconfiguration from and to wingborne flight a majority of potential hazards arise due to malfunctions while engaging and disengaging the powered-lift. This allows for addressing the hazards and the mitigation scenarios in a straight-forward manner.

Let M_{LTU} be the Finite-State Machine used to automate the engagement and disengagement process of the hover propulsion system of the flight control algorithms. M_{LTU} is depicted in Figure 3.6. In the figure, the transition conditions and actions, denoted in blue signify the ones that apply in nominal conditions. The ones, depicted in red are relevant for off-nominal applications. Furthermore, it can be seen that the State Machine graphical representation is divided into the flight phases (green), where the abbreviations for the flight phases stem from Section 2.4.2. The divisions are discussed later in Section 3.5.1.4. M_{LTU} 's state $s_{LTU} \in S_{LTU}$, where S_{LTU} set of states

$$S_{LTU} = \{Engaged, \\ Disengaging, \\ Disengaged, \\ Engaging\} \quad (3.1)$$

as visible from Figure 3.6. The starting state of M_{LTU} is

$$s_{LTU_0} = Engaged. \quad (3.2)$$

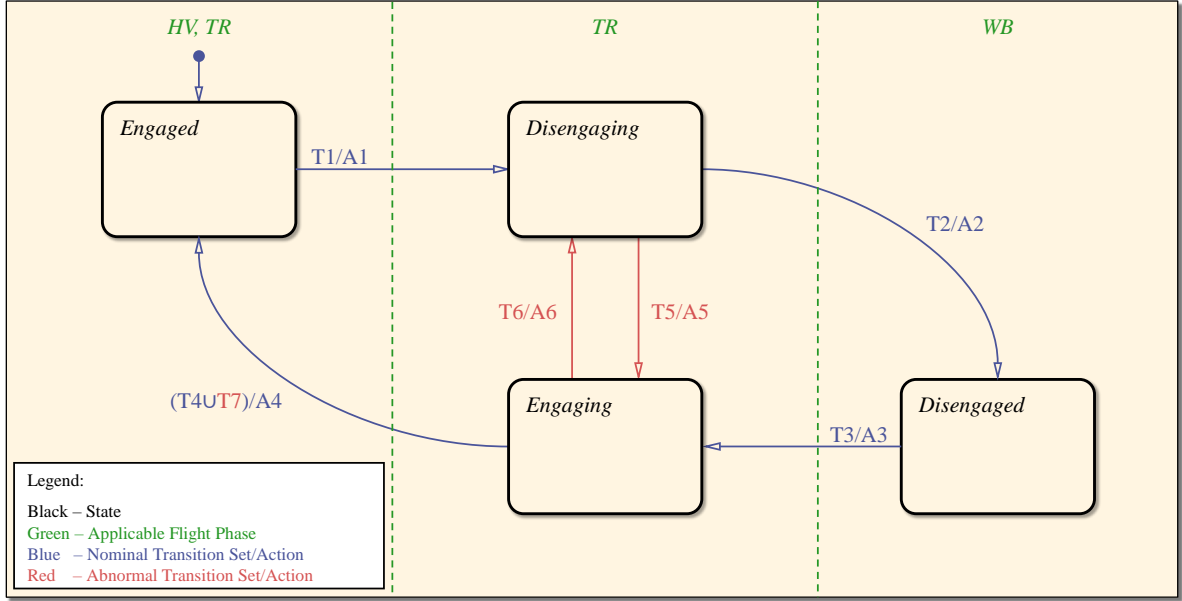


Figure 3.6: *High-Degree of Automation LTU State Machine*

Note that continuous operation of the algorithms is assumed here. Therefore, properties such as robustness against in-flight reboot or takeover from other instances of control algorithms are not considered in this chapter. In this chapter, the occasions where such properties would influence the derived considerations are mentioned in this section. The functionality with relation to such effects is described with the implementation example found in Chapter 5.

Table 3.2 summarizes the input alphabet of M_{LTU} and the meaning behind the input symbols. The underlying physical entities that they are processed from is mentioned. All of the members of the input alphabet belong to the boolean set. They are provided by the Decision-Atomics of the automation module, elaborated upon in the next section.

3.4.1.1 Decision-Atomics

The processing of the inputs, found in the Decision-Atomics is structured according to the origin of the raw signals as found in Table 3.2. Firstly, the HMI processing is explained, followed by the interpretation of the airdata system and inertial navigation system. The feedback of the LTU is explained last.

Processing the Human-Machine-Interface

The first two input symbols of Table 3.2 - $trans_{rqst}$ and $retrans_{rqst}$ are the processed operator inputs that determine whether wingborne flight or powered-lift flight is desired.

As for the choice of operator input to facilitate the command, the transition is requested whenever in the wingborne control inceptor region, as found previously is Section 2.4.3.1. In other words

$$trans_{rqst} = \chi_w(\delta_T) \quad (3.3)$$

Table 3.2: *Summary of the M_{LTU} Inputs*

Symbol	Interpretation	Signal Origin
$trans_{rqst}$	The operator requests wingborne flight.	Human-Machine-Interface
$retrans_{rqst}$	The operator requests powered-lift flight.	Human-Machine-Interface
$hover_{rqst}$	The operator requests precision hover flight.	Human-Machine-Interface
$highSpeed_{rqst}$	The operator requests flight, outside the precision hover envelope.	Human-Machine-Interface
$LTU_{override_{rqst}}$	The operator confirms the correctness of powered-lift flight reconfiguration. Used in abnormal scenarios.	Human-Machine-Interface
V_{trans}	Airspeed is above the value, where disengagement of the LTUs is deemed safe.	Airdata System. Dependent on the Configuration State
$V_{retrans}$	Airspeed is below the value, where engagement of the LTUs is deemed safe.	Airdata System
V_{high}	The kinematic speed is above the precision hover speed limit.	Inertial Navigation System
V_{low}	The kinematic speed is below the precision hover speed limit.	Inertial Navigation System
LTU_{ON}	The powered-lift system is engaged.	LTU feedback and Signal Integrity Checking
LTU_{OFF}	The powered-lift system is disengaged.	LTU feedback
LTU_{UNUSED}	The powered-lift system is engaged, but the demanded power output is low.	LTU feedback
$retrans_{timeout}$	The turn on of the powered-lift flight system fails within the predefined time frame.	Internal variable

and therefore

$$retrans_{rqst} = \chi_{\text{HUT}}(\delta_T), \quad (3.4)$$

for the following reasons. Firstly, from Section 2.4.1.1 it follows that deflecting the control inceptor from the powered-lift division to the wingborne division and vice versa provides adequate visual and haptic feedback. Hence, the pilot is aware when the need of change in operation has been processed. The subsequent configuration change thereby is transparent to the operator.

Secondly, the FSD-SVO velocity mapping provides an airspeed command that is above the stall speed with a good margin. From these two perspectives, the position alone suffices to specify which mode of operation is desired. In addition, there is no need to impose any additional complexity with regards to the crew operation of the aircraft for the reconfiguration. The throttle control inceptor being in the given region is directly interpreted as a request to enter that flight mode.

Please note that in this chapter the robustness measures to account for the effects, such as sensor noise, etc. are not demonstrated. These additional functions are in the scope of Chapter 5 and augment the evaluation introduced here.

The next two inputs - $hover_{rqst}$ and $highSpeed_{rqst}$ - indicate whether the operator requests to conduct a high-precision hover flight. This is calculated analogously as in Equations 3.3 and 3.4. Namely

$$hover_{rqst} = \chi_{\text{H}}(\delta_T) \quad (3.5)$$

and

$$highSpeed_{rqst} = \chi_{\text{TUW}}(\delta_T) \quad (3.6)$$

respectively.

$LTUoverride_{rqst}$ is again an operator input. This information originates from a discrete input and is necessary to detect that the transition to hover flight is confirmed by the operator in the event of a mitigation strategy as per rm2a) of Section 3.3.4. The need for this input is therefore derived from the procedure itself and is not reflected in the FSD-SVO. It must be noted that the exact choice of the operator actions that generate this input is outside the scope of this chapter but is discussed in Chapter 4 due to the importance of this item for the procedure harmonization. In this chapter, the variable is assumed to be a known input.

Processing the Airdata System

The next two input symbols of Table 3.2 originate from the airdata system processing. V_{trans} is true only if a sufficiently high airspeed is reached. In this thesis it is referred to as the “disengagement speed” $V_{disengage}$. Similarly $V_{retrans}$ is deemed true if the airspeed is below the so-called “engagement speed” V_{engage} . Therefore

$$V_{trans} = V_{CAS} > V_{disengage} \quad (3.7)$$

and

$$V_{retrans} = V_{CAS} < V_{engage}. \quad (3.8)$$

In the equation, V_{CAS} is the calibrated airspeed, measured by the airdata system. The two input symbols are used to check whether the envelope conditions suffice for disengagement or engagement of the hover propulsion system as seen later.

The choice of $V_{disengage}$ is derived from speed definitions found previously in Section 2.4.1.1.

$$V_{disengage} = \begin{cases} V_{OEI} & \text{if a TTU Failure is registered,} \\ V_{SAFE} & \text{otherwise.} \end{cases} \quad (3.9)$$

The terms in the two cases were introduced in Equation 2.34. The choice of disengagement speed guarantees that after the deactivation of the LTUs, the maximum obstacle clearance can be achieved regardless of the state of configuration. Prior to this, the maximum clearance can be obtained using the powered-lift system.

In order to guarantee that this speed will be reached in the first place, a clear speed command needs to be reflected in the FSD-SVO concept as well. Therefore, the requirement is that this value is mapped when in the control inceptor gate, i.e. at $\delta_{T,G}$ as introduced previously in Section 2.4.3.1. FSD-SVO takes the failure of the TTU into account via the automation and changes the mapping accordingly. At the same time, the automation modifies the check.

As for the engagement speed V_{engage} , the requirement originates from the structural limit restrictions, signified with V_{LSNE} . It can be formulated that

$$V_{engage} = V_{LSNE} - \Delta V_{LTU_{eng}}. \quad (3.10)$$

Here, the term $\Delta V_{LTU_{eng}}$ has to account for external disturbances such as gusts that may cause the airspeed to increase abruptly. Additionally, it has to account for inaccuracies in the measurement.

Processing the Inertial Navigation System

The next two input symbols - V_{high} and V_{low} - are necessary for the mode of operation information supply, i.e. HV , TR and WB . For this, the ground speed in direction of the aircraft longitudinal axis is utilized. Denoting this velocity component as V_K the symbols are calculated as

$$V_{high} = V_K > V_{HOVER} \quad (3.11)$$

and

$$V_{low} = V_K \leq V_{HOVER}. \quad (3.12)$$

Processing the LTU Feedback

The last three symbols of Table 3.2 - LTU_{UNUSED} , LTU_{ON} and LTU_{OFF} - are based on the feedback of the LTU revolution rates and the integrity of the units. They determine

whether the hover propulsion system is not currently required by the law for force and moment production and whether the hover propulsion system has been fully engaged and disengaged respectively.

For the processing of these inputs, $i = \{1 \dots nLTU\}$ is introduced, where $nLTU$ is the number of propulsion units of the aircraft. Then let ω_i be the the revolution rate feedback of the i 'th LTU. Let ω_{idle_i} be the assumed idle revolution rate of that LTU and Δ_i be a buffer to account for uncertainties in the assumed idle revolution rate that may be higher in reality due to measurement inaccuracies. Lastly, LTU_{avail_i} indicates that the LTU has been assessed as correctly functioning by both LTU itself and the Signal Integrity Checking. Hence, LTU_{ON} and LTU_{OFF} are true if for all i

$$\omega_i > \omega_{idle_i} - \Delta_{idle_i} \wedge LTU_{avail_i} \quad (3.13)$$

and

$$\omega_i < \Delta_{off_i} \quad (3.14)$$

respectively.

The last input symbol is computed as

$$LTU_{UNUSED} = Confirm(LTU_{idle}, t_{unused}), \quad (3.15)$$

where *Confirm* is the Confirmation Counter as described in Equation 2.12. t_{unused} is designed to guarantee that the LTUs are indeed not used by the law for force or moment production.¹ $LTU_{idle} \in \mathbb{B}$ is a check that the propulsion system is near the idle revolution rates. LTU_{idle} is true if

$$\omega_i < \omega_{idle_i} + \Delta_{unused_i} \vee \neg LTU_{avail_i}, \quad \forall i. \quad (3.16)$$

Compared to Δ_{idle_i} from Equation 3.13, the threshold Δ_{unused_i} is much larger. The reason for this is that the law actively utilizes the LTUs for its tracking objective. Thereby, deviations from the idle revolution rates cannot be excluded. In fact, the threshold Δ_{unused_i} is a function of the current state of configuration. The value is expected to change when for example an LTU or traction unit is lost. Then there might be a net moment due to the failure that needs to counteracted and therefore the remainder of the LTUs may need to produce noticeably higher RPM. Even though not explicitly written here, ω_{idle_i} is therefore a function of the state of configuration.

Lastly, $retrans_{timeout}$ is used to track how long the system state s_{LTU} has remained in the state *Engaging*. For recollection, according to step r4) of Section 3.3.2, the control allocation actively commands the ramp up in the commanded revolution rates of the LTUs to the idle revolution rates. It is therefore deterministic how long it should take for

¹Recall that in Section 1.1.4 a eVTOL with dedicated TTUs was considered. For tilt-rotor aircraft, additionally the tilt deflection can be monitored as a marker that wingborne flight is approached. In fact, the automation we present in [114] accounts for this.

the LTUs to engage because the sequence is predefined. If this time is exceeded, then a timeout is used as a means to make one of the mitigation strategies to powered-lift flight available as explained later. It is therefore defined that

$$retrans_{timeout} = Confirm(s_{LTU} == Engaging, t_{engage,timeout}), \quad (3.17)$$

where $t_{engage,timeout}$ is tightly coupled to the known ramp up sequence duration. Permitting hover flight prior to the ramp up is not feasible.

In this section the Decision-Atomics of M_{LTU} was presented. It processed all input sources using the constructs that were presented previously in Section 2.2.3 and prepared the input alphabet for the Decision-Making. As already seen in Figure 3.6, the core of the automation is a Mealy Machine. Its transition and output functions are explained in the next section.

3.4.1.2 Decision-Making

This section explains mechanics of the Decision-Making process. It presents the transition functions of the State Machine. In Figure 3.6 the transition sets and actions are denoted as Ti and Ai respectively with $i \in \mathbb{N}$ to conform with the conventions previously introduced in Section 2.2. In the same figure the edges marked in red are the ones that deal with the mitigation strategies, explained in Sections 3.3.3 and 3.3.4.

For the sake of clarity, the actions during each state are briefly summarized with the introduction of the transition functions. The detailed explanation is provided later on in Section 3.4.4. In addition, all input combinations that are not depicted on the edges of Figure 3.6 imply that the state is retained. This follows the pattern previously explained in Section 2.2.3 with Equations 2.22 and 2.24.

Entry Point

When entering s_{LTU_0} , the surrounding systems are supplied with the following information:

- The control allocation is passed the information that the powered-lift system may fully be used for force and moment production.
- The information that the LTUs are utilized is passed to the Envelope Protections. This schedules the upper and lower airspeed limits. This is described in Section 3.4.4.2.
- The State Machine state *Engaged* is passed to the Pilot Indication to initialize the display items. This is described in Section 3.4.4.4.

As already mentioned, continuous operation of the algorithms is assumed. Therefore, properties such as robustness against in-flight reboot or takeover are not considered in this chapter. In the cases where this occurs, the entry point may not be into the state

Engaged but then depends on the flight situation. Apart from that, an in-flight reboot has no influence to the next considerations. The functionality with relation to such effects is described Chapter 5.

T1/A1: Starting the Disengagement Process

As seen in Figure 3.6, t_1 is the condition for which

$$\delta(\text{Engaged}, u_1) = \text{Disengaging}. \quad (3.18)$$

This marks the start of motor disengagement. The transition condition is defined as

$$t_1 = \text{trans}_{\text{rqst}} \wedge V_{\text{trans}} \wedge \text{LTU}_{\text{UNUSED}}. \quad (3.19)$$

The first condition of the t_1 limits the disengagement process only in the cases where desired by the operator. This fulfills t1) of Section 3.3.1, which was motivated by the automation transparency.

The next two symbols together form check t3) of Section 3.3.1. With V_{trans} the disengagement is only allowed when the aerodynamic force is enough to sustain level flight, meaning that the propulsion units should not be used for supplementary lift production. It is thereby ensured that the disengagement process is started in the correct position within the flight envelope. This is directly correlated to the findings from Figure 2.10.

However, even above the stall speed the involvement of the propulsion system may be required for enhanced disturbance rejection due to the increased moment authority. Therefore, in these conditions it is not feasible to shut them down and decrease the control performance. In contrast, the controller not actively utilizing the propulsion is indicative that the aircraft is in a calm state, where the motor shutdown can be executed. For this reason the System Automation continuously checks whether the LTUs are indeed not used via $\text{LTU}_{\text{UNUSED}}$, adding an additional independent condition.

When the transition occurs, then the following actions take place:

- The information that the LTUs may no longer be utilized is passed to the Envelope Protections. This increases the lower airspeed limit. The exact limit value is described in Section 3.4.4.2.
- The State Machine state *Disengaging* is passed to the Pilot Indication. This triggers the Shutdown indication. This is described in Section 3.4.4.4.
- The control allocation is passed the information that the powered-lift system may no longer be used for force and moment production.
- The control allocation is passed the information to initiate the ramp down the LTU commands to zero.

These actions are tupled in $A1$.

T2/A2: The Successful Disengagement

According to Figure 3.6. the condition t_2 triggers the transition

$$\delta(Disengaging, u_2) = Disengaged. \quad (3.20)$$

In the state *Disengaging* the automation is waiting for the shutdown of the propulsion system. Hence, t_2 is directly tied to t7) of the transition procedure in Section 3.3.1 and therefore

$$t_2 = LTU_{OFF} \quad (3.21)$$

must be *true*.

Whenever the transition takes place, the following series of events take place (*A2*):

- The State Machine state *Disengaged* is passed to the Pilot Indication to indicate to the operator that the transition procedure was executed. This is described in Section 3.4.4.4.
- The information that the LTUs are fully disengaged is passed to the Envelope Protections. This schedules the upper airspeed limit. This is described in Section 3.4.4.2.
- The control allocation is passed the information that the LTU commands shall not be non-zero.

The actions are summarized under *A2* in the figure.

T3/A3: Starting the Engage

The LTU engagement process is denoted with the transition

$$\delta(Disengaged, u_3) = Engaging, \quad (3.22)$$

with u_3 specified any of the input combinations that cause the expression

$$t_3 = retrans_{rqst} \wedge V_{retrans} \quad (3.23)$$

to be *true*.

Transition Condition t_3 directly implements check r1) of Section 3.3.2, therefore the operator request $retrans_{rqst}$ plays a critical role. The activation is never initiated without the explicit intent of the crew, which is expressed by that input symbol.

In contrast to t_1 , with t_3 the usage of the LTUs cannot be taken as a criteria to determine whether they are indeed necessary. This is because by implication of *A2* the active decision not to utilize them is taken. Moreover, during the activation they should not be required. If they were, this would imply the system is either in stall or the moment authority of the aerodynamic control surfaces is not sufficient. This scenario is prohibited by the proper scheduling of the airspeed limits and the envelope protections enforcing the limits.

Instead, $V_{retrans}$ is used to verify that the airspeed is sufficiently low to rotate the motors without causing structural damage. That the aircraft reaches such an airspeed is ensured by the control concept as discussed previously in this chapter - the request from the pilot $retrans_{rqst}$ implies a deceleration of the aircraft.

The transition triggers the following set of system actions, expressed with $A3$:

- The information that the LTUs are no longer disengaged is passed to the Envelope Protections. This schedules the upper airspeed limit. This is described in Section 3.4.4.4.
- The State Machine state *Engaging* is passed to the Pilot Indication to inform the operator that the retransition procedure has commenced. This is described in Section 3.4.4.4.
- The control allocation is passed the information to initiate the ramp up the LTU commands to idle revolution rates.

T4/A4: The Successful Engaging

In the state *Engaging*, the automation is waiting for the hover propulsion units to engage. Should this occur, then on the one hand, deceleration to and airspeed below the stall speed is permitted. On the other, the control allocation is allowed to use the LTUs for force and moment production.

Therefore, the state transition is executed as

$$\delta(Engaging, u_4) = Engaged, \quad (3.24)$$

in which the condition is

$$t_4 = LTU_{ON}, \quad (3.25)$$

thereby fulfilling check r5) found in Section 3.3.2.

$A4$ then causes the following response:

- The information that the LTUs are utilized is passed to the Envelope Protections. This schedules the lower airspeed limit. This is described in Section 3.4.4.2.
- The State Machine state *Engaged* is passed to the Pilot Indication to visualize that the engagement process was completed without errors. This is described in Section 3.4.4.4.
- The control allocation is passed the information that the powered-lift system may fully be used for force and moment production.

T5/A5: Transition Mitigation to Powered-Lift Flight

The first abnormal scenario is examined where the disengagement process malfunctions and cannot complete as per Section 3.3.3. From the so far introduced mechanisms, it is visible that prior to the mitigation, the automation is in the state *Disengaging*. According to Section 3.3.3, a mitigation is called for if the disengagement cannot complete, i.e. t_2 is *false* for longer than a predefined duration.

In the cases where the operator decision is to revert to powered-lift flight with step tm1a), it must be accounted that the LTUs are in the disengagement process and they need to firstly be re-engaged. Hence, the transition

$$\delta(\textit{Disengaging}, u_5) = \textit{Engaging} \quad (3.26)$$

is necessary. In these cases the set member u_5 belongs to the tuples that cause

$$t_5 = \textit{retrans}_{rqst} \quad (3.27)$$

to be *true*.

In contrast to t_3 from Equation 3.23, here there is no need to verify that the aircraft is in the correct envelope. With regards to the flight condition, it is known that from t_1 the airspeed is above the disengagement speed and from *A1* the lower end of the envelope is maintained and protected. The upper end of the envelope is protected. By these means it is ensured that neither stall nor structural damage can occur prior or after the transition.

When performing the transition from *Disengaging* to *Engaging*, under *A5* the following set of actions are executed:

- The State Machine state *Engaging* is passed to the Pilot Indication to indicate to the operator that the retransition procedure has commenced. This is described in Section 3.4.4.4.
- The control allocation is passed the information to initiate the ramp up the LTU commands to idle revolution rates.

In *A5* it is not required to pass information to the Envelope Protection limits because the upper limit was already set correctly and need not change. The lower limit was altered prior to entering *Disengaging* with *A1*. The control allocation need not receive the information that the LTUs cannot be utilized, as this was done previously with *A1*.

It is important to repeat that the other option according to the mitigation strategies of Section 3.3.3 is to manually power off the malfunctioning LTU as per tm2b). This is, however, already indirectly included in the transition condition t_2 via Equation 3.21, as turning off the motor unit implicates that the rotation rate has to converge to zero.² Hence, there is no need to account for the abnormal scenario in the design.

²This is only partially true. Turning off the power supply to a LTU may cause it to windmill. This, however, is of no concern in terms of automation design. Either the LTU is capable of turning off or not. Therefore, either this transition is possible or not, but it does not impact the design.

T6/A6: Retransition Mitigation to Wingborne Flight

In the cases of retransition, the transition function

$$\delta(\textit{Engaging}, u_6) = \textit{Disengaging} \quad (3.28)$$

implements the pilot decision rm2b) of Section 3.3.4 to abort the retransition and revert to wingborne flight. This is the case when the engagement of the motors does not complete satisfactory, i.e. t_4 is *false*. Therefore, it is stated that u_6 is from the set of tuples, for which

$$t_6 = \textit{trans}_{rqst} \quad (3.29)$$

is *true*.

Similarly to the previous mitigation actions, unlike the nominal disengagement initiation from Equation 3.19, the aircraft is in the correct envelope - the LTUs were in the engagement process from $A3$ and are not used by implication. With regards to the flight condition, from t_3 it follows that the airspeed is below the engagement speed and with $A3$ it is additionally protected. The lower airspeed limit was not modified from $A1$ onward.

Therefore, the actions under $A6$ are as follows:

- The State Machine state *Disengaging* is passed to the Pilot Indication. This triggers the Shutdown indication. This is described in Section 3.4.4.4.
- The control allocation is passed the information to initiate the ramp down the LTU commands to zero.

As with the previous mitigation strategy, the airspeed limits need not be altered.

T7/A4: Retransition Mitigation to Powered-Lift Flight

The last transition to explain is

$$\delta(\textit{Engaging}, u_7) = \textit{Engaged}, \quad (3.30)$$

which is responsible for satisfying rm1a) of Section 3.3.4. The transition condition is introduced as

$$t_7 = \textit{retrans}_{timeout} \wedge \textit{LTU}_{override}_{rqst}, \quad (3.31)$$

in which the latter argument confirms the operator decision, found under rm2a), whereas the first argument implements a timeout for the control allocation.

Recall that when entering the state *Engaged*, the controller is required to actively use the powered-lift system for its tracking objective. With $A3$ an RPM command ramp up is executed that takes a known finite amount of time to reach the end command value. By means of $\textit{retrans}_{timeout}$ in Equation 3.31, the state *Engaged* is only available when this predefined time has elapsed.

With regards to the actions taken when doing the transition t_7 , they are $A4$ as seen in Figure 3.6. $A4$ was explained previously in this section.

Kinematic Speed and Speed Demand

As for the law, recall from Section 2.4.1.1 that the necessary supply is the mode of operation, i.e. HV , TR and WB . For this a Latch is utilized. The mechanics of this construct were explained in Section 2.2.3 with Equation 2.9.

Defining the two conditions as

$$t_{HS} = highSpeed_{rqst} \wedge V_{high} \quad (3.32)$$

and

$$t_{LS} = hover_{rqst} \wedge V_{low} \quad (3.33)$$

the additional Decision-Making output is $s_{HS} \in \mathbb{B}$ and is computed as

$$s_{HS} = Latch(t_{HS}, t_{LS}). \quad (3.34)$$

As previously mentioned, here in-flight reboot or a takeover from another flight control algorithm is not considered.

3.4.2 Introducing the High-Lift System Automation

The automation strategy for topologies that in addition require the use of a high-lift system is examined next. In this section, requirements for both the flap operation and the supplementation of the transition and retransition automation are set.

Firstly, for the sake of modularity, the existing automation strategy presented in Section 3.4.1 needs to be utilized. Therefore, the automation shall use an additional State Machine for the control of the flap motion. This State Machine shall be denoted with M_{HL} . As seen in the next paragraphs, M_{LTU} is supplemented to account for the operation of the flaps.

In terms of operation order in the transition phase, the higher lift production with deployed flaps is to be taken advantage of so as to execute the LTU disengagement process at a lower airspeed. Therefore, it follows that during the transition to wingborne flight full deployment of the high-lift system must be utilized. By implication, in the automation of the transition it must additionally be considered that a malfunction in the high-lift system may hinder the mentioned full deployment.

A similar rationale is made when observing the retransition phase. The aim is to execute it with fully extended flap system to increase the theoretically permissible turn on region as per Figure 2.10. Accounting for malfunctions is again necessary to ensure no stall occurs and that the greatest obstacle clearance can be reached if full deployment cannot be established.

From the last two requirements it is evident that in the nominal conditions an aircraft shall perform the entry to wingborne and powered-lift flight prior to retracting or after extending the high-lift system respectively. This therefore sets the nominal operation order for both cases. In the transition, first disengagement of the LTUs must take place

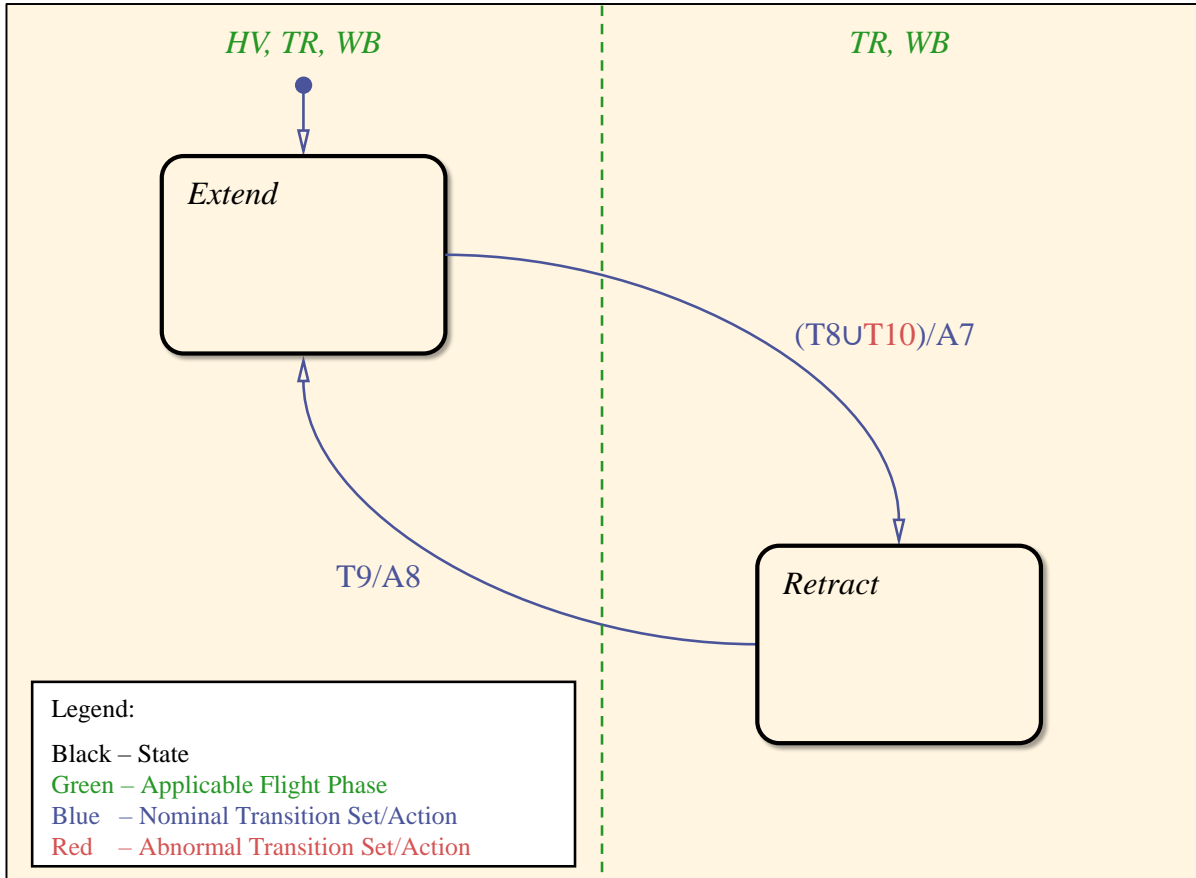


Figure 3.7: *High-Degree of Automation High-Lift System State Machine*

and only then is movement of the high-lift system out of the extended position allowed. For the retransition - it is the other way around - first fully deployed high-lift system is desired and then the LTU engagement can proceed.

This order of execution is important for the pilot situational awareness. In fact, this execution order needs to be maintained as much as possible also in the cases of faults in either system (LTU or high-lift) in order to minimize the automation opacity. Additionally, the awareness needs to be supported by coupling the operation of the high-lift system with the operator input.

Whenever the execution order cannot be maintained, adequate indications are necessary. One example for this is in the cases where the high-lift system has malfunctioned and remains fully retracted during the retransition process.

With this in mind, the State Machine for automation of the high-lift system is introduced. Let M_{HL} be the Finite-State Machine used to control the high-lift system. M_{HL} is depicted in Figure 3.7. The color pattern of the figure follows the one previously introduced with Figure 3.6. The states of M_{HL} belong to the state set

$$S_{HL} = \{Extend, Retract\}, \quad (3.35)$$

and furthermore

$$s_{HL_0} = Extend, \quad (3.36)$$

which is the initial state of the State Machine.

The input alphabet of M_{HL} that makes up the transition conditions of Figure 3.7 is summarized in Table 3.3 together with the physical entities that are used to process them. All of the entries are from the boolean domain. Each symbol is shortly elaborated in the next section.

Table 3.3: Summary of the M_{HL} Inputs

Symbol	Interpretation	Signal Origin
$extend_{rqst}$	The operator requests extension of the high-lift system.	Human-Machine-Interface
$retract_{rqst}$	The operator requests retraction of the high-lift system.	Human-Machine-Interface
$HL_{override_{rqst}}$	The operator confirms that aerodynamically efficient flight is desired. Used in abnormal scenarios.	Human-Machine-Interface
$V_{retract}$	Airspeed is above the value, where retraction of the high-lift system is deemed safe.	Airdata System
V_{extend}	Airspeed is below the value, where extension of the high-lift system is deemed safe.	Airdata System
$LTU_{diseng_{timeout}}$	The LTU disengagement process has timeout out. Used in abnormal scenarios.	Internal Variable

3.4.2.1 Decision-Atomics

The processing of the inputs, found in the Decision-Atomics, is structured according to the origin of the raw signals as found in Table 3.3. Firstly, the HMI processing is explained, followed by the interpretation of the airdata system. The internal feedback is explained last.

Processing the Human-Machine-Interface

The first two input symbols - $extend_{rqst}$ and $retract_{rqst}$ of Table 3.3 - originate from the pilot input and indicate the operator intention - whether to have the flaps extended or

retracted. The constraint

$$extend_{rqst} \neq retract_{rqst} \quad (3.37)$$

must hold to guarantee ambiguity in the operator intentions. Furthermore, to maintain the necessary execution order mentioned earlier in terms of operator input, the implications that

$$retract_{rqst} \implies trans_{rqst} \quad (3.38)$$

and

$$retrans_{rqst} \implies extend_{rqst} \quad (3.39)$$

is defined. Simply put, if a retraction is requested, then the transition command must have been commanded simultaneously at the latest. Similarly, from the second equation it follows that whenever the retransition request gets processed by the automation, then also the operator input to extend the flaps has to be registered at the same time instance at the latest.

With the two implications requirements on the interpretation of the HMI inputs to the automation are set. These are necessary to facilitate the proper mode awareness. If the pilot retransition command were to be independent of the extension command, it could very well be that the pilot demands the retransition without demanding the extension. In this example, the operator's mental picture could be that the retransition should be performed without engagement of the flaps which directly conflicts the order that must be maintained. Thereby automation transparency can no longer be claimed.

Instead, the two above-mentioned constraints by proper design of the logical decisions are enforced. The evaluation is

$$retract_{rqst} = \chi_{\mathbb{W}}(\delta_T). \quad (3.40)$$

Logically, it follows that

$$extend_{rqst} = \chi_{\mathbb{W} \cup \mathbb{T}}(\delta_T). \quad (3.41)$$

The evaluation is the same as for the transition and retransition requests, i.e. the implications previously introduced in this section can be proved. The coupling of $extend_{rqst}$ and $retract_{rqst}$ is solely to the pilot throttle lever. Thereby no additional functionality is required from the HMI. However, to maintain the desired execution order, the execution order by the State Machine design needs to be enforced instead. This is presented later on.

The third operator input - $HL_{override_{rqst}}$ - is used for the abnormal scenario where an LTU fails to disengage and - as per Section 3.3.4 - wingborne flight for a prolonged duration of time is required. This override will force the high-lift system to the state *Retract* if other conditions are also met. It must be noted that the exact choice of the operator actions that generate this input is outside the scope of this chapter but is discussed in Chapter 4 due to the importance of this item for the procedure harmonization. In this chapter, the variable is assumed to be a known input.

Processing the Airdata System

The next two inputs are coupled to the flight condition and are necessary to limit the operation of the high-lift system in the correct flight envelope with relation to the airspeed. The first of the pair - $V_{retract}$ - is evaluated as true if a sufficiently high airspeed is obtained. This speed is referred to as the “retractions speed”. Similarly, if the airspeed is below the so-called “extension speed”, then V_{extend} is set to true. As seen later, these conditions are a guarantee that the conditions for stall or structural damage are omitted. Therefore

$$V_{retract} = V_{CAS} > V_{SAFE_{FE}} \quad (3.42)$$

and

$$V_{extend} = V_{CAS} < V_{FE0} - \Delta_V. \quad (3.43)$$

The term Δ_V is used to account for external disturbances such as gusts that may cause the airspeed to exceed the structural limits shortly.

Processing the Internal Variables

The last input symbol is required for the abnormal scenarios when an LTU disengagement is not possible as discussed in Section 3.1.2. As previously explained, from the actions during the state *Disengaging* the ramp of the command to zero revolution rates is commanded. Therefore, the failure condition can functionally be accounted for by means of the timeout

$$LTU_{diseng_{timeout}} = Confirm(s_{LTU} == Disengaging, t_{ramp}) \quad (3.44)$$

where t_{ramp} specifies the known ramp down duration.

In the next paragraphs the state transition functions of M_{HL} are explained. They are seen in Figure 3.7.

3.4.2.2 Decision-Making

This section explains mechanics of the Decision-Making process. It presents the transition functions of the State Machine. This section and Figure 3.7 follow the convention of Sections 3.4.1.2 and 3.6.

For the sake of clarity, the actions during each state are briefly summarized at the end of the section. The detailed explanation is provided in Section 3.4.4. In addition, all input combinations that are not depicted on the edges of Figure 3.6 imply that the state is retained. This follows the pattern previously explained in Section 2.2.3 with Equations 2.22 and 2.24.

T8: Starting the Retraction

Nominally, the retraction process begins with the transition function

$$\delta(Extend, u_s) = Retract, \quad (3.45)$$

in which u_8 of the tuple set, for which

$$t_8 = retract_{rqst} \wedge V_{retract} \wedge (s_{LTU} == Disengaged) \quad (3.46)$$

is *true*.

The first term is necessary for adequate transparency and limits the process if not desired by the operator. With $V_{retract}$, the movement can only commence when the aerodynamic force is enough to sustain level flight and optimal obstacle clearance even with retracted high-lift system, ensuring that the retraction occurs in the correct flight envelope. Finally, the last check guarantees that the order of execution is maintained. In the nominal case the retraction starts after proper LTU disengagement. This is enforced by the check.

T9: Starting the Extension

The High-Lift deployment process is denoted with the transition

$$\delta(Retract, u_9) = Extend, \quad (3.47)$$

with the transition condition

$$t_9 = extend_{rqst} \wedge V_{extend}. \quad (3.48)$$

Extension is not initiated without the explicit intent of the crew, expressed by the first input symbol of check t_9 . With V_{extend} , on the other hand, it is verified that the airspeed is sufficiently low so as to not cause structural damage. Ensuring that this airspeed is reached by the control concept as discussed previously in Section 3.4.1.1 because the request $extend_{rqst}$ also implies an aircraft deceleration.

T10: Retracting in the event of a LTU Disengagement Malfunction

Availability of the retraction function in the events that the disengagement process does not execute correctly is accounted for. Recall from Section 3.3.3 that the aircraft may sustain flight at a high airspeed for a prolonged duration. The transition function here is motivated by this use-case, as in these occurrences reduction of the aircraft drag is desirable to increase the flight range.

The abnormal start of high-lift system retraction is signified with the transition function

$$\delta(Extend, u_{10}) = Retract, \quad (3.49)$$

in which combination found in u_{10} satisfy

$$t_{10} = retract_{rqst} \wedge V_{retract} \wedge LTU_{disengtimeout} \wedge HL_{override}_{rqst}. \quad (3.50)$$

When comparing t_{10} to check t_8 from Equation 3.46, the movement is only permitted when the clear intent of the operator is processed and if the correct envelope in terms of airspeed is maintained. Whenever the disengagement of the LTUs fails, then this is captured by means of the timeout. It is then up to the pilot's choice whether the retraction proceeds. This is done with the override.

Transition Actions

The state s_{HL} of the automata is passed to the Pilot Indications to display the current automation process. This state is also passed to the high-lift system deflection scheduling. The mentioned scheduling is described in Section 3.4.4.3.

Extend is reached at the entry point or via t_8 and therefore this set of actions are taken also at $A8$, whereas *Retract* is reached via t_8 or t_{10} and therefore this information is supplied with $A7$ as seen in Figure 3.7. A detailed explanation of the output functions is provided later on in Section 3.4.4

3.4.3 Supplementing the Powered-Lift Automation to Account for High-Lift System Operation

In order for M_{LTU} to take the high-lift system operation into account, the input symbols of Table 3.2 of the State Machine are supplemented with the ones, found in Table 3.4.

Table 3.4: M_{LTU} Input Supplement for High-Lift System Operation

Symbol	Interpretation	Signal Origin
HL_{ext}	The high-lift system is fully extended.	High-Lift Feedback
HL_{avail}	The high-lift system has no malfunctions. Used in abnormal scenarios.	Signal Integrity Checking. Internal Variable
$HL_{timeout}$	The High-Lift Extraction has timed out. Used in abnormal scenarios.	Internal Variable

How the input symbols originate in the Decision-Atomics is explained in the following section.

3.4.3.1 Decision-Atomics Supplement

Processing the Flap Feedback

HL_{ext} signifies whether the high-lift system is fully extracted. This symbol is necessary to facilitate the proper execution order of the two systems in the cases of the retransition. If δ_{F_i} is the deflection of an arbitrary flap and $\delta_{F_i max}$ is the deflection when δ_{F_i} is fully extracted, then HL_{ext} is true when

$$\delta_{F_i} \geq \delta_{F_i max} - \Delta_{F_i max}, \forall i. \quad (3.51)$$

The term $\Delta_{F_i max}$ accounts for sensor inaccuracies.

Processing the Internal Variables

The latter two inputs symbols in Table 3.4 are necessary for the abnormal scenarios when the high-lift system is not responsive. HL_{avail} originates from the failure detection mechanisms of the Signal Integrity Checking and signifies an error of the system.

For the cases of an undetected erroneous, the malfunction can functionally be accounted for by introducing that

$$HL_{timeout} = Confirm(s_{HL} == Extend \wedge V_{CAS} \leq V_{full,ext}, thresh_{extract}). \quad (3.52)$$

The timer starts running when extraction of the high-lift system is required and when the velocity for complete extension has been reached. The exact value for this airspeed $V_{full,ext}$ is introduced later with Equation 3.71 in Section 3.4.4.3. The timeout time $thresh_{extract}$ accounts for the necessary time for full deployment.³ It is visible that if HL_{ext} is never true during an undetected malfunction, this input symbol guarantees the liveness of the retransition function.

It is apparent that the transition conditions of M_{LTU} , found in Section 3.6 require modification.

When starting the disengagement process, the reconfiguration state is indirectly addressed in terms of high-lift system deployment by choice of $V_{disengage}$. This is especially necessary for off-nominal cases.

Apart from that, to start the engagement of the LTUs, the automation needs to enforce the sought after execution order but also ensures liveness of powered-lift flight for high-lift system malfunctions.

3.4.3.2 The Modified Transition Conditions

When taking the high-lift system into account, clearly changes in the State Machine M_{LTU} are necessary. In terms of input processing, only the disengagement speed $V_{disengage}$ needs alterations to account for a possible malfunction of the flaps. Additionally, the transition conditions for the start of LTU engagement need modifications.

Modifying the Disengagement Speed

The disengagement speed of Equation 3.9 is redefined to be

$$V_{disengage} = \begin{cases} V_{SAFE_{FE}} & \text{if } HL_{ext} \wedge TTU_{avail_i}, \forall i \\ V_{SAFE} & \text{if } \neg HL_{ext} \wedge TTU_{avail_i}, \forall i \\ V_{OEI_{FE}} & \text{if } HL_{ext} \wedge \neg TTU_{avail_i}, \text{ for any } i \\ V_{OEI} & \text{if } \neg HL_{ext} \wedge \neg TTU_{avail_i}, \text{ for any } i. \end{cases} \quad (3.53)$$

³One can see that this timeout method relies on very simplistic estimates. If more precise knowledge of the high-lift system mechanics is available, then this can be considered to increase the performance. This, however, will also undoubtedly increase the complexity.

In this equation, $HL_{extended}$ was introduced in Equation 3.51.

In the first line of the piece-wise function, the nominal case is depicted. Performing the LTU disengagement is enabled at lower airspeeds. The rest are abnormal scenarios, where the last would implicate a double error and is mentioned for the sake of completeness. Another property worth mentioning is that the previous definition of $V_{disengage}$ found in Equation 3.9 is equivalent to the current one for the high-lift system not being extended.

$\overline{T3}$ and $\overline{T5}$: Modifying the Engagement Transition Conditions

The modified motor engagement process is triggered by the transition functions

$$\delta(Disengaged, \overline{u_3}) = Engaging \quad (3.54)$$

and

$$\delta(Disengaging, \overline{u_5}) = Engaging. \quad (3.55)$$

The conditions $\overline{t_3}$ and $\overline{t_5}$ signify the modified conditions.

The supplementing condition

$$t_{HL} = HL_{ext} \vee \neg HL_{avail} \vee HL_{timeout} \quad (3.56)$$

is introduced, with which the modified conditions

$$\overline{t_3} = t_3 \wedge t_{HL} \quad (3.57)$$

and

$$\overline{t_5} = t_5 \wedge t_{HL} \quad (3.58)$$

is expressed. t_3 and t_5 are introduced with Equations 3.23 and 3.27 respectively.

The first condition of t_{HL} enforces the execution order in the nominal condition - engagement of the LTUs only commences once full extension of the flaps has taken place. The latter two conditions are necessary for the cases of high-lift system malfunctions. If such an error is detected by the Signal Integrity Monitoring, then the second term of t_{HL} is true and therefore the engagement is permissible from the perspective of the flap deployment. The last check is necessary for the cases of an undetected erroneous so as the liveness of the engagement is guaranteed regardless of the high-lift system operation.

3.4.4 Decision-Execution

With the State Machines of the automation defined, this section proceeds to specify the exact actions which are requested from the surrounding functional modules. Similar to Equation 2.27, the states of M_{LTU} and M_{HL} (s_{LTU} and s_{HL} respectively) are forwarded to the Decision-Execution module along with the state s_{HS} .

This section begins with Section 3.4.4.1, in which the operation of the control allocation and law with regards to the hover propulsion system is explained. Next, in Section 3.4.4.2, the limits that must be enforced by the airspeed protection function are examined. If a high-lift system is utilized - in Section 3.4.4.3 the intended system operation is explained.

Lastly, the behavior of the indications in Section 3.4.4.4 is explained. Those are necessary for proper crew awareness and correct vehicle operation.

3.4.4.1 Law and Control Allocation Scheduling

The first module that requires input from the Automation function is the control allocation, which is the sole instance of the control algorithm responsible for powered-lift command calculation. The mode of operation for the control allocation is solely coupled to s_{LTU} .

The first property is the allowed utilization of the hover propulsion system. This is allowed if $s_{LTU} == Engaged$, otherwise the powered-lift system cannot be used in the pseudocontrol distribution. Whenever the powered-lift system cannot be utilized, i.e. $s_{LTU} \neq Engaged$, the control allocation follows a predefined command pattern.

From Equation 3.19, it is known that the the previously commanded motor revolutions are in the proximity of the idle revolution rates. Whenever $s_{LTU} == Disengaging$, then the control allocation is given the task to drive the powered-lift system from the current command down to zero. Generally speaking, this is done in the form of a ramp and considers no change in the estimated net moment due to the hover propulsion units but is not in the scope of this thesis.

Similarly, whenever $s_{LTU} == Engaging$, the control allocation executes a predefined command ramp-up to idle. On one hand, this facilitates the check of proper motor engagement by the Automation, provided in Equation 3.13. On the other hand, this also allows for a smoother entry into the transition flight phase as the active pseudocontrol allocation starts from in a more dynamically deterministic RPM envelope when compared to starting the control allocation from zero RPM.

Lastly, when $s_{LTU} == Disengaged$, the aircraft is fully in wingborne flight. There the control allocation sends zero commands to all components of the powered-lift system.

With the states s_{LTU} and s_{HS} the behavior modes the law requires can fully be described and namely by

$$\begin{aligned} HV &= (s_{LTU} == Engaged) \wedge \neg s_{HS}, \\ TR &= (s_{LTU} == Engaged) \wedge s_{HS} \text{ and} \\ WB &= (s_{LTU} \neq Engaged). \end{aligned} \tag{3.59}$$

It is easily seen that only one of the modes can be active at a given time instance. The behavior of the closed-loop control algorithms is not in the scope of this thesis, but can instead be found in [8].

3.4.4.2 Airspeed Limit Scheduling

The next output function presented is the airspeed limit computation. Firstly, the lower limits are studied. The values that the lower airspeed limit V_{CASmin} can assume are seen in the Truth Table 3.5.

Table 3.5: *Underspeed Protection Limit Truth Table*

s_{LTU}	HL_{ext}	$\neg TTU_{avail_i}$, for any i	V_{CASmin}
<i>Engaged</i>	-	-	unused
$\neg Engaged$	true	false	$V_{SAFE_{FE}}$
$\neg Engaged$	false	false	V_{SAFE}
$\neg Engaged$	true	true	$V_{OEI_{FE}}$
$\neg Engaged$	false	true	V_{OEI}

The inputs to the table is the state of our automata s_{LTU} , whether or not the high-lift system is fully extended and whether an error in the traction system is registered. For recollection, the computation whether the high-lift system is extended was introduced with Equation 3.51.

As discussed from Section 3.4.4.1, whenever $s_{LTU} == Engaged$, then the control allocation has full command of the hover propulsion system. Therefore, the lower airspeed limit is no longer necessary as stall is mitigated by the the powered-lift production.

Whenever the hover propulsion system may not be used, i.e. $s_{LTU} \neq Engaged$, then the airspeed limit is solely a function of the aircraft configuration state. The characteristics of the safe speed are directly influenced by the high-lift system and whether an error in the traction system is registered. The subsequent lines of Table 3.5 reflect those configuration changes.

Please note that is theoretically possible to account for the high-lift system deployment by mapping scheduling the limit from V_{SAFE} to $V_{SAFE_{FE}}$ as a function of the state of their deployment. Here it is explicitly chosen for a more conservative and thus safe approach. Hence, only the lower speed of $V_{SAFE_{FE}}$ is allowed if there is a confirmed full deflection of the high-lift system.

In the cases, where the aircraft does not have a high-lift system, the logic is condensed by removing the configuration changes with relation to that system and taking only the entries, where the high-lift system is not extended. In other words, the differentiation “Flaps extended” need not be made anymore.

Table 3.6 for the upper airspeed limits follows a similar approach. Here, the input symbol LTU_{OFF} is used instead of the state of M_{LTU} . For recollection, the computation of the symbol is presented in Equation 3.14. The reason the state is avoided is for the cases where hardovers of the hover propulsion system occur after entry into wingborne flight. If such an error occurs, then the automation either forbids an acceleration beyond V_{LSNE} or initiates a deceleration until LTU fault detection and isolation algorithms manage to cope with the issue.

Table 3.6: *Overspeed Protection Limit Truth Table*

LTU _{OFF}	High-Lift System	V _{CASmax}
false	-	V _{LSNE}
true	¬Retracted	min(V _{sch} (δ _F))
true	Retracted	V _{NE}

Whenever the powered-lift system is disengaged, the state of the high-lift system is observed once more - in these cases a retraction of the flaps is checked. The high-lift system is evaluated as retracted if

$$\delta_{F_i} \leq \delta_{F_i min} + \Delta_{F_i min}, \quad \forall i. \quad (3.60)$$

The symbols of the equation were introduced in Equation 3.51. In these cases, the upper limit is relaxed to V_{NE}.

Otherwise - in the cases where Equation 3.60 is evaluated as false, then the upper airspeed limit is scheduled over the flap deployment, ensuring that no structural damage ensues. For every flap deflection measurement δ_{F_i} a speed is calculated, where no structural damage occurs with the function

$$V_{sch_i}(\delta_{F_i}) = V_{FEfull} + (\delta_{F_i} - \delta_{F_i max}) \cdot \frac{V_{FE0} - V_{FEfull}}{\delta_{F_i min} - \delta_{F_i max}}. \quad (3.61)$$

The safe airspeed is computed for all flap deflections and the lowest one is taken as seen in row two of Table 3.6.

3.4.4.3 Deriving the High-Lift Scheduling

This Section demonstrates how the high-lift system automation states *Extend* and *Retract* of Section 3.4.2 are used for the flap utilization. Here exact strategy for the deflection scheduling is discussed.

From the design execution order of LTU and high-lift system operation it is known that in order to enter *Retract* the LTUs need to be disengaged, i.e. wingborne flight has been requested.

Therefore the flap command strategy when in *Retract* is formulated in the following manner. Whenever $s_{HL} == Retract$, the system shall attempt to deflect the flaps, such that the aircraft is flying in an aerodynamically optimal configuration with respect to the high-lift system.

For this, the force-equilibrium equation around the kinematic frame without wind is observed. For steady-state flight from [100] the equation

$$0 = \frac{T - D}{mg} - \sin \gamma \quad (3.62)$$

has to hold. In the equation T is the currently applied thrust, m is the aircraft mass, γ is the climb angle. D is the aircraft drag is expressed as

$$D = \bar{q}S \cdot C_D(\boldsymbol{\delta}_F), \quad (3.63)$$

where \bar{q} is the dynamic pressure and S is the surface. The drag coefficient C_D is a function of the deflections of all high-lift units. From [100] it follows that

$$C_D(\boldsymbol{\delta}_F) = C_D(C_L(\boldsymbol{\delta}_F)). \quad (3.64)$$

Hence, all parameters that make up C_D are a function of the flap deflections.

For the given flight condition of Equation 3.62, the demanded forward thrust can be reduced by minimizing the drag and therefore the drag coefficient C_D . This is an optimization problem, subject to the following constraint.

The constraint arises from the force-equilibrium equation about the body-fixed x-Axis as per [100], i.e.

$$0 = \cos \mu \cdot L - mg \cdot \cos \gamma, \quad (3.65)$$

with the Lift expressed as

$$L = \bar{q}S \cdot C_L(\boldsymbol{\delta}_F). \quad (3.66)$$

For steady-state straight and level flight the lift coefficient C_L has to satisfy the constraint

$$C_L(\boldsymbol{\delta}_F) \stackrel{!}{=} \frac{mg}{\bar{q}S}. \quad (3.67)$$

Therefore, whenever $s_{HL} == \text{Retract}$, the system flap command shall satisfy be the solution of

$$\begin{aligned} C_D(\boldsymbol{\delta}_F) &\stackrel{!}{=} \min_{\boldsymbol{\delta}_F} C_D(\boldsymbol{\delta}_F) \\ \text{s.t. } &h_{C_L}(\boldsymbol{\delta}_F) = 0, \end{aligned} \quad (3.68)$$

where

$$h_{C_L}(\boldsymbol{\delta}_F) = C_L(\boldsymbol{\delta}_F) - \frac{mg}{\bar{q}S}. \quad (3.69)$$

From the perspective of the scheduling, \bar{q} in an input that originate from the sensor feedback. Given the flight condition, a deflection that shall minimize the aircraft drag can be found. On one hand, the necessary thrust to maintain steady state is lowered. On the other, the possible specific excess powered is increased, facilitating either faster climb gradients or higher forward acceleration rates.

The strategy whenever $s_{HL} == \text{Extend}$ from the execution order implicates that the system is either in powered-lift flight or a retransition to this flight condition is desired. As a consequence faster deceleration rates are required. Because with increasing extension of the high-lift system the drag also increases, ideally the flaps should deploy as much as

possible. However, the structural limits need to be taken into account. The flap deflection commands need to be calculated as

$$\delta_{F_i,cmd} = \begin{cases} \delta_{F_i,max} & \text{if } V_{CAS} < V_{FEfull} - \Delta_V \\ \delta_{F_i,min} & \text{if } V_{CAS} > V_{FE0} - \Delta_V \\ \delta_{F_i,max} - (V_{CAS} - V_{FEfull} + \Delta_V) \cdot \frac{\delta_{F_i,max} - \delta_{F_i,min}}{V_{FE0} - V_{FEfull}} & \text{otherwise.} \end{cases} \quad (3.70)$$

The term $\Delta_V > 0$ is necessary to account that disturbances do not temporarily increase the dynamic pressure above the structural limits. From Equation 3.70, the speed, below which the command should be $\delta_{F_i,max}$ is determined as

$$V_{full,ext} = V_{FEfull} - \Delta_V. \quad (3.71)$$

For recollection, this value of this parameter was necessary in Section 3.4.3.1.

The command mechanisms in the events of failures are not explained here. For example, the command mapping must change if a hardover in one of the high-lift units is registered. The methods to tackle such issues are application-specific and hence not in the scope of this chapter. Instead, such mechanisms are discussed in Section 5.

3.4.4.4 Indications

This section presents the feedback of the automation to the operator, which is facilitated by the indications. The information supply via this module is necessary to ensure situational awareness and guarantee automation transparency.

The pilot's situational awareness as to the state of reconfiguration is managed by the indication item in Section 2.4.3.2. Therefore, the proper function of the indications directly derives requirements on the automation. The latter needs to ensure that the different applicable color patterns of the indication item previously presented in Section 2.4.3.2 can unambiguously be generated. Here a short analysis as to how this is achieved is provided.

The computation of the necessary color coding is an algebraic function of the current automation state and the State Machine inputs. The exact logic for the pattern choice is found in Table 3.7. Each row of the table refers to a specific and unique color pattern. The color pattern is not in the scope of this thesis.

For the sake of simplicity, here the changes with relation to the causal chain of events during the transition and retransition are examined. From Table 3.7 it is visible that the relationships between automation data and indication patterns is unambiguous and purely algebraic.

Transition Causal Chain of Color patterns

The different indication color patterns during the transition are summarized in Table 3.8. In the second column of the table the indications, which are necessary for the operational

Table 3.7: High-Degree of Automation Transition Indication Item Truth Table

Description	sLTU	δ_T	sHS	LTU _{diseng^{timeout}}	retrans _{timeout}
In Hover, stay in Hover	<i>Engaged</i>	$\chi_{\mathbb{H}}(\delta_T)$	<i>false</i>	-	-
In Hover, leave Hover requested	<i>Engaged</i>	$\neg\chi_{\mathbb{H}}(\delta_T)$	<i>false</i>	-	-
In Transition, Hover requested	<i>Engaged</i>	$\chi_{\mathbb{H}}(\delta_T)$	<i>true</i>	-	-
In Transition, stay in Transition	<i>Engaged</i>	$\chi_{\mathbb{T}}(\delta_T)$	<i>true</i>	-	-
In Transition, Wingborne requested	<i>Engaged</i>	$\chi_{\mathbb{W}}(\delta_T)$	<i>true</i>	-	-
LTU disengagement, nominal	<i>Disengaging</i>	$\chi_{\mathbb{W}}(\delta_T)$	-	<i>false</i>	-
LTU disengagement, Retransition requested	<i>Disengaging</i>	$\neg\chi_{\mathbb{W}}(\delta_T)$	-	<i>false</i>	-
LTU disengagement, off-nominal, action necessary	<i>Disengaging</i>	-	-	<i>true</i>	-
In Wingborne, stay in Wingborne	<i>Disengaged</i>	$\chi_{\mathbb{W}}(\delta_T)$	-	-	-
In Wingborne, leave Wingborne requested	<i>Disengaged</i>	$\neg\chi_{\mathbb{W}}(\delta_T)$	-	-	-
LTU engagement, nominal	<i>Engaging</i>	$\neg\chi_{\mathbb{W}}(\delta_T)$	-	-	<i>false</i>
LTU engagement, Wingborne requested	<i>Engaging</i>	$\chi_{\mathbb{W}}(\delta_T)$	-	-	<i>false</i>
LTU engagement, off-nominal, action necessary	<i>Engaging</i>	-	-	-	<i>true</i>

procedures of Section 3.3 are signified. Note that there are several indication changes that are not directly related to the Functional Flow but are necessary for a better mode awareness of the crew.

Table 3.8: *Indication Item Causal Behavior during Transition. The Color Patterns Themselves are Not in the Scope of this Thesis and Can be Found in [9]*

Indication Data	Proc. Step
Entry Point	-
Transition Region Request	-
In Transition	-
In Transition, Wingborne Request	t2)
Start of Shutdown	t5)
End of Transition	t8)
Abnormal: Action Necessary	t8a)

The start is in the Hover region, which is depicted in the first row of Table 3.8. This would be the case when

$$s_{LTU} == Engaged \wedge \chi_{\mathbb{H}}(\delta_T) \wedge \neg s_{HS}. \quad (3.72)$$

Next, the operator deflects the control inceptor out of the hover region, i.e.

$$s_{LTU} == Engaged \wedge \neg \chi_{\mathbb{H}}(\delta_T) \wedge \neg s_{HS}, \quad (3.73)$$

which triggers a color pattern change, driven by row two of the table. This indication notifies the operator that the automated system is attempting to exit the hover flight phase as requested by the crew.

As the aircraft gains airspeed and crosses the threshold for the hover phase, dependent on the control inceptor deflection the color coding from either row three or row four are in effect. The former is applicable, if

$$s_{LTU} == Engaged \wedge \chi_{\mathbb{T}}(\delta_T) \wedge s_{HS}, \quad (3.74)$$

otherwise the latter would apply when

$$s_{LTU} == Engaged \wedge trans_{s_{rqst}} \wedge s_{HS}. \quad (3.75)$$

The latter is the precondition for the start of transition t2) as per Section 3.3.1.

As the aircraft gains airspeed, the conditions for the start of disengagement from Equation 3.19 are fulfilled, and therefore as per t5), the crew is notified with the color pattern of row five of Table 3.8. This is true when

$$s_{LTU} == Disengaging \wedge \neg LTU_{disengtimeout}, \quad (3.76)$$

where for recollection $LTU_{disengtimeout}$ is calculated as per Equation 3.44.

According to the functional breakdown from Section 3.3 the successful transition t8) is indicated with row six of the table. This would be the case when

$$s_{LTU} == Disengaged \wedge trans_{rqst}. \tag{3.77}$$

In the cases of a failure to disengage, i.e. t8a) of Section 3.3 the last color pattern is in effect. This is computed with

$$s_{LTU} == Disengaging \wedge LTU_{disengtimeout}. \tag{3.78}$$

This raises the awareness of the crew that mitigation procedures are in effect and that the automation is waiting on the operator input to proceed.

Retransition Causal Chain of Color patterns

Table 3.9 follows the color pattern changes during the retransition. The layout of this table is identical to that of Table 3.8.

Table 3.9: *Indication Item Causal Behavior during Retransition. The Color Patterns Themselves are Not in the Scope of This Thesis and Can be Found in [9]*

Indication Data	Proc. Step
Entry Point	-
Retransition Region Request	-
Start of Retransition	r3)
End of LTU Engagement	r6)
Abnormal: Action Necessary	r6a)
Hover Region Request	-
End of Retransition	r9)

Starting from the wingborne mode of operation, the entry point of the retransition procedure is equivalent to the end of the transition, i.e. the conditions of Equation 3.77 apply.

The request for the retransition is indicated by the color pattern in the second row of Table 3.9. This pattern is displayed when

$$s_{LTU} == Disengaged \wedge retrans_{rqst}, \tag{3.79}$$

notifying the crew that the request has been processed.

In accordance with Section 3.3.2, the start of the retransition is coupled to the engagement of the powered-lift system. In terms of indication, this is handled by r3) and the color pattern is denoted with the third line of the table. The pattern is set when

$$s_{LTU} == Engaging \wedge \neg retrans_{timeout}. \quad (3.80)$$

The success of the the LTU disengagement is communicated to the crew with row four of the table and this is directly coupled to step r6) of Section 3.3.2. The conditions for this color pattern were already introduced with 3.74.

Should step r6) fail as per r6a), then the color pattern for crew alert is visible in row five of Table 3.9. This occurs if

$$s_{LTU} == Engaging \wedge retrans_{timeout}. \quad (3.81)$$

This notifies that additional actions are required and that a mitigation procedure should be initiated.

The next two patterns - rows six and seven of Table 3.9 - are solely dependent on the kinematic velocity, the latter marking the entry to hover flight and therefore also the end of the retransition as per step r9) of Section 3.3.2.

The color pattern of row six would be applicable whenever

$$s_{LTU} == Engaged \wedge \chi_{\mathbb{H}}(\delta_T) \wedge s_{HS}, \quad (3.82)$$

whereas the conditions for row seven were already introduced with Equation 3.72.

Indications, dealing with High-Lift System Operation

With regard to the high-lift system operation, the state of M_{HL} is communicated in order to ensure awareness as to what the currently active scheduling is. Additionally, the extraction timeout $HL_{timeout}$ is passed as a warning, indicating that an functional issue has been detected with regards to the operation of the flaps.

3.5 Design Analysis

Having presented the design, this section explains how the proposed functions of Section 3.4 fulfill the set of objectives of Sections 3.3 and 3.1. Firstly, in Section 3.5.1 the process flow is studied and compliance with the desired high-level pilot-machine interaction and behavior of Section 3.3 is demonstrated. The analysis derives which states of the proposed Finite-State Automata M_{LTU} and M_{HL} are allocated to the different aircraft flight phases. Furthermore, the interactions of the different software modules are analyzed in this section, such as the scheduling of the high-lift system and the airspeed envelope protections. Finally, in Section 3.5.2 it is analytically demonstrated how the safety-objectives of Section 3.1 are fulfilled.

3.5.1 Integrated System Behavior

This section examines how the process flow of Section 3.3 fits into the proposed design. Similar to Section 3.3, this section begins with the analysis of the Transition in Section 3.5.1.1 and then with that of the Retransition in Section 3.5.1.2. The items of the process flow to events within the proposed design of Section 3.4 are linked.

Afterwards it is possible to allocate the Decision-Making states to the flight phases. This is performed in Section 3.5.1.4. In Section 3.5.1.5 further dependencies that cannot directly be linked to the process flow, but are necessary for the overall proper execution of the functions, are examined.

It should be noted that it is assumed that the crew executes the procedures as defined in Section 3.3. In Section 3.5.2 it is further demonstrated that deviations from the prescribed actions have no adverse effect on the system performance.

3.5.1.1 Transition

For the analysis of the transition procedure, Figure 3.2 is taken into account and it is demonstrated that the implementation follows the laid out process diagram. From Sections 3.4.1 and 3.4.2, it is visible that the initial conditions of the two Finite-State Automata are

$$s_{LTU} = Engaged \quad (3.83)$$

and

$$s_{HL} = Extend. \quad (3.84)$$

As seen in Table 3.5 of Section 3.4.4.2, this implies that no lower limit of the airspeed protections is set and the upper limit is V_{LSNE} .

For recollection in t1) of Section 3.3.1 the process starts with an operator request. This is registered by the software whenever $trans_{rqst}$ is evaluated to be *true* as per Equation 3.3.

From the control law specification found in Section 2.4.1.1, the aircraft eventually accelerates to an airspeed, which is higher than the disengagement speed $V_{disengage}$, which is calculated in Equation 3.51. This is visible when comparing Equations 3.51 and the definitions of the airspeed command mapping, found in Section 3.4.1.1.

During the aircraft acceleration, the operator is kept aware of the current state of process automation using the color patterns, which change accordingly due to the criteria from Equations 3.72 and 3.75. The latter is also reflected as t2) of Figure 3.3.

During this time, the check expressed in Equation 3.19 is running. This is the condition, which triggers the disengagement process. Therefore, the transition condition t_1 implements process step t3). Once the transition function of Equation 3.18 is executed, i.e. $s_{LTU} == Disengaging$, then the series of actions t4), t5) and t6) are performed simultaneously.

Taking the state change into account, from Table 3.5 it is visible that the automation sets the lower airspeed limit accordingly to ensure no stall can occur (t4). Subsequently, the indication t5) is driven from Equation 3.76. From Section 3.4.4.1 it follows that the physical shut down of the powered-lift system is executed by the control allocation (t6).

For now, it is assumed that no latent error in the powered-lift system is present, i.e. the disengagement completes successfully. The cases where a mitigation is necessary is covered later in Section 3.5.1.3. t7) is facilitated by the transition function in Equation 3.20. The transition of state s_{LTU} to *Disengaged* toggles the change of indication denoted in t8) as per Equation 3.77. The aircraft is fully wingborne and therefore the upper airspeed limit is released as visible from Table 3.6, which implements t9).

If a high-lift system is present, then the state transition of Equation 3.20 also triggers the change of high-lift system operation due to Equation 3.45. From this moment onward the flap deflection is scheduled over the airspeed to pursue an aerodynamically optimal configuration according to Section 3.4.4.3. Whenever this occurs and the movement of the flaps is out of the extended position, according to Table 3.5 the lower airspeed limit is set to V_{SAFE} .

3.5.1.2 Retransition

This section demonstrates that the design fulfills the procedure, depicted in Figure 3.3. The starting point from the perspective of the procedure is explained in Section 3.3.2 and in terms of automation, this would imply that

$$s_{LTU} = Disengaged \quad (3.85)$$

and

$$s_{HL} = Retract. \quad (3.86)$$

The high-lift system is scheduled as per Equation 3.68 to minimize the aircraft drag. The upper airspeed limit is scheduled according to the flap movement as per Table 3.6, whereas the lower airspeed limit is V_{SAFE} as visible from Table 3.5. Additionally, the state of indication is as per Equation 3.77.

According to r1) of Figure 3.3, the start of the retransition is initiated by the operator. The software registers this by means of $retrans_{rqst}$. By law design, the operator action induces an aircraft deceleration. At the same time, the color pattern of the indication changes because the logical relationship of Equation 3.76 holds. Thereby the operator receives feedback that the request has been processed correctly.

The first automation task is to adjust the scheduling of the high-lift system to extend. This is done whenever the airspeed is deemed low enough to mitigate structural damage. The condition for this is visible in Equation 3.48. Whenever the condition is fulfilled, the state of M_{HL} changes to *Extend*, which schedules the high-lift system to the highest setting possible as per the mapping of Equation 3.70. Thereby the drag is maximized, facilitating higher deceleration rates.

The check of r1) of Section 3.3.2 is implemented by the transition condition, found in Equation 3.23. This also initiates the motor engagement. Whenever the condition is *true*, i.e. the state of M_{LTU} changes to *Engaging*, the series of action r2) - r4) take place.

Firstly, the airspeed upper limit is readjusted to V_{LSNE} as seen from Table 3.6 (r2). The color pattern changes due to Equation 3.79 to facilitate r3). The operator is hence made aware that the powered-lift system activation process has commenced (r4). The turn on itself is executed by the control allocation as seen in Section 3.4.4.1.

Here it is assumed that there are no issues in the activation of the LTUs, implying that check r5) of Figure 3.3 is successful. The abnormal scenarios in that regard are covered in Section 3.5.1.3. The check r5) is implemented by the transition condition, found in Equation 3.25, which triggers the state transition of M_{LTU} to *Engaged*.

The indication that the aircraft is in powered-lift flight configuration is set due to Equation 3.74. This implements step r6). The state transition also releases the lower airspeed limit as visible from Table 3.5 (r7).

The entry into the hover region is up to the pilot. Whenever this is requested, i.e. $\delta_T \in \mathbb{H}$, the indication color pattern changes as per Equation 3.82 to indicate that the system correctly has processed the request. As the aircraft decelerates further, the check that is performed in step r8) is that of Equation 3.72 to indicate the entry to hover (r9).

3.5.1.3 Mitigation Strategies

Mitigation Strategies during Transition

The system response in the cases where during transition one or several LTUs are incapable of disengaging is analyzed first. Hence, the analysis of Section 3.5.1.1 continues assuming that t7), implemented by the transition condition in Equation 3.21, is never true.

Instead, t7a) of Section 3.3.1 has to apply by means of the elapsed timeout. It is implemented by the design solution with Equation 3.44. Whenever check t7a) is applicable, the operator is alerted that actions from their part are necessary as per t8a), which is supplied with the indications via Equation 3.78. This marks the starting point of the transition mitigation process of Figure 3.4. For recollection, the available mitigation at disposals is be to enter hover flight or to attempt to enter wingborne flight.

For the first mitigation - reverting to powered-lift flight - the request of tm2a) from the operator is registered by the automation via the transition condition of Equation 3.27. This causes the state s_{LTU} to change to *Engaging*, which starts the retransition, as already explained in Section 3.3.3. In this case, however, r1) is instead executed by the above-mentioned transition condition of Equation 3.27. The airspeed adjustment of r2) does not play a role, as the scheduling of t4) is equivalent as visible from Tables 3.6 and 3.5.

The second possible mitigation is to proceed to wingborne flight. It must be noted that whether the type of LTU fault can occur is a question of the design of the powered-lift units and the robustness of the failure-detection and isolation mechanisms. This was discussed

in Section 2.3.1.2. For example, erroneous non-zero RPM may always be detectable and subsequently trigger an automatic power supply cut-off of the unit. Even if an erroneous non-zero RPM is possible, the necessary crew actions is the manual deactivation of the problematic LTU (tm2b), which may not be technically feasible.

In this scenario the check of t7) is running continuously. Recall that t7) is implemented by Equation 3.20 and ensures the deactivation of all LTUs. In the cases where by some means the LTU in question manages to shut down, wingborne mode will be entered by design. Therefore, whether the power is removed automatically by the fault isolation mechanism or manually by the crew or not at all has no implication of the structure.

From the perspective of the procedures, an impossibility to deactivate a given LTU implies that only one mitigation is available and namely to revert to powered-lift flight. This, however, can be executed much later and in the mean-time greater distances can be covered by the aircraft due to the quasi-wingborne flight⁴. During this time the upper airspeed limitation mitigates structural damage as evident from Table 3.6.

Mitigation Strategies during Retransition

When performing the retransition, check r5) of Section 3.3.2 must fail to complete, i.e. an LTU cannot engage, implying that the function of Equation 3.24 is not performed.

Instead, what occurs is that step r5a) is in effect. This check in the process is evaluated with Equation 3.81, which in turn enables the color pattern change that implements step r6a). This informs the crew of the necessary actions and formally signifies the start of the retransition mitigation strategies, found in Section 3.3.4.

As seen in Figure 3.5, the available mitigation options are to proceed with powered-lift flight regardless of the fault of the LTU or to revert back to wingborne flight.

The motivation as to why it is reasonable to require crew confirmation prior to entering powered-lift flight was argued when introducing the procedure in Section 3.3.4. The decision of the crew to do so is communicated to the software by means of the processed input symbol $LTU_{override_{rqt}}$. This variable goes into the condition found in Equation 3.31, which for recollection, drives the state of M_{LTU} to *Engaged*. Recall from Section 3.5.1.2 that this state change causes step r5) of Section 3.3.2 to be evaluated as *true*, similar to Equation 3.25. This in turn continues the retransition from r6) onward.

In order to revert back to wingborne flight, the operator requests the transition procedure in accordance with rm2b) to begin as per Equation 3.29, changing the state of M_{LTU} to *Disengaging*. Similarly to the flow, found in Section 3.5.1.3, t1) and t3) are fulfilled automatically by design. As discussed in that section, the airspeed adjustment of t4) is equivalent to that of r2) so that no change in the limits occurs. The remainder of the procedure was already explained in Section 3.5.1.1.

⁴“Quasi-wingborne” flight here means that the aircraft may continue flying at high airspeed while not utilizing the powered-lift system. Therefore, the control algorithms are in their wingborne mode. However, one or more LTUs are still rotating, which by definition implies that wingborne flight has not yet occurred. Via the actions in Equation 3.50, the flaps can be set to increase the aerodynamic efficiency.

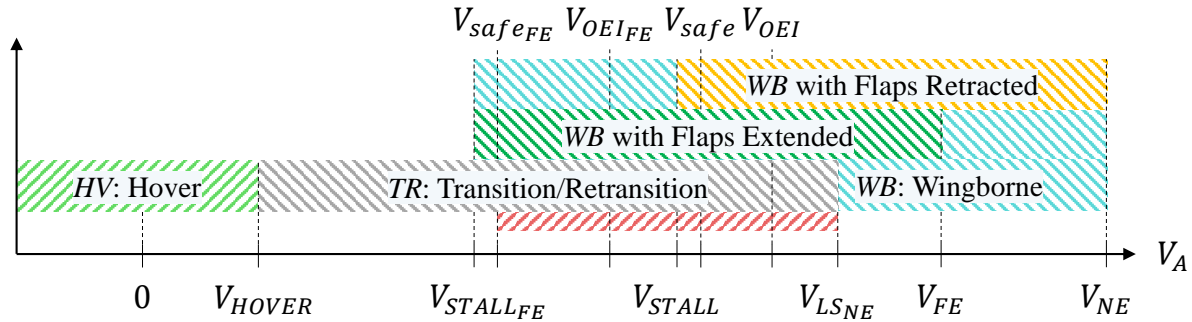


Figure 3.8: *Lift to Cruise Aircraft Flight Phase Allocation with Relation to the Airspeed. This Figure Supplements Figure 2.10 with All Airspeed Values, Used by the Automation. The Region Sizes are Chosen for Better Visibility and Need Not be to Scale.*

3.5.1.4 State Allocation to Flight Phases

The relationship between state-space of automation and the aircraft flight phases was indirectly discussed in the previous sections. For the sake of completeness, these properties are addressed here.

Figure 3.8 supplements the flight phase allocation with relation to the airspeed, defined with Section 2.4.2. The airspeed values that are relevant for the disengagement and engagement process are included in the figure. They play a part in the choice of LTU disengagement engagement speeds, found in Sections 3.4.1.1 and 3.4.3.2, the high-lift system deployment and retraction velocities of Section 3.4.2.1 and the airspeed scheduling from Section 3.4.4.2. Additionally, the region with red marking in Figure 3.8 is no longer the theoretically permissible LTU engagement and disengagement region, but the actual one.

With the use of the figure together with the summary of the system behavior discussions from the previous sections, the states of M_{LTU} and M_{HL} can be assigned to the aircraft flight phases. The findings are included in Table 3.10 and also denoted with green in the State Machine graphical representations, found in Figures 3.6 and 3.7.

The states of M_{LTU} that mark the end stage of the transition to wingborne flight and the beginning stage of the retransition to powered-lift flight - *Disengaging* and *Engaging* respectively are examined first. Those apply in the region, denoted with red in Figure 3.8. Therefore, they can be allocated to *TR*. From the figure it is visible that these states are applicable in the last portion of *TR* with regards to the airspeed. Thus, the remainder of the powered-lift flight must be performed in the state *Engaged*. Hence, this state is allocated to both *HV* and *TR*. By definition, wingborne flight *WB* requires the state *Disengaged*.

The scheduling of the high-lift system to decrease the drag as per Equation 3.68 is done whenever s_{HL} is *Retract*. This is nominally done in the wingborne phase and abnormally if the disengagement is not correct as per Equation 3.50, hence the allocation in Table

Table 3.10: *State Machine State to Flight Phase Allocation*

State Machine	State	Flight Phase		
		HV	TR	WB
M_{LTU}	<i>Engaged</i>	✓	✓	
	<i>Disengaging</i>		✓	
	<i>Engaging</i>		✓	
	<i>Disengaged</i>			✓
M_{HL}	<i>Extend</i>	✓	✓	✓
	<i>Retract</i>		✓	✓

3.10. Prior to the entry of TR , i.e. when s_{LTU} is *Engaging* the full High-Lift deployment is expected. Because of this, *Extend* has to occur in WB . Therefore, *Extend* applies to HV and TR and may be applicable in WB .

3.5.1.5 Additional Interactions

This section summarizes necessary properties that are considered in the design that do not have an immediate contribution on the suggested procedures.

Disengagement Speed and Underspeed Protection

The first property is the interaction between engagement and disengagement speeds and the airspeed protection scheduling. In fact, the region of the LTU engagement and disengagement occurs in the airspeed envelope, denoted with red in Figure 3.8.

The exact speed, above which the disengagement may occur is a function of the aircraft configuration as per Equation 3.53. It must be noted that the definition of the disengagement speed is consistent with the lower airspeed limit scheduling for non-engaged LTUs, found in Table 3.5. This implies that the airspeed protections begin enforcing the lower end of the of the mentioned envelope in the moment when the state s_{LTU} transitions to *Engaging*. Additionally, the aircraft is by design flying at an airspeed which is at least the same as the one, defining the lower end of the envelope. It is therefore ensured that there is no design cause, in which the aircraft needs to be automatically accelerated as it is beneath the lower airspeed limit.

One other interaction is in the events of failures in the traction system. The instance where this is relevant is again in the choice of envelope, in which the disengagement of the LTUs takes place, which manifests with the calculation of the disengagement speed as per Equation 3.53. The failure is also considered in the underspeed protection, found in Table 3.5.

Engagement Speed and Overspeed Protection

Similar considerations are made when executing the LTU engagement. The engagement speed definition of Equation 3.10 is consistent with the upper airspeed limit of Table 3.6 when s_{LTU} is *Engaging*. This means that, on the one hand, the envelope is enforced and, on the other hand, does not cause an unwanted automatic deceleration due to the design.

By definition during $s_{LTU} = \textit{Disengaging}$ or $s_{LTU} = \textit{Engaging}$ the LTUs are neither considered engaged nor disengaged. Therefore, the upper and lower airspeed limits exactly coincide with the theoretical engagement and disengagement envelope that is denoted in red in Figure 3.8.

Coordination between the two State Machines

The next interaction has to do with the automation itself. The coordination between the two Finite-State Automata M_{LTU} and M_{HL} in the cases of failures is examined. In terms of order of execution, the retraction of the high-lift system is designed to follow a disengagement of the powered-lift system. Similarly, the engagement of the powered-lift system is set to follow the extension of the high-lift system.

In terms of design, the management of the LTUs considers a possible failure of the high-lift system that causes it unable of extending fully as per Section 3.4.3. This is necessary in order to guarantee liveness of the retransition, otherwise a failure to extend renders the software incapable of engaging the LTUs. The way this is achieved is visible in Equations 3.54 and 3.55.

M_{LTU} considers an impossibility for the flaps to extend fully in the calculation of the disengagement speed as per Equation 3.53. In the abnormal scenario where they are not extended, the automation would initiate the LTU disengagement at a higher airspeed.

From Section 3.4.2 it can be observed that the management of the high-lift system accounts for the impossibility to disengage the LTUs fully. As already mentioned in Section 3.5.1.3, the crew may want to fly for a prolonged duration of time in this configuration state. In order to reduce the drag and improve the efficiency further, the crew can change the high-lift system scheduling to the aerodynamically optimal setting of Equation 3.68 by the transition condition of Equation 3.50.

On the other hand, M_{HL} does not need to account for an impossibility to engage an LTU. As mentioned in Section 3.5.1.4, the operation of the high-lift system is only in the phases where the LTUs are either in the process of disengaging or fully disengaged. Otherwise, they should be fully retracted, regardless of the type of failure in the powered-lift system.

3.5.2 What-If Analysis

In previous sections, the behavior of the system during the transition and retransition and how the design facilitates the process flow of Section 3.3 was examined. This section analyzes the behavior of the design in off-nominal scenarios that are not covered by the mitigation strategies. This section is structured as follows.

Previously, it was always assumed that the crew executes the procedures exactly as prescribed. Section 3.5.2.1 examines how the automation reacts if the crew actions deviate from the specification, thereby demonstrating that false crew actions cannot cause an adverse situation.

Afterwards the automated response in the event of faults of different surrounding components is analyzed. The major findings are covered in Section 3.5.2.2. There the effects of failures in the hover propulsion system, the high-lift system, traction system and different sources of sensor information are summarized.

3.5.2.1 Procedural Deviations

In order check the system response for diverse procedural deviations by the crew, the different types of actions deemed as “deviations” must be classified. They are summarized in the following three categories:

- Requesting a reconfiguration and withdrawing it prior to the start of reconfiguration.
- Requesting a reconfiguration and withdrawing it after the start of reconfiguration but at times where no action is expected with relation to the procedures.
- Executing a mitigation strategy before required.

Deviations during Transition

As per Section 3.5.1.1, the reconfiguration starts with step t3) of Figure 3.2, i.e. when s_{LTU} is *Disengaging*. Withdrawing the transition request prior to this triggers the following events. The way the transition request is communicated, the flight control algorithm would initiate a deceleration of the aircraft. The automation, on the other hand, would also not conduct the disengagement, as one of the conditions for t3) as per Equation 3.19 is namely that request which is now withdrawn. Therefore, no reconfiguration happens and the procedure is replaced with no adverse effects.

The next scenario is when the reconfiguration request to wingborne flight is withdrawn after initiation of the disengagement (i.e. after t3) but before the transition is either successful or a mitigation is in effect (t7) and t7a) respectively). From Section 3.5.1.1 it is evident that in this case the automation state s_{LTU} is *Disengaging*. From the transition condition in this mode as per Equation 3.27 this operator action initiates the motor engagement. Thus, the disengagement process is interrupted by an engagement process.

The last operator deviation to examine is the premature initiation of a mitigation strategy. The transition mitigation strategy to powered-lift flight as per 3.3.3 is equivalent to withdrawing the transition request. Therefore, this scenario is equivalent to the one of the previously discussed. Another premature action is to request a flap retraction prior to disengagement. From Equation 3.50 it is visible that this operator action will not have an immediate effect on the system. It is only accepted by the automation after $LTU_{disengtimeout}$, which by design of Equation 3.78 formally marks the start of mitigation strategies.

Deviations during Retransition

Similar to the last paragraph, the response of the software during retransition for unanticipated crew input is examined. Firstly, a withdrawal of the retransition request prior to the start of motor engagement is studied. In accordance with Figure 3.3 this is prior to step r1) or prior to the condition of Equation 3.23 evaluating as *true*. Here it must be differentiated whether the withdrawal occurs prior to initiating the high-lift system deployment or not.

Recall that the deployment of the high-lift system only starts when below a given airspeed as seen from Equation 3.48. Therefore, if the deployment of the high-lift system has not been initiated yet, the withdrawal has no effect on the automation. In the cases where the high-lift system is already extending, i.e. $s_{HL} = Extend$, then withdrawing the retransition request also withdraws the extension command. Thereby, the high-lift system resumes the scheduling, such that the drag is minimized.

Suppose the withdrawal occurs after the system has commenced the motor engagement. Then, in accordance with Equation 3.29, this withdrawal terminates the motor turn-on process and instead initiates a motor disengagement. This is equivalent to a premature execution of the mitigation strategy to wingborne flight. The automation need not differentiate between the two cases.

In the scenario where the operator sends the hover confirmation as per Figure 3.5 prior to it being necessary, from Equation 3.31 it is visible that this has no impact on the system. This is because the operator input is processed only after the built-in timeout $retrans_{timeout}$. This means that either the retransition is successful prior to the elapsed timeout or the timeout marks the start of the mitigation, where the operator action is accepted as the mitigation strategy.

3.5.2.2 Reaction to Faults

A study of the automated response in the event of faults of different components was performed and can be found in Appendix C. The findings are summarized here.

Should a hover propulsion unit fail arbitrarily during the different automation phases, two classes of failures are relevant. The first one of the failures is the unit failing completely and thereby not rotating. The other manifests in a non-zero RPM output regardless of the supplied RPM command.

An LTU failure to zero RPM has no impact on the transition automation. A failure along the envelope leads to a loss in performance and produces an initial transient. However, the control algorithm of FSD-SVO provides sufficient disturbance rejection and handling qualities in the presence of faults. During transition a shutdown of all propulsion units is necessary in any case. In the automation it must be considered that the thrust distribution of the remaining LTUs could be different than in the nominal case, which is covered by Equation 3.16.

A failure of an LTU producing non-zero RPM in the transition prior to s_{LTU} being *Disengaged* implicates that a mitigation strategy will be in effect as soon as the condition of Equation 3.78 applies.

Should such a failure occur when s_{LTU} is *Disengaged*, i.e. in wingborne flight, then there is no change in the states of the automata. Instead, in accordance with Table 3.6, the upper airspeed limit is capped to $V_{LS_{NE}}$. This means that the aircraft is forced to decelerate if the airspeed is above that value so as to mitigate possible structural damage.

In the retransition, both propulsion unit failures prior to s_{LTU} becoming *Engaged* implicates a mitigation strategy execution, whereas a failure during that state means no change in the automation.

From Sections 3.4.2 and 3.4.4.3 it is known that during transition the high-lift system should fully be deployed. Assuming a fault leads to the flaps not being fully extended, then the following events occur. Firstly, the disengagement speed $V_{disengage}$ is increased as per Equation 3.53. In addition, after the disengagement has initiated, the lower airspeed limit is also raised in comparison to the one in the nominal case as per Table 3.5.

Should the high-lift system become stuck or experience a hardover in wingborne flight, then the upper airspeed limit adapts according to the feedback as seen in Table 3.6 to mitigate structural damage. This implies that if the aircraft speed is high as the failure occurs, the algorithms initiate a deceleration in an attempt to save the aircraft.

In the scope of the retransition, when prior to engagement of the LTUs, the high-lift system should be fully deployed as per the scheduling of Equation 3.70. If a fault occurs, such that the deployment cannot succeed, the turn-on of the LTUs commences that regardless of the type of fault. This is due to Equations 3.54 and 3.55.

One other malfunction that requires attention is a fault in the traction system. Should this occur, then this has an impact on the disengagement speed $V_{disengage}$ and on the lower airspeed limit as seen in Equations 3.9 and Table 3.5 respectively.

Lastly, certain conditions within the automation module's decision-making process rely on operator input via dedicated channels. They are used solely in the instances of abnormal conditions such as the retraction of the high-lift system despite a failure do

disengage all LTUs or the entry to powered-lift flight despite an LTU failure and thus utilized in Equations 3.50 and 3.31. Therefore, a failure of these input items would render the mitigation strategies impossible to execute. However, the necessity to utilize these inputs implies also a failure in the powered-lift system and therefore the malfunction of the input items would equate to at least to simultaneous errors. Such considerations are hence out of scope in this thesis. However, it is advisable to mitigate latent errors via a Built-In Test (BIT) [124] prior to operating the aircraft to further reduce hazards.

Robustness against failures in other peripheral components was not demonstrated. For example, on many occasions knowledge of the airspeed or the deflection of the control inceptor in the decision-making process is required. Hence, arbitrary failures in the airdata will indeed have an adverse impact on the integrated system behavior.

Section 3.2.1 assumes that such events are detected by the signal integrity checking, meaning that the errors are known by the automation. The sensor information that was not covered in the analysis are required by the control law of FSD-SVO. Therefore, additional robustness measures are not necessary. Should such failures occur, then a takeover by a lower-automation algorithm is required regardless of the considerations in the automation design, as closed-loop stability cannot be ensured anymore. The lower-degree of automation algorithm responsible for flight in the event of such failures is covered in the next chapter.

3.6 Chapter Summary

This chapter presented the high-degree of automation procedures description for the transition from powered-lift flight to wingborne flight and back of VTOL aircraft. In addition, an automation design was derived that can execute the defined procedures in accordance with the crew actions. The proposed procedure and automation advance the state of technology in accordance with **Contribution 1**. It accomplished the following targets.

Compliance with the FSD-SVO

The suggested strategy complies with the FSD-SVO concept fully. In the nominal case, the procedures of reconfiguration from powered-lift to wingborne flight and back is fully automatic and requires no manual reconfiguration. This was evident in Section 3.3.

In addition, the automation fulfills the requirements set out from the FSD-SVO. During reconfiguration, it provides the law with the necessary information for correct execution. This is visible, on the one hand, from Section 3.4.4.1 where the algorithm is supplied with the commands in accordance with the state of reconfiguration. On the other hand, a clear distinction of the flight mode can be traced in Section 3.5.1.4.

Extension of the FSD-SVO

The high-degree of automation proposal extends the FSD-SVO with regards to robustness in both nominal and off-nominal scenarios. In Sections 3.1 and 3.2.1 the necessary airspeed protection scheduling was demonstrated. It is integrated into the automation design in 3.4.4.2. During reconfiguration with and without faults no potentially hazardous situation can be entered. A safe state is also enforced by the automation and airspeed scheduling.

The FSD-SVO was further extended to account for the utilization of a high-lift system. This was done in Sections 3.4.2 and 3.4.4.3. If a high-lift system is installed, the core State Machine responsible for reconfiguration from powered-lift to wingborne flight and back was designed modular and needs only to be supplemented by the proposal of Section 3.4.3.

Operator Support

The human operator is considered in every aspect of the procedure design. From Section 3.3 it is evident that the tasks, allocated to the operator, are simplistic and clear. In addition, the automation design accounts for deviations in the procedures as evident in Section 3.5.2.1 and never allows for the entry of a potentially hazardous envelope.

The state of automation is intuitive and easy to track by the crew. This statement is supported by the content of Section 3.4.4.4. The design can provide the indication items with all necessary information. The items are tailored to supply the crew with the aircraft flight state and that flight state can clearly be linked to the automation as done in Section 3.5.1.4. In abnormal scenarios the high-degree of automation produces a very limited set of possible actions from the crew. During the decision-making process again a safe system state is enforced, hence the decision-making is not time critical.

Lastly, the control inceptor concept of the FSD-SVO is considered by the design fully for flight mode selection. The required additional input is necessary for abnormal scenarios and is kept to a minimum as seen in Sections 3.4.1.1 and 3.4.2.1.

Chapter 4

Manual Transition and Retransition and the Industry Standard Compliance

FSD-SVO and the automation of Chapter 3 produce a highly-automated operational concept that reduces the operator workload. In this thesis the integration of control concept and automation is referred to as the “Nominal” system. At first glance, minimizing the human involvement and thereby reducing the possibility for man-made errors with the Nominal system appears very desirable. Even though safety and reliability of the methods and algorithms may be at the forefront of the development, the fact is that lift-to-cruise aircraft configurations are novel. As a consequence, the possibility of undesirable effects in the Transition and Retransition flight phases cannot be discounted. Because such properties may implicitly be unknown, they are not unaccounted for by the Nominal system.

Until the Nominal system is rigorously flight proven, the likelihood of design errors during Nominal system flight are a hazard that must be addressed. In the cases of a Nominal system error, an additional control concept is required. This control concept must be dissimilar and allow for an increase in operator authority as a means to reduce the possibility of the same design errors to persist. The system that can facilitate an operational concept, where a more active operator participation is expected, is referred to as the “Fallback” system. It is composed of the flight control law and its automation. The Fallback system enables an SVO1 concept. The composition of Nominal and Fallback systems thus create a fail-active FCS that can counteract a critical failure of the high-degree of automation control concept.

This chapter presents the logical decision-making process of the Fallback system that enables the reconfiguration from powered-lift to wingborne flight and back. It tackles the reconfiguration concept in the scenarios where a reversion from Nominal to Fallback is necessary, ensuring correct takeover conditions. Considerations are provided as to how both approaches - Nominal and Fallback - can be integrated into the aircraft operational

procedures from two perspectives. The first is with regards to the operator awareness. The derived methods must guarantee consistent behavior from the perspective of the operator regardless of the system in command. In addition, a smooth takeover to the Fallback system must be ensured at all times during the reconfiguration. Secondly, the integration of the reconfiguration scheme with respect to the industry standards is critical for the certification of the aircraft and must be examined.

The content of this chapter is structured as follows. Firstly, Section 4.1 analyses the different challenges that the automation concept needs to consider and directly address. If the goal of the Nominal system automation from Chapter 3 is to provide an intuitive and simplified operation that enforces a safe aircraft flight envelope, the objectives of the Fallback system are severely different. Used only in the event of adverse conditions, the focus of the Fallback automation design is to ensure maximum operator authority. At the same time it must be tailored in a way that its presence does not impose limitation on the Nominal concept. In addition, it needs to be coherent to Nominal system procedures and interaction concept. Section 4.1 provides an overview of the resulting Fallback system automation requirements.

Section 4.1 presents the problems that this chapter must address. Section 4.2 introduces the input processing and presents the State Machine design that can achieve the required properties. It discusses the actions, performed by the system depending on the state of the automaton. This includes the informational supply to the operator, to the aircraft effectors and also to the surrounding software components that are part of the control algorithm. In the section the manual utilization of a high-lift system is also shown.

The same section introduces the methods that solve the challenges imposed but does not elaborate the exact mechanisms as to how the issues are addressed. Instead, this is performed in the next two sections.

Firstly, the behavior of the Fallback as a stand-alone system is examined in Section 4.3. The section focuses on how a transition from powered-lift to wingborne flight and back can be performed. From there, transition and retransition procedures with the Fallback system are derived. In addition, Section 4.3 provides a discussion of the allocation of the State Machine states to flight phases and an analysis of the system reaction to different faults and operator mistakes.

Section 4.3 examines the system behavior of the Fallback in order to gain deeper understanding as to its operation. The chapter continues with Section 4.4, in which the behavior of the whole FCS is studied. The FCS is composed of both Nominal and Fallback systems and requires dedicated analyses. In the section, the input processing of both systems is completed. The derivations allow to harmonize the transition and retransition procedures with the two systems, which is demonstrated in that section. It also provides an analysis of the FCS behavior correctness following a takeover to the Fallback system. How the transition and retransition procedures can fit into the mission profile that is

imposed by the regulatory organs is examined. Lastly, the chapter is concluded with Section 4.5. The contribution beyond the state of technology as per **Contribution 2** is demonstrated with a summary of the chapter contents.

4.1 Problem Description

In order to specify the automation requirements, the context under which the Fallback system is intended to function must be understood. This section provides an analysis of the surroundings of the automation execution and thereby derives the requirements it must satisfy.

4.1.1 Flight in the Presence of Faults

The motivation for such a strategy is the Fallback from the high-degree of automation control laws. The dependence on sensor information of the Nominal system is high. Thus, apart from the occurrence of a design error, it must be taken into account that a loss of certain sensor information may be the reason for activation of the Fallback concept. This implies that both law and automation must be capable of operating with as little sensor feedback as possible.

4.1.2 Facilitate Maximum Operator Authority

The occurrence of a design error of the Nominal system may implicate that effects within the flight envelope have not been considered or were unknown during the development. These could for example be aerodynamic effects during transition and retransition. In these instances reverting to the operator is the only feasible option to return to safe conditions.

Should such a scenario occur, then the automation must be designed in a way, in which the operator authority is guaranteed throughout the flight envelope. This is necessary in order to mitigate the possibility of the same design error also for the Fallback. This in turn means that the reconfiguration during transition and retransition must be performed manually and envelope protections must be avoided. This must be facilitated by the automation.

4.1.3 Simplicity

Decisions shall be taken only with reliable sensor information and the amount of information types shall be minimized. The number of Finite-State Automata and the interactions between them shall be kept low. Keeping the automation functions simple reduces the possibility of design errors. It also allows for more efficient testing. The simplicity of the automation is enabled by the increased operator authority in the Fallback concept.

4.1.4 Adequate Operator Feedback and Reaction Times

Despite that the operator involvement is increased, there is no guarantee that the crew has awareness as to the state of reconfiguration. Loss of awareness may be experienced during a takeover from the Nominal system. Similar to the Nominal automation, the Fallback automation must provide the necessary data for appropriate pilot indications.

Another property that has to be taken into account is the increase in the crew workload due to the higher demand of primary the flight control and envelope adherence. Therefore, the automation must additionally have provisions to account that the decision-making process of the crew may be delayed. Consequently, the resulting automation concept must not require immediate operator actions.

4.1.5 Law Functional Decomposition

Section 2.4.1.2 provided explanation that the Fallback law is divided into two distinct control concepts, referred to as “hover mode” and “wingborne mode”. As such, the automation proposal must account for this property and provide a manual reconfiguration procedure between the two control concepts. To minimize system complexity and satisfy automation transparency, the proposed automation at the same time must be the procedure for reconfiguration from powered-lift to wingborne flight and back.

4.1.6 Harmonization of the Transition and Retransition Procedures

One important characteristic is that the Nominal system is the intended system that is to be flown during flight. As such, the whole operation of the aircraft is tailored towards the usage of the Nominal system. This introduces the following two requirements on the Fallback system automation.

Firstly, it must be ensured that all previously presented properties from Section 2.4.1.2 and from Chapter 3 can be facilitated. This implies that the interpretation of the control inceptors must be equivalent for the Fallback automation. The Fallback system may require additional actions and considerations. However, it must guarantee that its existence does not negatively influence the operation of the Nominal system. Secondly, the highly-automated transition and retransition procedure of Chapter 3 and the manual procedure of this chapter must be harmonized. A smooth Fallback takeover must be ensured. This is necessary in order to ensure a consistent operation for the crew in both Nominal and Fallback systems.

4.1.7 Compliance with Industry Standards

Lastly, the applicable requirements available from the industry must be considered in the design to guarantee that the flight control system can be certified. An analysis of the placement of the transition and retransition in the segments that are imposed by the regulatory organs must be performed. It must be demonstrated that both Nominal and Fallback and thus the whole FCS functional design comply with the set requirements. In addition, compliance with the regulatory requirements must be shown in the events where the Fallback takes over during the reconfiguration process.

4.2 Automation Design

The previous section described the properties that the Fallback system must satisfy. In this section the focus is on the design solution that can enable the pursued qualities. The high-degree of automation of the Nominal system presented in Chapter 3 is centered around the hover propulsion system. In contrast, the choice of abstraction for the Fallback automation system is the functional composition of the control laws. The main reasons for this are discussed below.

Firstly, to ensure maximum operator authority, the pilot is actively required to select the mode of operation of the law - “hover” or “wingborne”. Due to this, the automation is human-centered, as it is built around the operator decisions. It must be noted that such an automation strategy is only possible due to this increased pilot authority. Namely, during flight with the Fallback system, the correct entry into the different flight phases and the envelope adherence are the task of the crew. This contrasts the properties of the solution in Chapter 3, where the flight envelopes are maintained by the Nominal system. As seen later in this chapter, operator support as to the choice of Fallback law is provided by the procedure design.

Secondly, by severely altering the system abstraction from the one of Chapter 3, a certain degree of design independence is achieved. This reduces the danger of common mode error due to design flaws by introducing a higher level of dissimilarity. As seen in this section, the structure of the two approaches is different even though in the end both Nominal and Fallback automation have the same goal - to schedule the operation of the hover propulsion and actuation systems.

With this in mind, let M_{FB} be the multi-level Finite-State Machine used to automate the Fallback flight control algorithms with relation to the reconfiguration from powered-lift to wingborne flight and back. M_{FB} contains two levels and is depicted in Figure 4.1.

M_{FB} 's first level state - $s_{FB|1}$ - belongs to the set of states

$$S_{FB|1} = \{Hover, Wingborne\}. \quad (4.1)$$

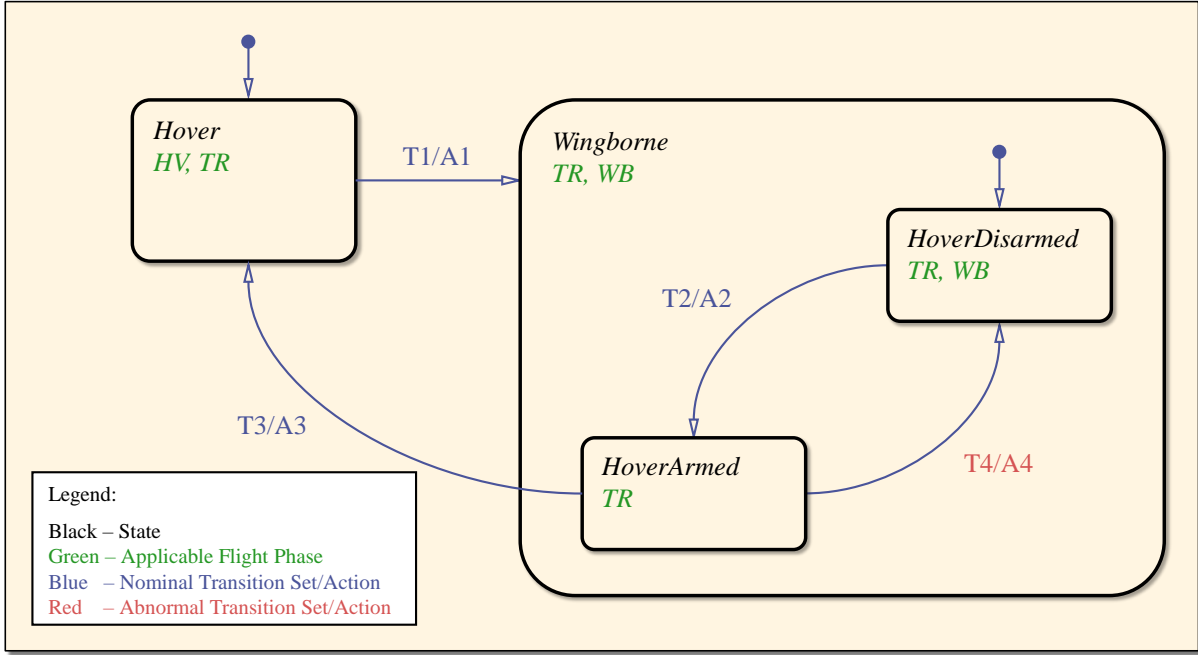


Figure 4.1: State Machine for the Fallback System Control Mode Selection

The second level of the automata - *Wingborne* - is from the set

$$S_{FB|2} = \{unused, \textit{HoverArmed}, \textit{HoverDisarmed}\}. \quad (4.2)$$

From Figure 4.1 the compatibility between the members of the sets $S_{FB|1}$ and $S_{FB|2}$ can be determined. For the sake of completeness, the information is summarized in Table 4.1.

Table 4.1: State Compatibility Matrix of M_{FB}

		$S_{FB 2}$		
		<i>unused</i>	<i>HoverDisarmed</i>	<i>HoverArmed</i>
$S_{FB 1}$	<i>Hover</i>	✓		
	<i>Wingborne</i>		✓	✓

The full definition of M_{FB} requires the specification of the initial state. However, one requirement set on the Fallback system is to be capable of an adequate behavior following a takeover. For the sake of transparency, the starting states of Figure 4.1 are chosen as

$$s_{FB|1_0} = \textit{Hover} \text{ and} \quad (4.3)$$

$$s_{FB|2_0} = \textit{unused}, \quad (4.4)$$

which would be equivalent to initializing in powered-lift flight on the ground. In reality, the starting states vary depending on the flight state during the takeover. As a consequence, the initialization must consider the possibility of different starting states. How this is achieved is explained later in Section 4.2.3, where the *Initialize* function and its applicability are explained.

The remainder of this section is organized as follows. Firstly, the section provides an overview of the selected State Machine, its states, allowed state transition and inputs. As seen later, the automaton is composed of two levels, therefore the choice of State Machine architecture is discussed.

Section 4.2.1 presents the automation’s Decision-Atomics. The input symbols of the State Machine are examined and the processing of the automation inputs that generate the symbols is derived. This is followed by Section 4.2.2, in which the transition conditions of the State Machine are introduced. There, the automation actions are discussed shortly.

In order to facilitate an adequate takeover, initialization of the State Machine is necessary. How this initialization is achieved is discussed in detail in Section 4.2.3. Lastly, the Decision-Execution of the Fallback automation is presented in Section 4.2.4. The communicated commands and requests to the surrounding systems are explained. Those include the law and control allocation actions, the method of high-lift system utilization by the operator and the information supply to the crew in terms of cockpit indications.

4.2.1 Decision-Atomics

Table 4.2 summarizes the input alphabet of M_{LTU} and the meaning behind the input symbols. The underlying physical entities that they are processed from are also mentioned. All of the members of the input alphabet belong to the boolean set \mathbb{B} . They are provided by the Decision-Atomics of the automation module, elaborated upon here.

Table 4.2: *Summary of the M_{LTU} Inputs*

Symbol	Interpretation	Signal Origin
$wingborne_{rqst}$	The operator requires fixed-wing control mode.	Human-Machine-Interface
$hover_{rqst}$	The operator requires the powered-lift control mode.	Human-Machine-Interface
arm_{rqst}	The operator requires to arm the powered-lift control mode.	Human-Machine-Interface
$disarm_{rqst}$	The operator requires to disarm the powered-lift control mode.	Human-Machine-Interface

As visible from Table 4.2, the input symbols of the Decision-Making process originate solely from the Human-Machine-Interface. It therefore follows that all state transitions are solely driven by the operator actions. The lack of restrictions by other sensor information demonstrates the full pilot authority with regards to the selection of operational modes. Furthermore, the robust automation design due to the high availability and reliability of the HMI sensors is ensured.

To derive the symbols in Table 4.2, the relationship

$$trans_{rqst} = \chi_{\mathbb{W}}(\delta_T) \quad (4.5)$$

is introduced. The division of the control inceptor was explained previously in Section 2.4.3.1. The boolean expression is the same as Equation 3.3 found in the high-degree of automation Decision-Atomics in Chapter 3. It indicates the similarity between the transition and retransition procedures. This similarity is discussed later in Section 4.4.

The first input symbol - $wingborne_{rqst}$ - is later used for the Fallback system reconfiguration to wingborne flight. It is calculated as

$$wingborne_{rqst} = trans_{rqst} \wedge shutdown_{rqst}. \quad (4.6)$$

The request to engage the Fallback wingborne law requires the discrete operator action $shutdown_{rqst}$. However, the request is only accepted for particular control inceptor setting, dictated via $trans_{rqst}$. This limits the possibility of inadvertent activation of wingborne mode. The choice of $shutdown_{rqst}$ is dependent on the Nominal and Fallback procedure harmonization and is hence explained later in Section 4.4. For the current discussions, it is assumed to be a known input.

The next two symbols - arm_{rqst} and $disarm_{rqst}$ - are used during the reconfiguration to powered-lift flight. The former is used in nominal conditions and engages the LTUs on operator request. The latter is used to revert back to wingborne flight in the event of abnormal events. They are calculated as

$$arm_{rqst} = \chi_{\mathbb{R}}(\delta_T) \wedge LTUengage_{rqst} \quad (4.7)$$

and

$$disarm_{rqst} = trans_{rqst} \vee (\chi_{\mathbb{R}}(\delta_T) \wedge LTUengage_{withdraw}) \quad (4.8)$$

respectively. Comparing the calculation of arm_{rqst} and the first term of Equation 4.8, the implication is that movement of the control inceptor from the right to the left portion of the gate would cancel the LTU arming request and instead trigger the disarming. The latter term of $disarm_{rqst}$ implicates a dismissal of the arming without movement of the inceptor, but instead with the operator action $LTUengage_{withdraw}$. Similarly to $shutdown_{rqst}$, the operator actions, processed via $LTUengage_{rqst}$, $LTUengage_{withdraw}$ and $hover_{rqst}$ are derived from the Nominal and Fallback procedure harmonization and are hence explained later on in Section 4.4. For now, it is assumed that their origin is known.

The automation is responsible for the operation of the high-lift system in the Fallback concept. For this reason the Decision-Atomics provides the Decision-Execution with the following additional information.

To ensure the pilot authority with regards to the high-lift system operation, the movement of the flaps is managed by the crew. This is done via $flaps_{UP}$ and $flaps_{DOWN}$ that are communicated to the automation via dedicated input items already introduced in Section 2.4.3.1. The average flap deflection δ_F is sent as well. In addition, the airspeed V_{CAS} is forwarded along with $V_{CAS_{avail}}$. The latter originates from the signal integrity checking and evaluates whether the airspeed information is valid. The two variables are required in order to ensure the vehicle's structural integrity. This is achieved using a protection function that is explained in Section 4.2.1.

4.2.2 Decision-Making

This section explains the mechanics of the Decision-Making process. It presents the transition functions of the State Machine M_{FB} . In Figure 4.1 the transition sets and actions are denoted as T_i and A_i respectively with $i \in \mathbb{N}$ to conform with the conventions previously introduced in Section 2.2. In the figure, the edges marked in red are the ones that deal with the mitigation strategies, explained later in Section 4.3.1.

For the sake of clarity, the actions during each state are briefly summarized with the introduction of the transition functions. The detailed explanation is provided later on in Section 4.2.4. In addition, all input combinations that are not depicted on the edges of Figure 4.1 imply that the state is retained. This follows the pattern previously explained in Section 2.2.3 with Equations 2.22 and 2.24.

T1/A1: The Reconfiguration to Fixed-Wing Mode

The first transition function expressed in Figure 4.1 describes the transition into the state *Wingborne*. This transition is relevant during the transition procedure, explained later in Section 4.3.1.1. As evident from the figure,

$$\delta(\{Hover, unused\}, u_1) = \{Wingborne, HoverDisarmed\}, \quad (4.9)$$

where

$$t_1 = wingborne_{rqst} \quad (4.10)$$

is the transition condition.

With this state, the “wingborne” control law is engaged. In addition, the control allocation is fed the information to not utilize the LTUs for force and moment production anymore and instead drive them down and disengage them. Lastly, when this state is entered, the indication items are provided with the information that the reconfiguration has taken place. The actions are explained in Section 4.2.4.

T2/A2: The Arming of the Powered-Lift Mode

As explained later in Sections 4.3.1.2 and 4.3.1.4, this transition function is required for the retransition procedure. It is formulated as

$$\delta(\{Wingborne, HoverDisarmed\}, u_2) = \{Wingborne, HoverArmed\}. \quad (4.11)$$

The transition condition is

$$t_2 = arm_{rqst}. \quad (4.12)$$

This edge is visible in the second State Machine layer in Figure 4.1.

The mode of the law remains unchanged, i.e. the wingborne law is engaged. However, in this state configuration, the control allocation is forced by the automation to initiate a ramp up of the LTU RPM to idle revolution rates. It also triggers an indication item pattern change, so as to provoke the operator awareness for the steps ahead. Those are explained later in Section 4.3.1.

T3/A3: The Reconfiguration to Powered-Lift Mode

The function presented here is the last automation step towards achieving powered-lift flight and retransitioning to hover flight. As seen later in Sections 4.3.1.2 and 4.3.1.4, it applies for both full and decreased powered-lift system performance.

The transition function is expressed as

$$\delta(\{Wingborne, HoverArmed\}, u_3) = \{Hover, unused\}, \quad (4.13)$$

for which the transition condition is

$$t_3 = hover_{rqst}. \quad (4.14)$$

This disengages the wingborne law and instead engages the previously armed hover mode. In this state constellation, the control allocation is allowed to fully utilize the powered-lift system for moment and force production. Lastly, the indication items inform the operator of the transition. All these actions are elaborated upon later in Section 4.2.4.

T4/A4: The Retransition Mitigation back to Fixed-Wing Mode

The last state changes are related to the mitigation strategies during retransition as explained later in Section 4.3.1.2. The Fallback reverts back to fixed-wing mode with the powered-lift system fully disarmed. This is done with the function

$$\delta(\{Wingborne, HoverArmed\}, u_4) = \{Wingborne, HoverDisarmed\}, \quad (4.15)$$

with the transition condition

$$t_4 = disarm_{rqst}. \quad (4.16)$$

In the state constellation $\{Wingborne, HoverArmed\}$, t_4 is performed instead of t_3 , previously introduced in Equation 4.13. Thus, the engagement process of the LTUs is aborted. As a consequence, the Fallback system actions are equivalent to the ones, previously introduced with $A1$.

With Equation 4.15, the transition functions of M_{FB} are fully specified. When comparing the transition conditions, found in this section, with the transition conditions of the Nominal system, found in Chapter 3, it is evident that the complexity is reduced substantially. The transition conditions of the Fallback system are directly driven by the operator actions. In contrast, the Nominal system includes checks that the state changes are performed in the correct envelope under proper external conditions. Apart from plausibility checks of the operator requests, the automation concept does not impose any restrictions as to the operator authority.¹

It must be noted that the word “mode” is used in the description of the transitions of this section and the word “phase” is avoided. This is intentional because in this automation concept, the responsibility of the function is the provision of the required law. The task to ensure that the appropriate mode is engaged in the correct flight phase is instead allocated to the operator. For this, clear procedures are defined later on in this chapter with Section 4.3.1 that aid in the selection by procedure design.

4.2.3 Takeover State Evaluation

Because the Fallback can takeover at any moment during the operation of the Nominal system, it was already mentioned that the correct initialization of the State Machine M_{FB} must be ensured. Therefore, the starting state of the Mealy Machine must be specified. This is established via the so-called “initialization” function, which is introduced here.

It must be noted that according to the theory of State Machines and the theory summary, found in Section 2.2.3, an automaton may have only one starting state. Why the consideration here is permissible and why the starting state may vary based on the situation at the moment of takeover is explained in Appendix D.

The initialization function, responsible for the correct starting state assignment of M_{FB} can be defined as

$$Initialize : \mathbb{B} \times S_{LTU} \rightarrow S_{FB|1} \times S_{FB|2}. \quad (4.17)$$

¹Alternatively, certain restrictions may be imposed. For example, the airspeed could be utilized to prohibit the disengagement of the LTUs at low dynamic pressures (which requires a modification of t_1 in Equation 4.10). This, on the one hand, increases the automation dependency on external sensors with less reliability. On the other hand, it must be noted that then certain actions will no longer be possible. For example, a use-case may be that the operator performs the reconfiguration at a severely low airspeed. Should this restriction be imposed, this maneuver will no longer be enabled. This reduces the automation flexibility and may pose an issue if such a dive is required in the event to unforeseen hazards. Due to the known uncertainties of the novel eVTOL configurations, full operator authority in the Fallback system is actively pursued.

The range of the function are the initial states. The first entry of the domain (function input) is the evaluation whether the Nominal system is engaged prior to the moment of takeover. For the sake of readability, this is expressed as $NominalEngaged \in \mathbb{B}$. The other domain member is the state of the high-degree of automation State Machine, found in Chapter 3. The set was introduced with Equation 3.1. Based on those two variables, the starting states can be determined unambiguously. This is done in accordance with the Truth Table 4.3. Therefore, the initial states are obtained as

$$\{s_{FB|1_0}, s_{FB|2_0}\} = Initialize(NominalEngaged, s_{LTU}). \quad (4.18)$$

Table 4.3: *Takeover Function Truth Table*

NominalEngaged	s_{LTU}	Initialize	
<i>false</i>	-	<i>Hover</i>	<i>unused</i>
<i>true</i>	<i>Engaged</i>	<i>Hover</i>	<i>unused</i>
<i>true</i>	<i>Disengaging</i>	<i>Wingborne</i>	<i>HoverDisarmed</i>
<i>true</i>	<i>Disengaged</i>	<i>Wingborne</i>	<i>HoverDisarmed</i>
<i>true</i>	<i>Engaging</i>	<i>Wingborne</i>	<i>HoverArmed</i>

The first row of Table 4.3 implies that the Nominal system was not in command. This could be applicable for example during testing of the prototype solely with a Fallback system in order to flight-prove the Fallback algorithms. In this case, the default values are taken. In the cases, where prior to a takeover the Nominal system was in command, then the starting states are differentiated with regards to the state of s_{LTU} .

If M_{LTU} is in the *Engaged* state, then from Chapter 3 it is known that the LTUs are fully utilized. From the explanations in Section 4.2.2 this corresponds to the tuple, found in the second row of Table 4.3.

Whenever s_{LTU} is either *Disengaging* or *Disengaged*, then the LTUs are not being used by the control allocation of the Nominal system and are in the process of shutting down or fully shutdown respectively. The Fallback system does not differentiate between the two modes as seen in Section 4.2.2. During *Wingborne*, the fallback control allocation no longer utilizes the LTUs and in *HoverDisarmed*, they are gradually reduced to standstill. Therefore, the output tuples of the initialization function in rows three and four of Table 4.3 are justified.

Lastly, the LTUs are commanded to idle RPM during M_{LTU} 's state *Engaging*, which according to the explanations of Section 4.2.2 is equivalent to the statement in last row of Table 4.3.

It must be noted that in the beginning of this chapter the statement is made that one of the reasons for a takeover is if the Nominal system experiences a design error. Therefore, taking the state of the State Machine M_{LTU} may provide unsuitable starting states if the state of s_{LTU} is erroneous. This is counteracted in the following manner.

Parallel to the Nominal system, a so-called “functional monitor” is being executed. The monitor provides continuous checking as to the plausibility of the Nominal system. Among others, the monitor ensures that the sequences of commands, states and actions of the system are performed only in the allowed flight envelopes. The envelope monitoring is established in a manner, which is independent of the Nominal system, increasing the confidence that the signals produced by the Nominal system are correct. The functional monitor is being developed by Hannes Hofsäß of TUM Institute of Flight System Dynamics and is not in the scope of the thesis. The last set of functionally correct values that the Nominal system has outputted are referred to as “last valid values”. Instead of taking the last value of s_{LTU} that is available to the Fallback system, the last valid value is used for the computation. This ensures the correctness of the initialization of the takeover function.

4.2.4 Decision-Execution

With the State Machines of the automation defined, this section proceeds to specify the exact information communicated to the surrounding functional modules and the actions performed due to that data. The information, processed by the Decision-Atomics of Section 4.2.1 and the state of M_{FB} are forwarded to the Decision-Execution in order to generate the information, required by the surrounding systems.

The Fallback functions, presented here help facilitate the transition and retransition. The automation schedules operational aspects of the law and allocation. Those two topics are discussed first. Afterwards, the operation of the high-lift system is necessary to set cruise or landing configurations. Hence, the Decision-Execution is responsible for the operation of the flaps as well, which is discussed next. Lastly, the crew information supply via the indication items is discussed.

Law

Based on the summary in Section 2.4.1.2, the law is split into two separate modes - “hover” and “wingborne”. The change of mode modifies the command variable selection and certain envelope limits. The choice of the active law mode is solely dependent on the state $s_{FB|1}$.

In addition to this, as discussed in Section 2.4.1.2, the hover mode of the law includes a command variable blend which is done over the dynamic pressure. The robustness of the blend can be increased in the cases of a failure of the dynamic pressure. This can be done in the following manner.

The blending between the command variables in the law is done with the use of the blending variable $\lambda = \{\lambda \in \mathbb{R} | 0 \leq \lambda \leq 1\}$. The output of the blending is

$$x = (1 - \lambda)x_1 + \lambda x_2. \quad (4.19)$$

In the equation, x_1 and x_2 are the two different command variables subject to the blending. The generation of λ is calculated as

$$\lambda = \begin{cases} \lambda_N(V_{CAS}) & \text{if } V_{CASavail} \\ 0 & \text{if } \neg V_{CASavail} \wedge (\delta_T \leq \delta_{T,D}) \\ 1 & \text{otherwise.} \end{cases} \quad (4.20)$$

In the equation, $\lambda_N(V_{CAS})$ is the nominal blending function, dependent on the airspeed. This is not in the scope of this thesis. $\delta_{T,D}$ is the division of the control inceptor that separates the hover and transition regions, explained previously in Section 2.4.3.1. Here it is implied that whenever λ is zero the command variable required for precise hover is taken.

Lastly, if necessary, the state of reconfiguration with regards to the high-lift system may need to be communicated to the law for proper thrust mapping. Why this may be necessary is explained later on in Section 4.3.4.2.

Control Allocation

As evident from the structure of the State Machine M_{FB} in Section 4.2.2 and in Figure 4.1, the *Hover* mode can be armed, but the *Wingborne* does not need arming. The arming of the hover mode is used to determine the state of the powered-lift system prior to engaging the hover mode. The wingborne mode utilizes solely the control surfaces. As explained in Section 2.4.1.2, they are used throughout the whole flight envelope. Therefore, a check of control surface actuation integrity is not required.

Table 4.4: *Control Allocation Action Truth Table*

S_{FB} 1	S_{FB} 2	Control Allocation Action
<i>Hover</i>	<i>unused</i>	Full utilization of the LTUs.
<i>Wingborne</i>	<i>HoverDisarmed</i>	Ramp down of the LTU commands.
	<i>HoverArmed</i>	Ramp up of the LTU commands to idle RPM.

Whenever in the hover mode, the automation function communicates to the control allocation that the full utilization of the LTUs is permitted. When in the wingborne mode, the LTU usage for force and moment generation is forbidden. The actions that the Control Allocation takes are dependent on the state $S_{FB|2}$.

Whenever *HoverArmed* applies, the control allocation is given the instruction to provide idle RPM to the powered-lift system. This is a necessary step to facilitate proper awareness of the operator about the state of the LTUs prior to entry into the hover mode. In addition, this allows for a smooth engagement of the hover mode, because the LTUs are already functional.

In the *HoverDisarmed* mode, the control allocation is responsible for driving the RPM of the LTUs to standstill. It must be noted that in the Nominal system discussions of Chapter 3, there is differentiation between ramp down and zero RPM command via the states *Disengaging* and *Disengaged* of the automaton M_{LTU} respectively. This can be done because the check for motor disengagement is performed by the automation function. In the Fallback system, this action is allocated to the operator as explained later in Section 4.3.1. Therefore, this state separation by the automation is not necessary. The actions, communicated to the control allocation, are summarized in Table 4.4.

In addition to this, the control allocation is informed of any effector malfunction. This causes a reconfiguration in the algorithms and enables a more feasible pseudocontrol generation with the remainder of the effectors.

High-Lift System

In the Fallback system, the operation of the high-lift system is within the responsibilities of the operator. However, the automation provides a simple protection in the cases where a critical airspeed is exceeded. In these events, the flaps are automatically retracted gradually so as to avoid structural damage. This can be done as follows.

The information with regards to the aerodynamic speed and its availability is communicated to the Decision-Execution by the Decision-Atomics with the variables V_{CAS} and $V_{CASavail}$. In addition, the operator requests - $flaps_{DOWN}$ and $flaps_{UP}$ - are sent. Lastly, the average flap deflection is determined via δ_F . The upper airspeed limit is determined as

$$\delta_{F_i,lim} = \begin{cases} \delta_{F_i,max} & \text{if } \neg V_{CASavail} \\ \delta_{F_i,max} & \text{if } V_{CAS} < V_{FEfull} - \Delta V \\ \delta_{F_i,min} & \text{if } V_{CAS} > V_{FE0} - \Delta V \\ \delta_{F_i,max} - (V_{CAS} - V_{FEfull} + \Delta V) \cdot \frac{\delta_{F_i,max} - \delta_{F_i,min}}{V_{FE0} - V_{FEfull}} & \text{otherwise.} \end{cases} \quad (4.21)$$

The airspeed definitions in the relational operators stem from Section 2.3. Provided the dynamic pressure is not available, it is visible from the equation that the upper limit is not restricted. Otherwise when in range, the upper limit is computed such that no critical value can be exceeded.

The limit of above needs to be applied. For this, the limit function is introduced. It ensures that its output is within a specified range. An example of a limit function is

$$limit(x, [x_{min}, x_{max}]) = max(min(x, x_{max}), x_{min}), \quad (4.22)$$

where the first argument is the non-restricted value and the latter two arguments are the respective upper and lower limits in that order.

With the computation of the allowed upper limit of the deflection, the commanded flap deployment angle $\delta_{F_i,cmd}$ is then

$$\delta_{F_i,cmd} = \mathit{limit}(\delta_{F_i,cmd,raw}, [\delta_{F_i,min}, \delta_{F_i,lim}]). \quad (4.23)$$

In the equation $\delta_{F_i,cmd,raw}$ is a function of the operator inputs $flaps_{DOWN}$ and $flaps_{UP}$.

The command mechanisms in the events of failures are not explained here. For example, the command mapping must change if a hardover in one of the high-lift units is registered. In addition, the cases where the system receives both a flap retraction and deployment request simultaneously must be handled. The methods to tackle such issues are application-specific and hence not in the scope of this section. The mitigation mechanisms are instead explained in Chapter 5.

Indications

In order to facilitate adequate situational awareness as to the state of reconfiguration and law mode in the Fallback system, the indication item previously presented in Section 2.4.3.2 is used. The automation must provide the data required for unambiguous generation of the different applicable color patterns of the indication item.

The exact logic for the pattern choice of the indication items of Section 2.4.3.2 is found in Table 4.5. Each row of the table refers to a specific and unique color pattern. The color pattern is not in the scope of this thesis. In the table $\Delta\lambda = \{\Delta\lambda \in \mathbb{R} | 0 \leq \Delta\lambda \leq 1\}$ is a threshold, beyond which the command variable blending to transition flight of Equation 4.20 is no longer negligible. $caution_{dismiss} \in \mathbb{B}$ provides information whether the cautions, triggered by the automation module to the cautions and warnings indication item, have been dismissed. From Table 4.5 it is noticeable that the computation of the necessary color coding is an algebraic function of the current automation state and the State Machine inputs.

The link between pilot and automation actions and changes in indication item color pattern within the transition and retransition procedure are elaborated upon in the next sections. In addition to this, the operator is informed via the cautions and alerts in the event of high-lift system protection unavailability due to airdata loss. A similar warning is generated for detected effector failures.

4.3 Design Analysis

In the previous section the design of the Fallback automation was presented. It is responsible for command supply to the high-lift system and scheduling of the law and control allocation modes. In addition, it supplies the operator with feedback to ensure situational awareness.

Table 4.5: Fallback System Transition Indication Item Truth Table

Description	s_{FB1}	s_{FB2}	δ_T	λ	$w_{b,rfg}^*$	caution	dismiss
In Hover, stay in Hover requested	<i>Hover</i>	-	$\chi_H(\delta_T)$	$< \Delta\lambda$	-	-	-
In Hover, leave Hover requested	<i>Hover</i>	-	$-\chi_H(\delta_T)$	$< \Delta\lambda$	-	-	-
In Transition, go to Hover requested	<i>Hover</i>	-	$\chi_H(\delta_T)$	$\geq \Delta\lambda$	-	-	-
In Transition, stay in Transition	<i>Hover</i>	-	$\chi_T(\delta_T)$	$\geq \Delta\lambda$	-	-	-
In Transition, go to Fixed-wing	<i>Hover</i>	-	$\chi_{WV}(\delta_T)$	$\geq \Delta\lambda$	-	-	-
LTU disengagement, throw caution	<i>Wingborne</i>	<i>Hover Disarmed</i>	$\chi_{WV}(\delta_T)$	-	<i>true</i>	-	-
LTU disengagement, caution not dismissed	<i>Wingborne</i>	<i>Hover Disarmed</i>	$\chi_{WV}(\delta_T)$	-	<i>false</i>	<i>false</i>	-
In Fixed-wing, after caution dismissal	<i>Wingborne</i>	<i>Hover Disarmed</i>	$\chi_{WV}(\delta_T)$	-	<i>false</i>	<i>true</i>	-
In Fixed-wing, retransition Requested	<i>Wingborne</i>	<i>Hover Disarmed</i>	$\chi_T(\delta_T)$	-	-	-	-
LTU Engagement, throw caution	<i>Wingborne</i>	<i>Hover Armed</i>	$\chi_T(\delta_T)$	-	-	-	-

Calculated as $EdgeIncrease(s_{FB1} == Wingborne)$. For the calculation of $EdgeIncrease$ see Equation 2.14.

The automation design addresses and fulfills the properties, summarized in Section 4.1. However, the mechanics of how the requirements of that section are fulfilled were not explained. This is instead performed in this section.

This section is organized as follows. The automation module is tailored to allow for increased operator authority and oriented towards precise control concept scheduling by the pilot. However, in order to allow for an adequate takeover from the Nominal system and from then on enable the reconfiguration to wingborne flight and back, procedural considerations are necessary. Section 4.3.1 demonstrates that with the given automation design, where transition and retransition procedures can be derived.

Out of the derived procedures, a requirement on the Fallback control law is derived. This requirement is elaborated upon in Section 4.3.2 and specifies how the mapping of the throttle should be designed with relation to the control inceptor deflections δ_T to enable a non-time critical operator decision-making process during the transition and retransition reconfiguration procedures.

In addition, the transition and retransition procedures establish a clear link between aircraft flight phases and applicable automation states. For the sake of completeness, Section 4.3.3 summarizes the mentioned state allocation to the aircraft flight phases. Lastly, in Section 4.3.4, the system response in the event of failures or crew deviations is analyzed.

4.3.1 Transition and Retransition Procedures with the Fallback System

This section summarizes the actions of pilot and automation that facilitate the transition and retransition with the Fallback system for both normal and abnormal scenarios. The causal chain of events during the transition and retransition is presented. A link between these events and the actions of the automation is established. Firstly, this observation is performed for the transition and retransition under nominal conditions. This can be found in Section 4.3.1.1 for the transition and in Section 4.3.1.2 for the retransition process.

Then, the same is done in abnormal scenarios in Sections 4.3.1.3 and 4.3.1.4. After each analysis, the chain of events is summarized in a process flow graph for better understanding. It must be noted that for the sake of simplicity, it is assumed that the crew does not deviate from the procedures. Deviations are instead observed later on in Section 4.3.4.

4.3.1.1 Normal Transition Procedure

In these observations, it is assumed that the aircraft is in the hover phase in order to cover the full transition to wingborne flight. This implies that the control inceptor position is

$$\delta_T \in \mathbb{H}, \quad (4.24)$$

whereas the states of M_{FB} are $\{Hover, unused\}$. As a consequence, in terms of indication patterns, the first row of Table 4.5 is applicable.

In order to start the process, the operator requests an aircraft acceleration. This is achieved when the pilot moves the throttle lever position δ_T forward. The automation informs the pilot that the request to leave the hover region is processed via a change in the indication item. This occurs when the detent position is crossed, i.e. when $\delta_T \notin \mathbb{H}$. There, the color pattern of row two of Table 4.5 is applicable.

As the aircraft accelerates further and gains airspeed, the blending begins to take effect. The blending variable calculation is presented with Equation 4.20. Whenever

$$\lambda \geq \Delta\lambda, \quad (4.25)$$

the indication item changes. The color pattern is dependent on the current throttle position. Generally, it is good practice for the operator to keep the throttle lever in the transition region of the control inceptor in order to limit the possibility of inadvertent powered-lift system deactivation due to wrong actions. In this case, i.e. $\delta_T \in \mathbb{T}$, then row four of Table 4.5 is applicable and the crew is informed that the transition phase has been reached.

Whenever the control inceptor has been increased to the maximum allowed value in the throttle region, the operator must wait for the disengagement speed to be reached. This holds after reaching V_{STALL} , however V_{SAFE} is advisable in order to guarantee obstacle avoidance capabilities after powered-lift system shutdown. Subsequently, the operator may start preparations for the an LTU disengagement, the pilot proceeds to move the throttle to the left portion of the control inceptor gate, i.e. $\delta_T \in \mathbb{L}$. This implies that $trans_{rqst}$ in Equation 4.5 is *true*. By definition of the indication, row five of Table 4.5 is in effect. This notifies the crew that the Fallback system has processed the movement of the throttle into the wingborne region of the control inceptor.

As seen in Chapter 3, the conditions to initiate the LTU disengagement in the Nominal system are managed by the automation. In the case of the Fallback system, this task is allocated to the pilot. Therefore, the pilot is required to determine that the LTUs are unused by the control allocation and can therefore be turned off. The condition can be fulfilled by maintaining straight flight and ensuring that an excessively high climb is not commanded. Both of the maneuvers would require LTU utilization. The check of the LTU usage can be done via separate cockpit indications, which are not in the scope of this thesis.

When the operator assesses the flight conditions as suitable, the disengagement is triggered by the $shutdown_{rqst}$ action. As a consequence, the transition condition t_1 of Equation 4.10 holds, and the state transition to the wingborne mode of Equation 4.9 is applicable. The State Machine M_{FB} thus transitions to the $\{Wingborne, HoverDisarmed\}$ state tuple. In addition, the color pattern of the indication items changes to reflect the change via row six of Table 4.5.

In the state *Wingborne*, the control allocation shuts down the LTUs. This is explained previously in Section 4.2.2. As seen in Chapter 3, the Nominal system automation is required to verify the correctness of the LTU disengagement. For the Fallback system, it is

up to the operator to ensure full shutdown of the powered-lift system. This is achieved via the remainder of the indication items that show the distributed propulsion system status and also via visual confirmation. The enabling of this check is not in the scope of this thesis.

As evident from the indication status in effect - row six of Table 4.5 - the automation commissions a warning that is forwarded to the cautions and warnings system. This is meant to raise awareness that the aircraft is already in the wingborne mode of the Fallback control concept. However, whether the wingborne flight phase is truly obtained depends on the status of the powered-lift system. Prior to ensuring the full disengagement, the operator must not exceed the airspeed of V_{LSNE} . This airspeed is introduced previously in Section 2.3.1.2.

Provided the LTUs are fully off, the warning is dismissed by the operator. The dismissal is not in the scope of this thesis. However, for the sake of completeness, this indication status is reflected in row seven of Table 4.5. If LTUs are not capable of shutting down, then mitigation actions are required. The reason for such a scenario is provided in Section 2.3.1. In this section, no failures are assumed. The mitigation actions due to the failure are examined later in Section 4.3.1.3.

Upon reaching the wingborne flight phase, the reconfiguration process is continued by retracting the flaps. How this is done was explained previously in Section 4.2.2. After fully retracting the flaps, the pilot may fully use the wingborne region of the control inceptor and accelerate beyond the airspeed V_{FE} . V_{FE} was introduced in Section 2.3.2.

From the above-listed actions, a transition procedure emerges. The derivation of the transition procedure with the Fallback system be found in Figure 4.2. The execution flow of the procedure is depicted in Figure 4.2. The figure considers the actions of the software or crew. The actions of the latter are depicted in gray. In this and the following sections, the identifiers of the procedural steps conform to the following convention. Each step begins with either a “t” or “r”, indicating whether this step belongs to the transition or retransition procedures respectively. If present, the next character “m” signifies that this is part of the mitigation strategy. This is followed by the numbering in order of the causal chain of events. If the step is part of the expected flow, then the identifier ends. Alternatively, if this step is off-nominal - such as the start of the mitigation strategy or the mitigation strategy options themselves, then the identifier is supplemented with additional characters (e.g. “a” or “b”) to indicate this.

4.3.1.2 Normal Retransition Procedure

In order to observe the full reconfiguration to the powered-lift mode and entry to the hover phase, it is assumed that the aircraft is in the wingborne flight phase. This means that M_{FB} 's states are $\{Wingborne, HoverDisarmed\}$. Furthermore, the flaps are fully retracted.

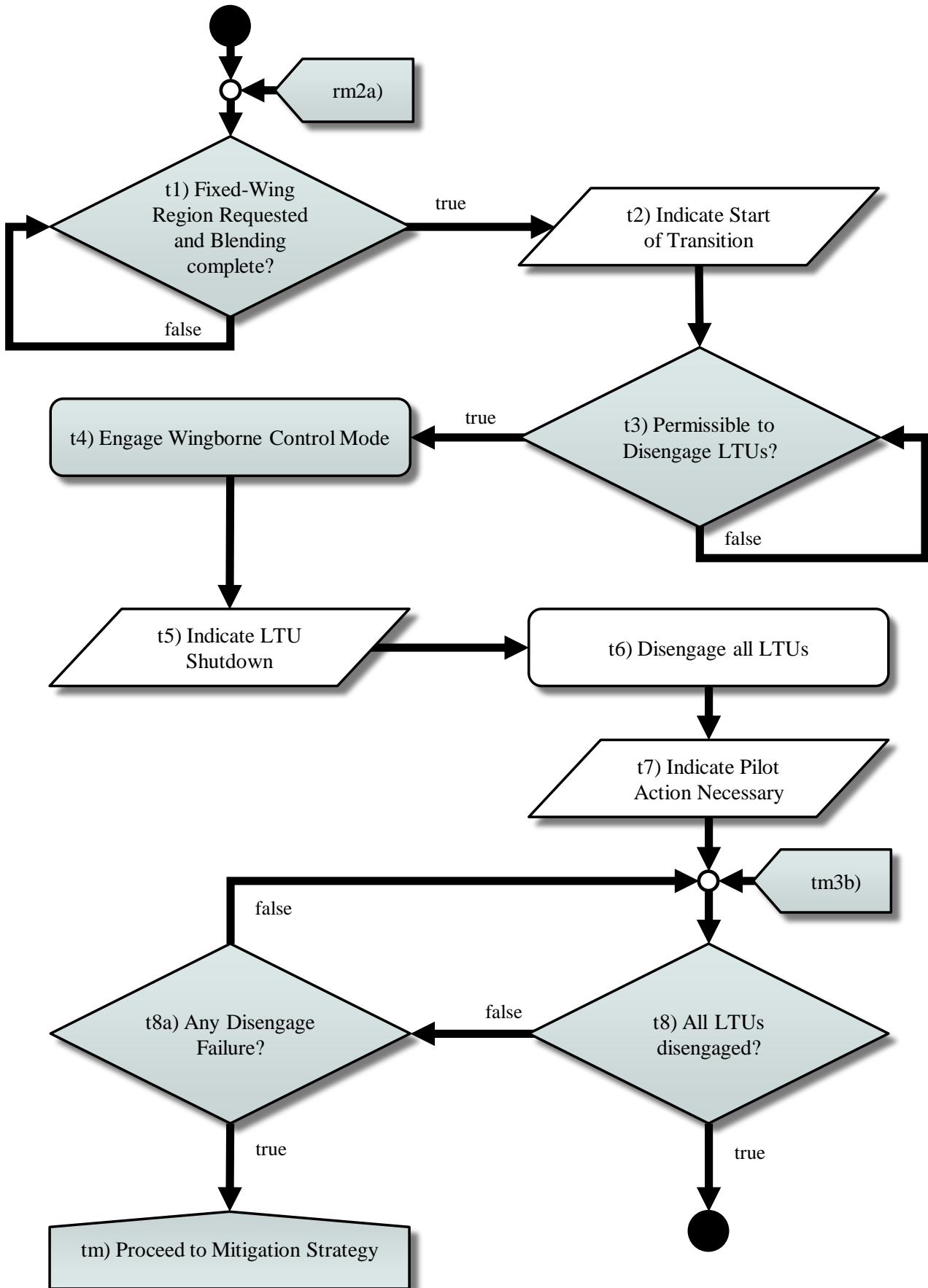


Figure 4.2: Transition Process Flowchart

Initially, the operator reduces the throttle, whereby the aircraft starts to decelerate. When the speed is below the never exceed speed for dirty configurations V_{FE} , the pilot can start deploying the high-lift system. The operation of the flaps was explained previously in Section 4.2.2.

Next, the V_{LSNE} speed is crossed. Below this speed, the activation of the powered-lift system can commence. The operator is subsequently required to move the control inceptor to the transition region. As a consequence, the event $\delta_T \in \mathbb{R}$ is registered by the Fallback system. The automation sends this condition to the indication item, which change their status as per row nine of Table 4.5, facilitating the pilot awareness.

In order to actually begin the LTU engagement process, the operator requests it via $LTUengage_{rqst}$. Consequently, the conditions for t_2 are applicable. Thus via Equation 4.11, the Fallback system automation transitions to the state tuple $\{Wingborne, HoverArmed\}$. As explained in Section 4.2.2, the control allocation ramps up the LTU RPM to the idle revolution rates. Also seen from Section 4.2.2, the automation triggers an indication item status change via row ten of Table 4.5. It is seen from the table that a warning is produced as well.

The reason for the warning is that the pilot must check the correctness of LTU engagement. Only after ensuring that the LTUs are fully engaged is the activation of the hover control mode of the Fallback system permissible. For recollection, in the Nominal system this check was performed by the automation as presented in Chapter 3. Here the task is allocated to the system operator.

If the powered-lift system cannot fully engage, then mitigation strategies are in effect. They are discussed in detail in Section 4.3.1.4. The assumption is that the operation of the powered-lift system is fault-free. Upon performing the check of activation correctness, the crew proceeds to engage the hover mode of the Fallback system by performing the request, summarized with $hover_{rqst}$. This triggers the state change of the automation to $\{Hover, unused\}$ via Equation 4.13. The control allocation is permitted to utilize the LTUs for force and moment production. Furthermore, the indication item changes according to row four of Table 4.5.

The control mode reconfiguration for the transition phase is fully performed and a deceleration below V_{STALL} is permitted. The pilot can do so by moving the throttle more in direction of the hover region. As long as this region is entered, in other words

$$\chi_{\mathbb{H}}(\delta_T) \tag{4.26}$$

is registered by the system, the operator is informed of the hover flight phase request via the indication item change as per row three of 4.5. With further deceleration, the blending variable λ decreases and consequently the indication color patterns change in accordance to row one of Table 4.5. This informs the operator that the hover flight phase is reached.

As performed for the transition in the previous section, here a retransition process flow is derived. It is visible in Figure 4.3. The convention in terms of coloring and numbering was already explained in Section 4.3.1.1.

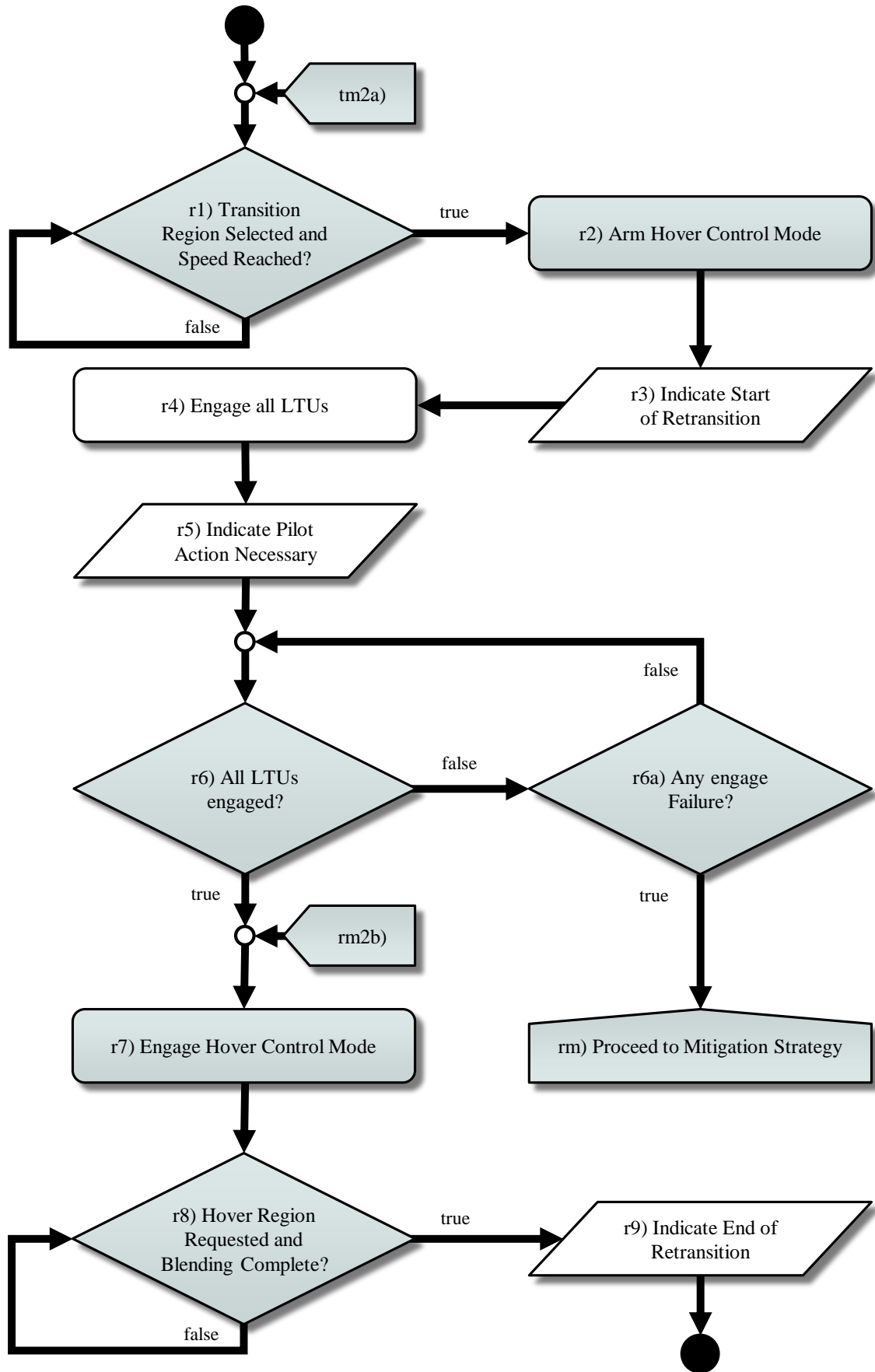


Figure 4.3: Retransition Process Flowchart

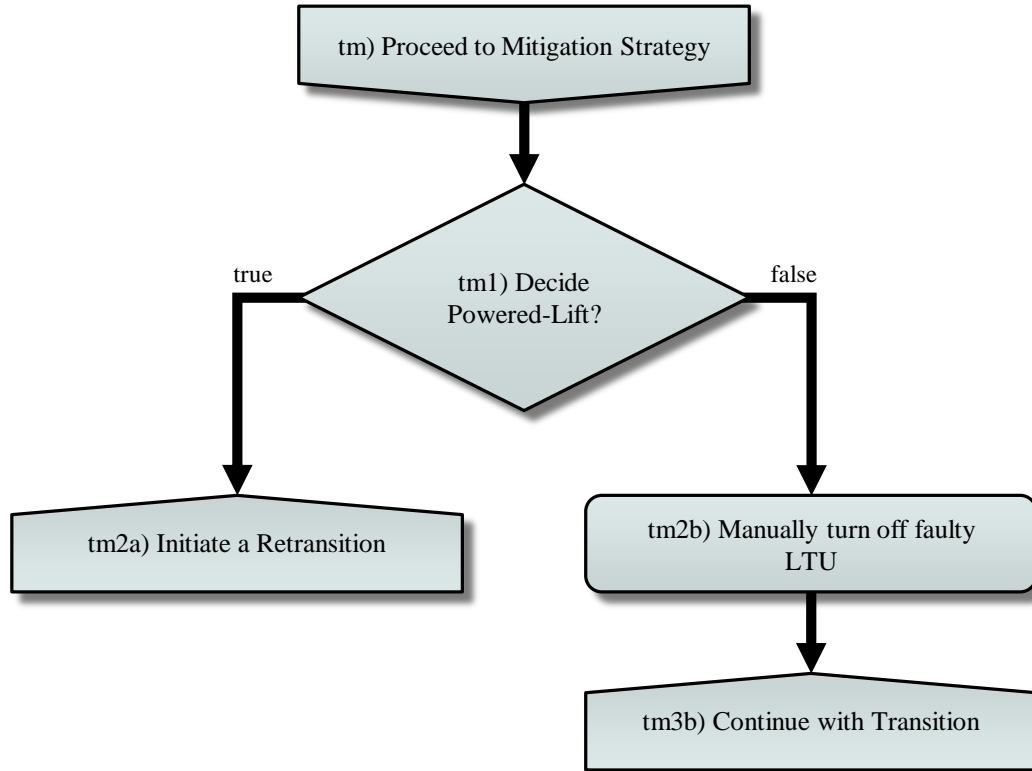


Figure 4.4: *Transition Mitigation Process Flowchart*

4.3.1.3 Abnormal Transition Procedure

The abnormal transition procedures are in effect if one or several LTUs are incapable of disengaging. The analysis during the transition process hence continues with step t4) onward. The sequence leading to this step is found in Section 4.3.1.1.

In step t4) the operator has engaged the wingborne Fallback control mode. The State Machine M_{FB} is therefore in the state tuple $\{Wingborne, HoverDisarmed\}$. During the operator check t8a), the assessment is made that the LTUs did not fully disengage. The failure cases where this may occur are discussed in detail previously in Section 2.3.1.

Similarly to Chapter 3, the options at the pilot's disposal are to retain the wingborne control mode and or to revert to powered-lift flight. Those two options are examined in detail below. Figure 4.4 is used to illustrate the transition mitigation actions.

In the case of reversion to powered-lift flight, the initiation of the retransition procedure is required as visible in step tm2a). This implies that the whole retransition has to be performed as previously described in Section 4.3.1.2 and be performed in accordance to Figure 4.3. In this scenario, the check r1) need not be performed, as the aircraft is in the appropriate airspeed range by design of the transition procedure.

Full reconfiguration to wingborne flight requires a manual shutdown of the problematic LTU as visible in step tm2b) of Figure 4.4. Then an acceleration beyond V_{LSNE} is permissible.

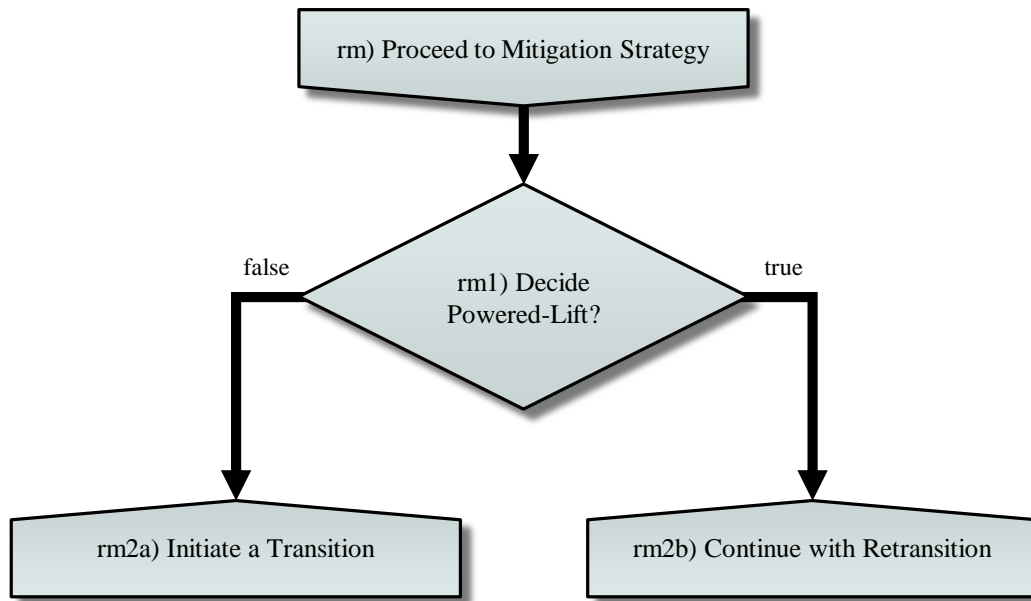


Figure 4.5: *Retransition Mitigation Process Flowchart*

As discussed previously in Section 2.3.1, a manual shutdown may not be possible. Then, by definition of Section 2.4.2, the aircraft has not fully entered the wingborne flight. Nevertheless, the operator is capable of retracting the high-lift system to reach a more aerodynamically efficient configuration. Section 4.2.2 demonstrates that the automation does not prohibit such actions. It is the pilot responsibility to not exceed V_{LSNE} . In addition, the operator decision does not need to be immediate, meaning it is not time-critical. The reason for this is that the aircraft is in a controlled state, facilitated by the law's wingborne mode.

The automation design remains unchanged regardless of whether or not LTUs may be manually disengaged. In order to revert to wingborne flight, the operator needs to execute the retransition process of Section 4.3.1.2. Therefore, the lack of manual disengagement does not add additional considerations on this mitigation strategy. By design, in mitigation to wingborne flight, the State Machine does not experience any state changes and therefore does not require functional modifications.

4.3.1.4 Abnormal Retransition Procedure

The retransition mitigation strategies assume a fault during retransition. Thus, this section continues from step r2) of Figure 4.3 found in Section 4.3.1.2.

By implication, the State Machine M_{FB} has the states $\{Wingborne, HoverArmed\}$. The control allocation via the relationship in Section 4.2.2 has set the command to the LTUs to idle RPM. However, the operator assesses the LTU engagement as failed, implying that one or several LTUs are incapable of engaging. This is the starting point of the retransition mitigation strategies. The available operator options in this scenario are found in Figure 4.5.

The options available to the operator are the same as in the Nominal system seen in Chapter 3. The pilot may either revert back to wingborne flight or confirm the entry into powered-lift flight. It must be noted that during this time the operator decision is not time-critical. The crew needs to only ensure that the airspeed does not exceed V_{LSNE} and does not go below V_{STALL} . The reason for the former is that the LTUs are in motion as the control allocation is sending idle commands. The reason for the latter is that the LTUs are not yet actively used for force and moment production, thus stalling the aircraft is possible.

The operator may decide to go back to wingborne flight in order to conduct a go-around and reattempt the retransition. For a more severe powered-lift failure, a wingborne landing may be necessary. Regardless of the reason, in order to revert back to wingborne flight, the operator is required to disarm hover mode. This is done via t_4 found in Equation 4.16. This implies that the request $LTUengage_{withdraw}$ is necessary. According to Equation 4.15, the automation reverts back to the states $\{Wingborne, HoverDisarmed\}$.

Provided not a critical number of LTUs has failed, going to powered-lift flight is possible. Because a manual inspection of LTU activation is enforced, the operator is aware of the performance losses of the hover mode. The pilot can continue with the retransition procedure with step rm2b) of Figure 4.5. As evident from Figure 4.3, the subsequent actions are identical to the nominal case. In addition, prior to these actions the automation has not experienced a state change. As a consequence, the mitigation strategy does not add complexity to the automation design. The identical pilot actions in normal and abnormal situations that lead to powered-lift flight do not add to the system's operational complexity.

As previously seen in Chapter 3, the Nominal system is responsible for maintaining safe envelopes. This included not stalling in pure wingborne flight and also not exceeding any critical speeds, like V_{LSNE} or V_{FE} . With the exception of the protection for exceeding V_{FE} of Section 4.2.2, such functions are not available in flight with the Fallback system. This is actively pursued by the system design. This choice allows for unrestricted operator authority and minimizes the dependency on the airspeed.

Comparing the process flows of Fallback found in this section with the ones of the Nominal System found in Section 3.3, it is visible that in the Fallback system the crew workload increased. Firstly, the amount of actions performed by the pilot is much greater. Not represented in this section are the adherence to a safe envelope, the management of the high-lift system and the law operation. They increase the workload further. The elevated workload is a consequence of reducing the level of automation and requiring more operator authority.

Similarities can be found between the sequences of events in this section and the ones for the Nominal system found in Section 3.3 of Chapter 3. Namely, the order of the actions during transition and retransition overlap largely, despite the large differences of control concepts and underlying automation. In the Fallback system, however, the complexity

of the automation is reduced greatly. This is evident by the number of transitions, their complexity and the decreased sensor dependency. This makes the Fallback automation more robust and transparent.

This section presented solely the transition and retransition procedures with the Fallback system. In **Contribution 2** the claim of harmonization of the procedures with Fallback and Nominal system is made. From the figures found in this section and the ones for the Nominal System in Section 3.3 of Chapter 3, this does not appear evident yet. The exact methods the procedures are harmonized with is instead explained later on in Section 4.4 where the whole integrated operation of Nominal and Fallback system is analyzed.

4.3.2 Fallback Throttle Mapping and Blending Requirements

In the previous section it was evident that individual steps in the process flow require the achievement of given airspeed ranges. For example, in order to initiate the LTU disengagement, the aircraft needs to exceed the stall speed but at the same time never exceed the limit of V_{LSNE} . In addition, requirements to the control inceptor placement during the sequence are set. In the example above, it was required that the throttle lever position is in the gate.

In order to facilitate the proper execution of the transition and retransition sequences, requirements on the traction thrust mapping of the Fallback concept with relation to the control inceptor setting are set. This section analyzes the procedures of Section 4.3.1 with regards to the thrust settings at the relevant control inceptor positions.

First, the borders of the hover and transition/retransition flight phases are examined. In the procedure descriptions they are handled in t1) and r8) of the transition and retransition procedures respectively. The state indications in Table 4.5 address them in the first three rows.

According to the flight phase division of Section 2.4.2, the border of the two flight phases is dependent on the aircraft speed, where speeds below V_{HOVER} are attributable to the hover flight phase. The control inceptor division between the two regions - hover and transition/retransition is distinguished by the detent at the throttle lever position $\delta_{T,D}$.

In order to establish a link between control inceptor, traction thrust command and procedure flow, the following requirement is imposed onto the command mapping and nominal blending parameter. Whenever the throttle lever is at the detent position $\delta_{T,D}$, the traction thrust command shall be equivalent to the thrust necessary to sustain a steady-state straight and level flight with a ground speed of at least V_{HOVER} . In addition, the nominal blending of $\lambda_N(V_{CAS})$ mentioned in Equation 4.20 shall initiate at an airspeed, with which at minimum a kinematic speed of V_{HOVER} is flown. Both properties must be achieved even in the presence of the most undesirable external conditions according to the aircraft ConOps.

The reasons for these requirements are as follows. In the hover region of the control inceptor, the operator's objective is to track the ground speed in order to perform high-precision vertical take-off and landing patterns. The first property guarantees that the operator would be capable of obtaining the necessary agility without leaving the allocated control inceptor region. The second property ensures that even in the presence of headwind, the blending would never engage the transition command variables.

When performing the LTU engagement and disengagement process, from the underlying transition conditions of the State Machine, found in Equations 4.10 and 4.14, it is implied that the throttle lever is in the gate position $\delta_{T,G}$. The transition process requires an acceleration beyond V_{STALL} but not beyond V_{LSNE} . This was discussed in Section 4.3.1.1. Similarly, the retransition process required the deceleration to the same region as explained in Section 4.3.1.2. From this, the requirement for the commanded traction thrust at the $\delta_{T,G}$ can be derived. Namely, at $\delta_{T,G}$ the commanded thrust shall be equivalent to the thrust that ensures a steady-state straight and level flight with an airspeed of $V_{CAS} \in [V_{STALL}, V_{LSNE}]$. The mapping shall be robust with relation to the allowed external disturbance margins and also high-lift system configuration.

It must be noted that in the Fallback system the throttle commanded is solely a feedforward of the inceptor command. The lack of closed-loop feedback of the kinematic or aerodynamic speed is omitted in order to increase the system robustness in the presence of faults. Therefore, the speed V_{HOVER} cannot exactly be matched in all situations. Due to the nature of the commanded variable - the throttle of the traction system - the achieved velocity is a function of the aircraft parameters and uncertainties. It may vary due to deviations between actual and assumed traction unit efficiency, aircraft drag and others. However, the conservative requirement formulation allows for facilitation of the necessary command variable selection. As seen later in this chapter, this mapping in addition aids the harmonization of the Nominal and Fallback transition and retransition procedure in the scenarios where a takeover is initiated.

4.3.3 State Allocation to Flight Phases

The relationship between state-space of automation and the aircraft flight phases was indirectly discussed with the introduction of the transition and retransition chain of causal events in Section 4.3.1. For the sake of completeness, this section assigns the state constellations of M_{FB} to the aircraft flight phases they are meant to operate in.

When observing the transition and retransition procedures in Sections 4.3.1.1 and 4.3.1.2, the pilot is expected to accelerate to the LTU disengagement speed and decelerate to hover with utilized powered-lift system in steps t1-t3) and r7-r9) respectively. Following the explanation in the corresponding sections, it is evident that this is performed in the state tuple $\{Hover, unused\}$. From this it can be derived that this state is used in the *HV* and *TR* flight phases. The flight phases definitions were introduced previously with Section 2.4.2.

In step t4), of the process flow in Section 4.3.1.1 the transition function is triggered, leading to the state constellation of $\{Wingborne, HoverDisarmed\}$. It persists for the whole duration of the procedure and subsequently in wingborne flight. Because of this, the tuple can be assigned to the *TR* and *WB* flight phases. The entry into higher airspeed is in the responsibility of the operator, therefore no other differentiation is necessary.

Table 4.6: *State Machine State to Flight Phase Allocation*

$S_{FB 1}$	$S_{FB 2}$	Flight Phase		
		HV	TR	WB
<i>Hover</i>	<i>unused</i>	✓	✓	
<i>Wingborne</i>	<i>HoverDisarmed</i>		✓	✓
	<i>HoverArmed</i>		✓	

The findings are included in Table 4.6 and also denoted with green in the state machine graphical representation, found in Figures 4.1.

4.3.4 What-If Analysis

In the previous sections, the behavior of the system automation during the transition and retransition was examined. The way the design facilitates the process flow and interacts with the surrounding systems was discussed. This section analyzes the behavior of the design in off-nominal scenarios that are not covered by the mitigation strategies. This section is structured as follows.

Previously, it has been assumed that the crew executes the procedures exactly as prescribed. Section 4.3.4.1 examines how the automation would react if the crew actions deviate from the specification. In order to ensure maximum operator authority, the automation by implication cannot include protections against dangerous crew commands. This section observes the design and analyses which combination of flight conditions and procedural deviations may lead to hazardous situations.

Afterwards the automated response in the event of faults of different surrounding components is analyzed. The major findings are covered in Section 4.3.4.2.

4.3.4.1 Procedural Deviations

This section focuses solely on the deviations from the procedures that would lead to hazards. The exact crew actions that would cause this are explained. First, the transition procedure and the mitigation actions are examined. This is followed by the retransition and its mitigation strategies.

Deviations during Transition

With relation to the State Machine transitions, there is no state change prior to the LTU disengagement. However, when observing the LTU disengagement process, the first identifiable hazard is if a deactivation of the powered-lift system below the stall speed of the aircraft is demanded. This would cause the aircraft to dive. The severity of the dive depends on the lift deficiency due to the no longer engaged LTUs and the degree of criticality depends on the distance from the ground.

When examining the events that need to occur for the hazard to manifest, the transition condition of Equation 4.10 and its atomic elements need to be examined. From the Expression 4.6 it is visible that in order for the state transition to be in effect, the operator needs to have deflected the control inceptor to the wingborne segment. In addition, a command via a discrete input needs to be communicated to the automation. According to the procedure in Section 4.3.1.1, the movement of the control inceptor in the above-mentioned range must only occur when the stall speed has been exceeded. The movement itself is along the gate, which by itself is tactile cue for the upcoming reconfiguration and hence needs to be purposelessly demanded. It can be concluded that in order for the hazard of above to occur, the pilot must execute at least two operational errors. Namely, the premature movement of the inceptor to the wingborne region and the premature disengagement command.²

Another scenario is the aircraft acceleration beyond the never exceed speed V_{LSNE} with a powered-lift system operational. From the requirements set on the thrust allocation in Section 4.3.2, at the gate position the throttle is such that this speed cannot be exceeded unless severe descent rates are commanded. By implication, this means that the operator must have commanded a lever deflection beyond the gate position. This again implies two operational errors. Prior to the lever deflection, the disengagement was not triggered and the operator has not ensured the deactivation of the powered-lift system.³

Alternatively, it could be that a failure of an LTU is in effect, because of which it is incapable of coming to a halt. The reasons for such failures are mentioned in Section 2.3.1. This scenario implicates both a hardware malfunction and a procedural deviation. The latter is evident when looking at steps t8) and t8a) that deal with the check for correctness of disengagement. The operator is required to perform the check and in the event of the above-mentioned failure, executes the mitigation strategy.

²One solution to this hazard may be to prohibit the deactivation of the powered-lift system below the stall speed. Apart from increasing the dependency on the airspeed, this strategy carries one big disadvantage. Namely, the pilot deviates from the prescribed procedures twice, on one occasion even ignoring the strong tactile cue of the lateral lever movement. It is highly-likely that this dive is actively pursued. Including the prohibition, on the one hand, limits the operator authority to conduct this dive. On the other hand, this introduces Literalism to the automation design.

³Here it must be noted that this hazard in particular is further minimized by the introduction of the barrier functions. They are explained in Section 4.4.1.

The last potential hazard is exceeding the airspeed V_{FE} with a non-retracted high-lift system, after which structural damage ensues. From Section 4.2.4 it is evident that the protection function would prohibit this. The cases where the function would be unavailable is if the Fallback system has registered a failure in the airdata system. However, in Section 4.2.4 it is explained that this function unavailability is indicated to the pilot. In addition, according to the procedure definition 4.3.1.1, a throttle lever deflection out of the gate should be performed only when the high-lift system is retracted. This implies that the operator has both neglected the supplied warning and deviated from the procedures.

Deviations during Retransition

This analysis examines the retransition flow of Section 4.3.1.2 and identifies operator commands that lead to adverse behavior. The first of the series of events is the deployment of the flaps. According to the procedure, this needs to be performed below V_{FE} . In order for this hazard to occur, the deployment should initiate at higher dynamics pressures. Therefore, the protection function must be unavailable. The conditions for which this is in effect are described in the previous paragraph. In addition, deployment must be demanded by the operator. This implies that the pilot must neglect the warnings on the cockpit indications with regards to the function unavailability. In addition, a deviation from the prescribed procedure has to occur.

The next potential hazard is the arming or the activation of the powered-lift system above the never exceed speed V_{LSNE} . According to the procedures and the underlying automation, the utilization of the LTUs by the law can only follow the arming sequence. Therefore, solely the arming process at speeds, higher than V_{LSNE} , is examined.

The arming of the powered-lift system requires a state transition of M_{FB} , namely to $\{Wingborne, HoverArmed\}$. Therefore, the transition condition of Equation 4.12 needs to hold. Analyzing the processing of the Decision-Atomics in Equation 4.7 it follows that firstly the control inceptor must be in the transition segment. Secondly, the operator must demand the arming process by means of a discrete input. By its definition, the retransition procedure in Section 4.3.1.2 requires the movement of the inceptor along the gate into the transition segment only if the airspeed is below V_{LSNE} . Thus, in addition to this procedural deviation, a significant tactile cue needs to be ignored prior to the discrete input.

The next applicable hazard is stalling the aircraft. This can only occur if the powered-lift system is disengaged while decelerating past the stall speed. From the procedure definition in Section 4.3.1.2 it is visible that the control inceptor may not be moved out of the gate in pursuit of lower throttle commands without previously engaging the Fallback

hover control mode. At the gate, thrust levels that cause the aircraft to stall cannot be commanded in the nominal case as visible in Section 4.3.2. Therefore, for this scenario to occur, multiple violations in the retransition procedure must have been performed.⁴

The last observation deals with the event, in which the arming of the Fallback hover mode is performed by the operator correctly and in accordance with the procedure in Section 4.3.1.2. However, multiple LTUs fail to engage, such that the aircraft will be rendered uncontrollable solely with the powered-lift at low dynamic pressures. This implies two mishaps. Firstly, the error itself needs to occur. In Section 2.3.1 it is argued that such a scenario is highly unlikely due to the inevitable catastrophic event in pure hover flight. In addition, the pilot has to have neglected to perform the check of powered-lift activation in steps r6) and r6a).

This section presented the occurrence of failures and operator mishaps during the transition. In the analysis performed, it is visible that in order for hazards to occur on the aircraft-level in the fault free case, multiple inconsistencies in the procedure execution need to be performed. Component faults do not lead to adverse effects by themselves and instead require at least one additional deviation from the prescribed procedures. In the next section, the component malfunctions that are not directly tied to the procedures are studied and their effect on the Fallback system operation is discussed.

4.3.4.2 Reaction to Faults

This section covers the system response to component malfunctions that are not depicted in the transition and retransition procedures, discussed in Section 4.3.1. Due to the scope of this thesis, the focus is on the automation reaction. However, for the sake of clarity, the Fallback system behavior is discussed shortly as well.

The first fault is the failure of an LTU at any point during operation that is not depicted in the derived procedures in Section 4.3.1. In contrast to the automation of the Nominal system, the failure has no implication on the execution of the Fallback automation functions. Instead, the failure is registered and is forwarded to both control allocation and cockpit indications. The former adapts the algorithms accordingly to facilitate better pseudocontrol generation. The latter serves as an alert of the operator of the decrease of performance in powered-lift flight.

The responsibility of the automation and the system reaction to a failure in a control surface is equivalent to an LTU malfunction. The reason for this is that similar to the vertical propulsion system, the control surface actuation system must be fail-active as well. Therefore, the operator is notified of the performance loss in wingborne flight and the control allocation is informed of the failed effector for proper force and moment distribution.

⁴In addition, this scenario is mitigated further by the introduction of the barrier operation. This is introduced later in Section 4.4.

Should a failure in a TTU occur, the design of the logics must not perform other actions, apart from forwarding the information to cockpit indications, control law and control allocation. The former informs the operator of the one engine inoperative scenario. The control concept is responsible for handling the failure via appropriate command variable mappings and allocation reconfiguration. This is not in the scope of this thesis.

The response to a malfunction in the high-lift system is highly dependent on the failure mode. However, the responsibility of a proper reaction to the error is allocated to the operator. If the flaps are stuck at a non-retracted state or if a hardover towards full deployment is experienced, then the pilot must not exceed the structural limit speed V_{FE} . Therefore, the system must forward this failure to the cockpit indications so as awareness of the limitation is gained.

In the failure cases where full high-lift system retraction is exhibited, then the disengagement of the powered-lift system without altitude loss must be after V_{STALL} . As seen from Section 4.3.1.1, this has no implication on the transition procedure, because this is the nominal condition for performing the deactivation. In order to facilitate this at the gate position as required by the procedures, this must be accounted for in the throttle mapping requirements of Section 4.3.2. If necessary, then the state of the configuration needs to be forwarded to the control algorithms, as discussed in Section 4.2.4.

The last dependency is that the Fallback system has a loss of sensor information. Data about the aircraft rotational rates, translational acceleration and attitude are necessary as discussed in Section 2.4.1.3. They stem from inertial sensors and are thus highly reliable. Nevertheless, should such a loss occur, then certain resilience can be achieved by degrading the control mode. Thereby, the control mode can tolerate a loss of both translational acceleration and attitude information. The consequences are modification of the transition and retransition strategies. This, however, is not in the scope of this thesis but rather a topic of further publications.

In the cases of airdata loss, the dynamic pressure is not known to the Fallback system. Therefore, the reconfiguration from *HV* and *TR* control mode and back is enabled solely via the control inceptor as seen in Section 4.2.4. The airspeed knowledge is critical to the pilot in order to conduct the transition and retransition procedures. Therefore, the likelihood of this failure occurring is deemed as low.

In terms of input items which exclude inceptor deflections, the used pilot inputs include *flaps_{DOWN}*, *flaps_{UP}* and *OPEN_{GATE}*. A failure in the former two would make the capability to operate the flaps by the operator impossible. A fault in the latter would prohibit the correct transition and retransition procedures. It must be noted, however, that all above-mentioned failures including the failures in the input items come in addition to a malfunction in the Nominal system. Otherwise, the operation of the Fallback would not have been necessary. Arguably, a takeover may be induced due to an error in an effector that would induce a transient large enough for the functional monitor to raise a false positive. This, however, is not the case for the input items. It can therefore be

concluded that a fault in the operator inputs would only be hazardous in combination with an unrelated Nominal system malfunction, which is at least a double failure and therefore very unlikely. Nevertheless, it is advisable to mitigate latent errors via a BIT [124] prior to operating the aircraft to further reduce the probability of the occurrence of this hazard.

Lastly, it is visible that the automation of the Fallback system is highly dependent on the control inceptor. However, in the design of the automation in Section 4.2, no reaction is present for the failure of the pilot sticks. The reason for this is that in the event of control input loss, the system is rendered uncontrollable and is thus this scenario is classified as catastrophic. Therefore, in the design it is assumed that the control inputs are realized in a redundant manner so that this failure mode can never occur. The automation always receives valid operator input information from the signal integrity checking of the Fallback system.

4.4 Nominal and Fallback System Integration in the Aircraft Operation

The previous sections of this chapter deal with the design of the Fallback automation functions. The solution is analyzed and procedures are derived, with which the transition and retransition from powered-lift to wingborne flight could be performed. The system robustness with relation to component errors and operational errors is analyzed.

In addition to the properties analyzed in the previous sections, one of the major motivations for the existence of the Fallback system is the necessity to takeover from a potentially erroneous Nominal system. Nominal and Fallback system together form the Flight Control System of the aircraft. This section presents the characteristics that both Nominal and Fallback system need to exhibit so as to ensure adequate aircraft operation during transition and retransition. More specifically, from the perspective of the pilot, the handling of both needs to exhibit a consistent behavior, especially in the event of contingencies. In addition, this chapter discusses the additional automation functions in terms of haptic feedback that are necessary in order to ensure safe operation. Lastly, a discussion as to how the transition and the retransition can be fit within the mission profile, imposed by the regulatory organs, is necessary.

This section is structured as follows. The control inceptor includes barriers that can restrict the access of certain throttle lever regions. Until this point in this thesis, the conditions, under which opening or closing of the barriers is required was not discussed. In Section 2.4.3.1 the operation of the barrier elements is presented. The barriers add an additional layer of operator awareness as to the allowed actions and available flight phases with the different automation modes and thus aid in enforcement of the operational procedures of both Nominal and Backup system.

In Section 3.4 of Chapter 3, certain logical criteria with relation to operator actions was defined for the Nominal system. However, a clear link to the available input items in 2.4.3 is not established. The same statement holds for the Fallback System, where such criteria are introduced in Section 4.2.1. Section 4.4.2 summarizes the not yet defined variables and derives the link between the logical expressions of the Decision-Atomics modules of Nominal and Fallback automation and the operator input items.

This derivation achieves one of the properties of **Contribution 2**, namely the harmonization of the procedures. A comparison between Nominal and Fallback operation during transition and retransition is performed in Section 4.4.3, demonstrating the above-mentioned property. This analysis is supplemented by the one in Section 4.4.4, in which the correctness of the Fallback behavior following a takeover from the Nominal system is demonstrated.

Lastly, Section 4.4.5 analyses the requirements, imposed by the regulatory organs in terms of mission profile. In the section it is demonstrated that the transition and retransition procedures comply with the demands of the MOC SC-VTOL and AS94900A and therefore their applicability in the aircraft Concept of Operations.

4.4.1 Control Inceptor Barrier Operation

In addition to the gate, Section 2.4.3.1 introduced tactile cues at the entry/exit points of the gate, referred to as barrier. These input items are capable of restricting the movement of the control inceptor, not allowing the operator to cross to potentially harmful throttle settings. The operation of the barrier functions is therefore utilized during the transition and retransition procedures in order to ensure increased awareness of the state of automation. This section introduces the logical relationships that deal with the operation of the two barriers. The motivation behind the choice of logic is discussed shortly. A comprehensive analysis of the behavior in the scope of the whole aircraft operation is presented in Section 4.4.3.

Firstly, the behavior of the two barriers with relation to the input item $OPEN_{GATE}$ is discussed. It must be noted that via $OPEN_{GATE}$, the operator forces the barriers to lift regardless of the underlying automation states. This was previously mentioned in Section 2.4.3.1. This action is performed at the control inceptor in a mechanical or electromechanical manner. Hence, it is not in the scope of this thesis because it requires no automation design considerations. However, this information is necessary in order to understand the mechanics of the transition and retransition procedures. Therefore, it is considered in this section.

When observing the upper barrier placement on the control inceptor in Figure 2.11 in Section 2.4.3.1, it is visible that it is placed at the upper end of the inceptor gate. According to the procedures of this chapter and Chapter 3, it is concluded that the purpose of the upper barrier limits the possibility to command higher airspeed or throttle in Nominal and Fallback system respectively whenever the powered-lift system is engaged.

From here the behavior of the gate depending on the system in command can be derived. In the case of Nominal system operation, the powered-lift system confirmation of disengagement occurs whenever $s_{LTU} == Disengaged$. As previously discussed in Section 3.5.1.5 of Chapter 3, in the event of a transition malfunction the crew may require to perform a flight at a higher airspeed than V_{SAFE} at a more aerodynamically efficient configuration.⁵ This is performed via the $HL_{override_{rqst}}$ input item that causes the transition of s_{HL} to *Retract*. Therefore, an additional condition to lift the upper barrier is for the *Retract* to be reached. For the case of Nominal system operation, the upper barrier is opened whenever the boolean function

$$t_{open} = (s_{LTU} == Disengaged) \vee (s_{HL} == Retract) \quad (4.27)$$

is *true*. The former relational operator is fulfilled under normal conditions and the latter is necessary for the abnormal scenario, in which prolonged flight at higher airspeed is necessary. To close the upper barrier, the boolean expression

$$t_{close} = (s_{LTU} == Engaging) \vee (s_{LTU} == Engaged) \quad (4.28)$$

needs to be *true*. Thus, using Equation 2.9, the upper barrier behavior is dictated according to

$$OPEN_{up|NOM} = Latch(t_{open}, t_{close}). \quad (4.29)$$

When the Fallback system is in command, the powered-lift system is not utilized by the control allocation whenever the wingborne control mode is engaged. This was explained in Section 4.2.4. Therefore, when the Fallback system is engaged, then the upper barrier is opened whenever

$$OPEN_{up|FB} = (s_{FB|1} == Wingborne) \quad (4.30)$$

is *true*. Even though in the state *HoverArmed* the LTUs are actively commanded to idle RPM, the status of the upper barrier does not require knowledge of $s_{FB|2}$. Section 4.4.3 discusses why no differentiation between *HoverArmed* and *HoverDisarmed* is required.

As visible from Figure 2.11 in Section 2.4.3.1, the lower barrier can restrict the possibility of commanding lower throttle settings and airspeed demands. Lifting of the barrier must be performed if the powered-lift system is fully utilized. In addition, for the cases where an LTU malfunction was registered by the Nominal system, then a confirmation of the degraded transition and hover control mode is necessary as discussed in Section 3.5.1.3 of Chapter 3. In both scenarios the Nominal system automation state s_{LTU} is the same. Therefore, the lower barrier is opened whenever

$$OPEN_{low|NOM} = (s_{LTU} == Engaged) \quad (4.31)$$

is *true*. Analogously, if the Fallback system is in command, then the barrier is open for the *true* evaluation of

$$OPEN_{low|FB} = (s_{FB|1} == Hover). \quad (4.32)$$

Table 4.7: *Barrier Status Truth Table*

$OPEN_{GATE}$	$FallbackEngaged$	$NominalEngaged$	$OPEN_{up}$	$OPEN_{low}$
<i>true</i>	-	-	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	-	$OPEN_{up FB}$	$OPEN_{low FB}$
<i>false</i>	<i>false</i>	<i>true</i>	$OPEN_{up NOM}$	$OPEN_{low NOM}$
<i>false</i>	<i>false</i>	<i>false</i>	<i>default</i>	<i>default</i>

The derivations of this section are summarized in Table 4.7. The system in command is expressed using the variables $FallbackEngaged \in \mathbb{B}$ and $NominalEngaged \in \mathbb{B}$. The command selection is not in the scope of this thesis. It is visible that a prioritization is in effect. The highest priority is the $OPEN_{GATE}$ input item. If not requested by the crew, then the Fallback system is taken, provided it is in command. The Nominal system receives a lower prioritization in order to facilitate the correct selection of the barrier status in the event of a takeover with the Fallback system. The last row of the table implies that no system is in command. There, the barriers shall assume their default status. Because neither automation function can influence the status, this is not in the scope of this thesis and left undefined.

It must be noted that the communication to the inceptor was not mentioned in both Nominal and Fallback Decision-Atomics descriptions. This is done intentionally, as it is a design decision whether the computation of the status of the gates is performed decentralized on Nominal and Fallback system and forwarded to the control inceptor or performed centralized on the inceptor itself. The latter involves the forwarding of the required states that are visible in the equations above. In the former, solely the results of the equations need to be sent. These considerations were omitted in the corresponding sections of the Nominal and Fallback software design for the sake of readability.

4.4.2 Completing the Human-Machine-Interface Processing of both Nominal and Fallback System

Section 2.4.3.1 provides an overview of the throttle control inceptor and the $OPEN_{GATE}$ input item, responsible for lifting the two gate barriers. The previous section demonstrates how the usage of this item in combination with the automation status of both Fallback and Nominal system can facilitate the complete operation of the barriers.

⁵Chapter 3 demonstrates that during this state the flaps are retracted to minimize the drag. In addition, the airspeed protections enforce a safe flight envelope.

However, the management of the automation includes the introduction of variables, the origin of which was not explained fully. In the Decision-Atomics modules of both systems, found in Sections 3.4.1.1 and 3.4.2.1 for the Nominal system and Section 4.2.1 for the Fallback system, the variables are listed. They are introduced here due to the importance of this item for the procedure harmonization.

For the sake of readability, first a short summary of the above-mentioned variables is provided. Afterwards, the HMI processing supplementation of the Decision-Atomics of the two automation modules is provided. The motivation behind the choice of processing is shortly elaborated. In Section 4.4.3 a comprehensive analysis of the harmonization of the procedures including the input items is provided.

The automation of the Nominal system includes two variables, with which the procedure harmonization is facilitated. With $LTUoverride_{rqst}$ the operator communicates to the automation that entry into powered-lift mode is acceptable. This is necessary for the detected degradation of powered-lift flight due to a fault in the LTUs. Situational awareness is ensured because the operator is forced to acknowledge the upcoming degraded aircraft performance via the actions in $LTUoverride_{rqst}$. This variable is an input symbol of M_{LTU} and is used in the transition condition t_7 in Equation 3.31.

The next variable in the Nominal system Decision-Atomics is $HLoverride_{rqst}$. This is an input symbol of M_{HL} and used in the abnormal event that the deactivation of the powered-lift system fails to succeed. Via the actions, summarized in this variable, the pilot communicates to the system that the abnormal event is acknowledged. $HLoverride_{rqst}$ is used in transition condition t_9 in Equation 3.50 and subsequently performs the retraction of the high-lift system to ensure a more aerodynamically efficient flight condition.

The automation of the Fallback system utilizes four variables for the operation of the State Machine with the help of which procedure harmonization is ensured. The Decision-Atomics of M_{FB} processes the actions of the pilot and generates $shutdown_{rqst}$ that communicates to the system that wingborne flight needs to be entered. Therefore, the wingborne control mode of the Fallback system must be engaged and the powered-lift system is disengaged. The input item is used in combination with the throttle lever position to trigger the state transition as visible in Equation 4.6.

The actions of the operator, summarized in $LTUengage_{rqst}$, communicate the start of the retransition procedure. The system begins the activation of the powered-lift system whenever this action is processed in combination with the inceptor deflection and causes the transition condition t_2 of M_{FB} to be *true*, triggering a state transition.

The operator actions are used for the retransition mitigation strategy to wingborne flight. $LTUengage_{withdraw}$ is processed by the Decision-Atomics and communicates to the Fallback system that the activation of the LTUs (and the arming of the Fallback hover mode) is canceled. The aircraft then reverts back to wingborne flight. This variable is used in Equation 4.8 and is directly linked to the transition condition t_4 .

Lastly, the pilot uses $hover_{rqst}$ to confirm the entry into powered-lift flight and thereby activate the Fallback hover control mode. This is used by M_{FB} in transition condition t_3 .

It is visible that all conditions of the Fallback State Machine are related to operator actions, subject to procedure harmonization. This is intentional because as previously mentioned in this chapter in Section 4.1, the Nominal system is the control concept, intended for the aircraft operation. The Fallback system is used in the abnormal scenarios, where a Nominal system fault is detected and thus a safe takeover and landing must be ensured. As a consequence, the Fallback system must be tailored in a way that conforms with the intended Nominal system operation. Thus, the majority of the requirements on procedure harmonization fall onto the Fallback system.

A significant increase of operational complexity would be introduced if each of the actions were communicated to the automation modules via a separate input item. In addition, this would impose multiple requirements on the HMI in terms of cockpit layout so as to ensure adequate usability of the system.

In order to mitigate this and facilitate a harmonization of the operation during transition and retransition of Nominal and Fallback system, the solution to generate the above-mentioned six boolean variables utilizes solely the input item $OPEN_{GATE}$ and the control inceptor position.

Firstly, the relationship

$$\underbrace{shutdown_{rqst}}_{\text{Fallback Decision-Atomics}} = \overbrace{HL_{override_{rqst}}}^{\text{Nominal Decision-Atomics}} = OPEN_{GATE} \quad (4.33)$$

is introduced. For recollection, $shutdown_{rqst}$ in combination with the throttle inceptor position at the left region of the gate activates the Fallback wingborne control mode. This allows the subsequent entry to higher throttle commands as evident from Equation 4.30. Similarly, $HL_{override_{rqst}}$ at the same throttle position induces a retraction of the flaps in the Nominal system and also allows the command of higher airspeed demands. Thus, whenever the pilot engages the $OPEN_{GATE}$ input item, the upper barrier is lifted. Upon releasing $OPEN_{GATE}$, the upper barrier would remain open regardless of the system that is currently operating the aircraft.

The next two dependencies are

$$\underbrace{LTUengage_{rqst}}_{\text{Fallback Decision-Atomics}} = OPEN_{GATE} \quad (4.34)$$

and

$$\underbrace{hover_{rqst}}_{\text{Fallback Decision-Atomics}} = \overbrace{LTUoverride_{rqst}}^{\text{Nominal Decision-Atomics}} = OPEN_{GATE} \wedge \chi_{\mathbb{T} \setminus \mathbb{R}}(\delta_T). \quad (4.35)$$

For recollection, whenever in the Fallback system wingborne mode the lower barrier would be closed according to Equation 4.32. It would be opened when $OPEN_{GATE}$ is commanded by the pilot. According to Equation 4.34 of above, the conditions to arm the powered-lift

system would be fulfilled as well. In order to engage the Fallback system hover mode, according to Equation 4.35, the operator would subsequently need to move the throttle out of the gate into the transition region. The second equation also depicts the condition to engage the Nominal law's transition mode. Prior to this, according to Equation 4.31, the lower barrier is closed. Due to $OPEN_{GATE}$, the barrier is opened and crossing it both allows the usage of the distributed propulsion by the control allocation and permits the command of lower airspeeds by the operator.

Lastly, in the Fallback mode, the reversion back to wingborne flight for the case of faults in the powered-lift system is performed using Equation 4.8. This involves either moving the inceptor in the left portion of the gate or when remaining in the right portion of the gate and executing

$$\underbrace{LTUengage}_{\text{Fallback Decision-Atomics}}_{\text{withdraw}} = EdgeDecrease(OPEN_{GATE}). \quad (4.36)$$

From the equation above, the Fallback system would in turn transition into the *HoverDisarmed* and according to Table 4.7, the lower barrier closes, not allowing to enter low throttle commands in the Fallback wingborne mode. The motivation behind explicitly using an edge detector is motivated by the takeover conditions and the aircraft reaction following this takeover. This is analyzed in more detail later on in Section 4.4.4.

This section provided the remainder of the HMI processing for both Nominal and Fallback system. The computed variables are necessary for the harmonization of the transition and retransition procedures. In the next section, the complete procedures are examined. References to the operator actions that trigger those variables are made.

4.4.3 Nominal and Fallback Transition and Retransition Comparison

Achieving a harmonized transition and retransition between Nominal and Fallback system allows for the execution of the reconfiguration in a similar fashion. This reduces the operational complexity and the amount of training required for the pilot to become familiar with the system and its underlying control concept. In order to facilitate a similar strategy for both control systems, three aspects need to be accounted for.

The first property is to guarantee coherency of the control concept. More precisely, the variables, commanded via the control inceptors, need to be consistent. This is presented previously in Sections 2.4.1 and is not in the scope of this thesis.

Secondly, the feedback of the automation to the pilot needs to be accounted for. This aspect is divided into the consistent operation of the cockpit indications and the haptic feedback. The design of the indication item color pattern is not in the scope of this thesis. Therefore, topics such as human factors, operator perception, quick and concise indication interpretation are not covered here. However, this thesis derives the necessary information supply so that full operator awareness with regards to the state of transition

and retransition can be ensured. This was performed in Sections 4.2.4 and 3.4.4.4. The haptic feedback consistency is related to the operation of the barriers and is found in Section 2.4.3.1. Their operation was derived in Section 4.4.1.

Lastly, the procedures during transition and retransition need to trigger comparable sequences of events. This facilitates a consistent operator mental image as to the state of the aircraft and the underlying automation. It aids in the decision-making process in the event of mitigation strategies. Furthermore, in the event of a Takeover by the Fallback system, the need for adaptation of the pilot to the new control mode and the continuation of the procedure is reduced. Lastly, having consistency in the mitigation strategies reduces the duration of the operator decision-making process and aids in the correctness of the execution.

This section analyzes the transition and retransition procedures for both Nominal and Fallback systems. It observes the exact pilot actions and tasks during the execution of the procedures and the system response in terms of information supply and haptic feedback. The focus is on the differences in behavior between Nominal and Fallback systems. The details of the automation mechanics for each system were discussed at length previously in this chapter for the Fallback system and in Chapter 3 for the Nominal system. Therefore, this section demonstrates the harmonized operation of the two systems during the reconfiguration from powered-lift to wingborne flight and back.

Firstly, the normal transition and retransition procedures are examined in Sections 4.4.3.1 and 4.4.3.2 respectively. In the section, the fault-free case is studied and it is demonstrated that the sequence of events on an aircraft-level is the same, whereby the manual effort is increased for the Fallback system. Next, in Sections 4.4.3.3 and 4.4.3.4 a comparison is made as to the operator actions and the system behavior when a mitigation is in effect.

4.4.3.1 Transition Comparison in the Failure-free Case

Previously, the analyses of the reconfiguration process were performed from a holistic perspective and were centered around the automation design. The analyses focus on the execution and how it facilitates the prescribed process flow. In this and the following sections, the procedures are broken down from the perspective of the operator in order to analyze the consistency of the procedures.

Table 4.8 summarizes the actions performed on system-level during the transition from powered-lift to wingborne flight. It provides an allocation of the necessary tasks among the involved parties - pilot, law (control mode), and automation. It must be noted that whenever the pilot is not directly responsible for the correct execution of the tasks, the operator is required to monitor the system. This ensures situational awareness in the event of abnormal events, such as component malfunctions, a takeover of the Fallback system and more.

Table 4.8: Comparison: Normal Transition Procedure - Fault-free Case

Action		Nominal			Fallback		
		Performed by			Performed by		
		Pilot	Law	Auto*	Pilot	Law	Auto*
1	Move Throttle in Right Gate	✓			✓		
2	Aircraft Reaches V_{STALL}		✓			✓	
3	Move Throttle in Left Gate	✓			✓		
4	Check Fixed-Wing Conditions			✓	✓		
5	Initiate LTU Disengagement			✓	✓		
6	Execute LTU Disengagement		✓	✓		✓	✓
7	Confirm High Speed Entry			✓	✓		
8	Retract High-Lift System			✓	✓		
Aircraft is in Fixed-Wing Mode of Operation.							

* Stands for “Automation”

To initiate the transition, the operator is required to deflect the throttle control inceptor to the end of the transition region. This is depicted in row one of Table 4.8. This creates an acceleration that is managed by the control mode, whereby the throttle mapping was discussed in Sections 3.4.1.1 and 4.3.2 and it is guaranteed that the attained speed would eventually be in a region, where the disengagement of the LTUs is allowed.

Whenever V_{STALL} is exceeded, the theoretically permissible shutdown region has been entered. This region is introduced previously in Section 2.4.2. Therefore, the pilot can proceed to deflect the throttle inceptor in the gate to the wingborne region.

The next step is to ensure that the disengagement of the LTUs can commence. This is done by the Nominal system automation via Equation 3.19. In the Fallback it is the pilot’s responsibility and is depicted in row four of Table 4.8. Once the conditions are met, then the disengagement starts. This is the first event during transition, in which both systems differ. In the Nominal system, this is done by the automation. In the Fallback system this is done by the pilot via the input item $OPEN_{GATE}$ that can immediately be released. Both automation modules communicate the the respective control allocation systems that the ramp down needs to execute. However, as presented in Section 4.2.4, a warning is issued in the case of flight with the Fallback system as a reminder to the operator to verify the correctness of disengagement.

To move to higher airspeed regions, the LTU disengagement needs to be confirmed as depicted in row seven of Table 4.8. Here the second difference in the procedure task allocation is noticeable. As already discussed, for the Nominal system this is managed by the automation. For the Fallback system, the pilot must ensure the correctness of the shutdown. In the Fallback, the previously issued warning can be dismissed.

Lastly, high-lift system needs to be retracted. This action is only necessary in the presence of flaps and can be omitted in the cases where the aircraft is not equipped with such a system. The modularity of both Fallback and Nominal systems with regards to the different configurations was demonstrated previously in the chapter and in Chapter 3 respectively. The management of the high-lift system is automatic for the Nominal system, whereas in the Fallback, the operator must perform the retraction manually.

When comparing the procedures during Nominal and Fallback system operation, the pilot involvement is visibly higher in the Fallback system. In particular certain responsibilities of the transition are allocated to the operator. However, the sequence of events is the same. In addition, the nature of the tasks is equivalent. The difference is that instead of the automatic execution, they are performed manually by the operator. Another difference so far not mentioned is the haptic feedback behavior. This is analyzed below.

Prior to the LTU disengagement initiation in row five of Table 4.8, the upper barrier is closed and the lower one is open. This holds for both systems according to the considerations of Section 4.4.2. Being fully automatic, during flight with the Nominal system, no input is required by the operator. Therefore, during the state transition flow from *Engaged* to *Disengaging*, the lower barrier closes, implying that both restrictions are in effect. The throttle is thereby limited only to movements in the gate. The subsequent completion of action seven in Table 4.8 implicates the transition to *Disengaged*, where the upper barrier is open. This sequence is evident when observing the haptic feedback operation in Section 4.4.2.

During flight with the Fallback system, the LTU disengagement is initiated by the operator with the use of the input item $OPEN_{GATE}$. At this moment both barriers are lifted due to the logic previously presented in Section 4.4.2. At the same time, the transition from *Hover* to *Wingborne* is triggered as evident from Equation 4.10. According to the derivations in Section 4.4.1, the subsequent release of $OPEN_{GATE}$ drops only the lower barrier.

Though the initial and final constellations of the barrier are the same, the different behavior during the transition is visible. The reason for this is the increased automation involvement in the Nominal system. This property allows for the closing of both barriers and as a consequence, the safe envelope is enforced further. Later in Section 4.4.3.3 similarities in the haptic feedback response in abnormal events is demonstrated.

It must be noted that the split between inceptor deflections from right to left regions in the gate (rows one and three of Table 4.10) depending on the airspeed can be omitted for the Nominal system. Instead, the control inceptor can be deflected in the left portion of the gate immediately. Chapter 3 demonstrates that this does not have an impact and the procedure can be conducted with no implications. The importance of this division

in two parts is in the event of a takeover. Deflecting to the wingborne region only after exceeding the stall speed mitigates the possibility of hazards due to procedure deviations following the takeover. This hazard is discussed previously in Section 4.3.4.1.

4.4.3.2 Retransition Comparison in the Failure-free Case

Similar to the transition in the previous section, this section provides a comparison of the reconfiguration process during retransition. This is summarized in Table 4.9. The table follows the same convention as Table 4.8 found in the previous section.

Table 4.9: *Comparison: Normal Retransition Procedure - Fault-free Case*

Action		Nominal			Fallback		
		Performed by			Performed by		
		Pilot	Law	Auto*	Pilot	Law	Auto*
1	Move Throttle in Left Gate	✓			✓		
2	Aircraft Reaches V_{LSNE}		✓			✓	
3	Move Throttle in Right Gate	✓			✓		
4	Deploy High-Lift System			✓	✓		
5	Initiate LTU Activation			✓	✓		
6	Execute LTU Activation		✓	✓		✓	✓
7	Confirm Powered-Lift Entry			✓	✓		
Aircraft is in Powered-Lift Mode of Operation.							

* Stands for “Automation”

Initially, wingborne flight in the clean configuration is assumed. First, the pilot is expected to deflect the control inceptor into the lower end of the wingborne regions. This implies into the left portion of the gate. This triggers the deceleration of the aircraft and once below the structural limit speed V_{LSNE} , the throttle can be deflected to the transition region (the right of the gate). These steps are depicted in rows one to three of Table 4.9.

First, the deployment of the flaps is performed. If the system is not equipped with a high-lift system, then this step can be omitted. Chapter 3 demonstrates the decoupling of the LTU and High-Lift automation for the Nominal system. The deployment is conducted manually or automatically by Nominal and Fallback systems respectively.

Upon the full extension of the flaps, the activation of the powered-lift system can commence as depicted in row five of Table 4.9. In the Nominal system, the initiation performed fully automatic, whereas the Fallback system awaits the operator confirmation with the input item $OPEN_{GATE}$ as explained in Equation 4.34.

In both systems the activation is performed via the automation that supplies the control allocation with the command for activation in both control systems. This is introduced in Chapter 3 and in this chapter in Section 4.2.4 for Nominal and Fallback systems respectively.

The correctness of LTU activation must then be confirmed. In the Nominal system this is automatic, as already discussed in Chapter 3. For the Fallback system this is done via the actions in Equation 4.35. Here, the operator is required to move the throttle out of the gate region and cross the opened barrier.

Similarly to the transition, the difference between the actions of both systems arises due to the allocation of the select activities to the pilot's responsibilities. However, as seen in Table 4.9, the sequence of actions (and therefore events) is the same. Next, the behavior of the haptic feedback is examined.

Prior to row four of Table 4.9, the upper barrier is open and the lower barrier is closed. This allows entry into the gate in the wingborne region. During the actions of rows five to seven of Table 4.9, the Nominal system's State Machine M_{LTU} transitions from *Disengaged* to *Engaging* and then to *Engaged*. According to the behavior of the barriers found in Section 4.4.1, during the first state transition, both barriers are closed, restricting the movement solely in the gate. After the second state transition, the lower barrier lifts, allowing the entry into lower airspeed following the correct powered-lift system activation.

During flight with the Fallback system, the activation of the powered-lift system is initiated by the operator with the use of the input item $OPEN_{GATE}$. At this moment both barriers are lifted due to the logic previously presented in Section 4.4.1. The state transition from *Wingborne* to *Hover* is performed after executing the actions as per Equation 4.35. According to the derivations in Section 4.4.1, the subsequent release of $OPEN_{GATE}$ drops only the upper barrier.

Similarly to the observations during transition, the initial and final constellation of the barrier is the same. However, the behavior during the retransition is different. Again, the increased automation involvement in the Nominal system allows for the closing of both barriers and therefore the further enforcement of a safe envelope. Section 4.4.3.4 demonstrates similarities in the haptic feedback response in abnormal events.

It must be noted that the split between inceptor deflections from left to right regions in the gate (rows one and three of Table 4.14) can be omitted for the Nominal system. Instead, the control inceptor can be deflected in the right portion of the gate immediately. Chapter 3 demonstrates that this does not have an impact and the procedure can be conducted with no implications. The importance of this division in two parts is in the event of a takeover. Deflecting to the transition region only when below V_{LSNE} mitigates the possibility of hazards due to procedure deviations following the takeover. This hazard is discussed previously in Section 4.3.4.1.

4.4.3.3 Transition Mitigation Actions Comparison

This section examines the application of mitigation strategies during the transition from the pilot’s perspective. The response of the Nominal system to abnormal events was discussed previously in Section 3.5.1.3. For the Fallback system, the same is available in Section 4.3.1.3. There, a thorough analysis of the system reaction is provided. Here the focus is on the differences in the procedures from the perspective of the operator and the required actions.

A high-lift system error implies that the actions in row eight of Table 4.8 cannot be performed. Otherwise, no implications from the transition procedure are evident. During flight with the Fallback system, the operator must be aware that the speed V_{FE} may not be exceeded. Similar to previous discussions, here it assumed that asymmetric hardovers are handled by other functions and are not in the scope of the considerations here. Chapter 5 provides an example of how such high-lift system errors are handled.

Other failure modes that influence the transition are failures of the LTU, such that a shutdown cannot be performed. First, the reversion to powered-lift flight is observed. The actions in the event of this decision are provided in Table 4.10. The start of the process is the same as Table 4.8, therefore the initial steps are omitted for the sake of readability. In this table and all subsequent comparison tables, the entries marked in red depict the actions that deviate from the previously introduced normal procedures. These abnormal tasks are furthermore denoted with “a” in the actions column.

Table 4.10: Comparison: Abnormal Transition Procedure - Powered-Lift Flight Reversion

Action		Nominal			Fallback		
		Performed by			Performed by		
		Pilot	Law	Auto*	Pilot	Law	Auto*
⋮							
5	Initiate LTU Disengagement			✓	✓		
6	Execute LTU Disengagement		✓	✓		✓	✓
LTU(s) do not disengage.							
a	Move Throttle in Right Gate	✓			✓		
Proceed with Retransition Procedure							

* Stands for “Automation”

In this mitigation, the crew communicates the intent to perform the reversion to powered-lift flight by deflecting the throttle control inceptor back into the transition region within the gate. From then on, the respective retransition procedures need to be performed. The automation reactions are discussed at length previously in this chapter for the Fallback

system and in Chapter 3 for the Nominal system. During flight with both systems, the lower barrier is closed, as evident by the decision-making process of the haptic feedback, found in Section 4.4.1. It is lifted in accordance to the respective procedures.

Table 4.11 provides an overview of the crew actions following the detection of the error if a manual shutdown of the LTU is possible. In this case, the crew can decide to execute this shutdown in order to fully perform the transition. This is depicted in the table and discussed next.

Table 4.11: *Comparison: Abnormal Transition Procedure - Reconfiguration to Fixed-Wing Mode*

Action		Nominal			Fallback		
		Pilot	Law	Auto*	Pilot	Law	Auto*
⋮							
5	Initiate LTU Disengagement			✓	✓		
6	Execute LTU Disengagement		✓	✓		✓	✓
LTU(s) do not disengage.							
a	Manually Shutdown LTU(s)	✓			✓		
7	Confirm High Speed Entry	✓		✓	✓		
8	Retract High-Lift System			✓	✓		
Aircraft is in Fixed-Wing Mode of Operation.							

* Stands for “Automation”

As explained in Chapter 3, a warning is issued to the crew in the event that the shutdown of the powered-lift system times out during Nominal system flight. From then on, the crew is expected to manually turn off the malfunctioning LTU. This action must be performed during flight with the Fallback system as well. Therefore the mitigation action is identical. In the subsequent step (row seven of Table 4.11), the Nominal automation is capable of confirming the entry into wingborne flight as discussed previously in Section 3.5.1.3. The added pilot action is the dismissal of the previously issued warning.

As previously discussed, if a manual shutdown of the problematic LTU is impossible, then the crew may want to continue to operate in “quasi” wingborne flight in order to cover a larger distance with an aerodynamically efficient configuration. The crew actions in this event are found in Table 4.12.

During flight with the Nominal system, the upper barrier is closed. When the warning is issued, the crew must communicate to the automation the intent of higher airspeed entry. According to Equation 4.33, this is done via the input item $OPEN_{GATE}$. This lifts the upper barrier and due to Equation 4.33 the state transition to $s_{HL} = Retract$

Table 4.12: Comparison: Abnormal Transition Procedure - Entry to Higher Airspeed

Action		Nominal			Fallback		
		Performed by			Performed by		
		Pilot	Law	Auto*	Pilot	Law	Auto*
⋮							
5	Initiate LTU Disengagement			✓	✓		
6	Execute LTU Disengagement		✓	✓		✓	✓
LTU(s) do not disengage.							
a	Confirm High Speed Entry	✓			(✓)		
8	Retract High-Lift System			✓	✓		
Aircraft can enter Higher Airspeed.							

* Stands for “Automation”

is executed as per Equation 3.50. By implication, in Section 4.4.1 it is visible that the release of $OPEN_{GATE}$ would keep the upper barrier open. Following this, the crew must dismiss the transition warning, which is contained in row “a” of Table 4.12.

In the Fallback system, the procedure is the same as the nominal procedure. This is evident when comparing the actions following the error - in order to detect the issue, the pilot has already triggered the deactivation using $OPEN_{GATE}$. By implication, the upper barrier is lifted, whereas the lower barrier is closed. It must be noted that the workload of the operator during operation with the Fallback system is increased due to the responsibility to not exceed the structural safe speed V_{LSNE} . However, the above-mentioned mitigation strategies to revert to powered-lift flight or to manually turn off the LTU are executed by the pilot identically in Fallback and Nominal system.

To enter a flight regime with higher airspeed it is visible that the crew actions in the Nominal system are the same as the ones necessary in the Fallback system. The pilot tasks after the fault also lead to a consistent upper and lower barrier behavior. The crew engages the item $OPEN_{GATE}$, at which point both gates are lifted. Then $OPEN_{GATE}$ is released, which leads to a closing only of the lower barrier.

In this section it is visible that the mitigation strategies during transition are the same for Nominal and Fallback system in terms of crew actions. Furthermore, one of the mitigation strategies - the entry to higher airspeed - is equivalent to the normal procedure for the Fallback system.

4.4.3.4 Retransition Mitigation Actions Comparison

The response of the Nominal system to abnormal events is discussed previously in Section 3.5.1.3. For the Fallback System, the same is available in Section 4.3.1.4. Here the retransition mitigation strategies are examined from the perspective of the operator.

A high-lift system error implies that the actions in row four of Table 4.9 cannot be performed. Otherwise, no implications from the transition procedure are evident. Similar to previous discussions, here it is assumed that asymmetric hardovers are handled by other functions and are not in the scope of the considerations here. Chapter 5 provides an example of how such high-lift system errors are handled.

Apart from the failure in the high-lift system, other relevant malfunctions that change the retransition procedure execution are faults in the powered-lift system. In such instances the mitigation strategies are either to revert back to wingborne flight or to enter powered-lift flight with an acknowledged performance reduction. The motivation behind either action depends heavily on the applicable mission profile and is discussed at length previously in this chapter for the Fallback system and in Chapter 3 for the Nominal system.

Table 4.13: *Comparison: Abnormal Retransition Procedure - Reversion to Wingborne Flight*

Action	Nominal			Fallback		
	Performed by			Performed by		
	Pilot	Law	Auto*	Pilot	Law	Auto*
5	Initiate LTU Activation		✓	✓		
6	Execute LTU Activation		✓		✓	✓
LTU(s) do not engage.						
a	Move Throttle in Left Gate	✓		✓		
Proceed with Transition Procedure						

* Stands for “Automation”

The actions in the event of a reversion to wingborne flight are depicted in Table 4.13. The start of the process is the same as Table 4.9, therefore the initial steps are omitted for the sake of readability. The issue in the powered-lift system would be detected at the latest during the LTU activation process.

As discussed previously in Section 4.4.3.4, either pilot or automation would verify the activation correctness for Nominal and Fallback system respectively. After the conclusion that the activation process does not succeed, the actions of the crew would be identical

with both Nominal and Fallback system. The pilot has to deflect the throttle control inceptor in the gate to the wingborne region. After this, the corresponding transition procedure would be re-initiated.

With regards to the haptic feedback during Nominal system operation, due to $s_{LTU} = Engaging$, both barriers are closed as evident from Section 4.4.1. This restricts the command of an airspeed that is lower than V_{SAFE} , thereby continuously enforcing a safe envelope and the operator awareness until the procedure is complete.

Table 4.14: Comparison: Abnormal Retransition Procedure - Confirm Powered-lift Flight

Action	Nominal			Fallback		
	Performed by			Performed by		
	Pilot	Law	Auto*	Pilot	Law	Auto*
⋮						
5	Initiate LTU Activation			✓		
6	Execute LTU Activation		✓	✓		✓
LTU(s) do not engage.						
a	Confirm Powered-Lift Entry	✓			✓	
Aircraft is in Powered-Lift Mode of Operation.						

* Stands for “Automation”

The remaining scenario is to continue to powered-lift flight with the reduction of performance. This is depicted in Table 4.14. Section 4.3.1.4 already discusses that for the Fallback system, the set of actions for powered-lift flight entry is the same for both normal and abnormal events. The crew must engage the Hover control mode in exactly the same manner regardless if there is fault or not. This is depicted in Table 4.14.

For the Nominal system, a crew confirmation for powered-lift entry is necessary. This is done via the actions, found in Equation 4.35. As evident in the equation, the actions are equivalent to the ones for the Fallback powered-lift mode of operation that are explained below. The subsequent dismissal of the issued warning is also the same.

This concludes the comparison of the pilot effort during the transition and retransition and their corresponding mitigation strategies. In normal conditions, the pilot workload with the Fallback is unavoidably increased. This is visible in Sections 4.4.3.1 and 4.4.3.2. This increase of workload is necessary in order to ensure the required higher control authority.

In terms of procedures, Sections 4.4.3.1 and 4.4.3.2 demonstrated that the difference between high- and low-degree of automation operational modes is the shift of the tasks from automation to pilot. The sequences of events are the same. This creates the synergy

that in the event of a takeover the shift of the pilot role from system supervisor to system operator is fluid. In addition, the consistent chain of events allows for a more efficient monitoring by the pilot prior to this takeover.

By design, the procedures support the pilot whenever abnormal events occur. Sections 4.4.3.3 and 4.4.3.4 demonstrate that for component malfunctions the required pilot actions are the same regardless of the currently engaged system. This reduces the operator workload because there is less consideration on the mechanisms to execute the mitigation strategy once it is chosen. Instead, the focus can be put on other objectives.

Under normal conditions, solely the Nominal system is intended to be flown. Therefore, the takeover can be seen as an abnormal event. By implication, this occurrence during transition or retransition needs to be studied in order to prove that the integrated system can cope with a Nominal system malfunction. In the next section such an analysis is provided.

4.4.4 Takeover Correctness

As introduced previously in Section 4.2.3, whenever a takeover is mandatory, the evaluation in Equation 4.18 ensures that the automation selects the correct initial states of M_{FB} .

When observing a takeover outside the procedures, i.e. during powered-lift or wingborne flight, then the complexity of the initialization is manageable. The correctness of the takeover in these phases can be concluded when observing the starting state selection from Table 4.3 and the state allocation of the automaton to the aircraft flight phases found in Table 4.6. Correctness of takeover during the reconfiguration from powered-lift to wingborne flight and back is critical and is therefore examined in this section.

During a transition, the Nominal system's State Machine M_{LTU} first experiences state change from *Engaged* to *Disengaging*. Section 3.4.4.1 of Chapter 3 specifies that during the state *Disengaging* a ramp down is commanded by the control allocation in order to disengage the LTUs. According to the Decision-Execution of the Fallback system found in Section 4.2.4, the same activity is conducted in M_{FB} 's the state constellation $\{Wingborne, HoverDisarmed\}$. As evident in Table 4.3, this is also the correct takeover starting state. If the takeover occurs a later point in time, then the State Machine M_{LTU} transitions to the state *Disengaged*. During this state, the aircraft is in wingborne flight and the correctness of the Fallback initialization is evident from discussion of the previous paragraph.⁶

⁶ M_{FB} does not distinguish between an LTU deactivation process and flight with deactivated LTUs. Section 4.2.4 discusses that a zero RPM command is sent following the ramp down. Therefore, the takeover when M_{LTU} is either in *Disengaging* or *Disengaged* leads to the same starting state evaluation as per Table 4.3. However, it is important that the control allocation of the Fallback also initializes correctly. If the takeover happens during the Nominal system ramp down, then the Fallback control allocation needs to continue where the Nominal system was rejected. Arguably, correct initialization of the control allocation commands following a takeover needs to be ensured regardless of the state constellation of M_{FB} . This is a task of the control allocation design and hence not in the scope of this thesis.

Prior to a takeover during a shutdown, the throttle inceptor is in the left portion in the gate. If M_{LTU} is in the state *Disengaging*, then the barriers are both closed. This is visible when observing the logical decision of the haptic feedback of Table 4.7. At the moment of takeover, the Fallback system initializes M_{FB} with $\{Wingborne, HoverDisarmed\}$. According to Table 4.7, then the upper barrier opens. The transition procedure is continued with the Fallback system as if it was conducted solely with that system. The reason for this is that the status of haptic feedback, inceptor deflection and state of M_{FB} are identical to the ones during row six of Table 4.8.

During a retransition with the Nominal system, the State Machine M_{LTU} goes through the state transition from *Disengaged* to *Engaging*. Whenever in the latter state, the control allocation of the Nominal system commands a ramp up of the LTU RPM as per Section 3.4.4.1. This is necessary in order to automatically check for activation correctness. According to the Decision-Execution of the Fallback system, found in Section 4.2.4, this action is performed whenever M_{FB} has the states $\{Wingborne, HoverArmed\}$. As evident in Table 4.3, a takeover from the Nominal system during the state *Engaging* leads to the above-mentioned state constellation of M_{FB} .

Prior to the takeover, with the Nominal system the throttle inceptor is in the right portion in the gate and the $OPEN_{GATE}$ is not utilized. Therefore, according to the logic in Table 4.7, the two barriers are both closed. At the moment of takeover, the Fallback automation assumes the states $\{Wingborne, HoverArmed\}$ and according to the same table, the upper barrier opens but the lower barrier remains closed. This is evident from the transition conditions to leave the state constellation, found in Equations 4.14 and 4.16. To execute the former would imply that powered-lift flight is entered. This would correspond to conducting the normal retransition procedure with the Fallback system as per Section 4.3.1.2. The latter abandons the retransition in the pursuit of wingborne flight. This corresponds to the execution of one of the abnormal retransition procedures, found in Section 4.3.1.4. Prior to either decision, after the takeover no action is undertaken by the system. This provides the operator with sufficient time for the assessment of the situation and subsequent decision-making.

4.4.5 Fitting the Transition and Retransition in the SC-VTOL Mission Profile

Chapter 3 demonstrates the capability and mechanics of conducting a transition and retransition with the Nominal system. In Section 4.3.1 of this chapter, the procedure of doing the same with the Fallback system is presented and in Section 4.4.3, the compatibility of both procedures is established. In Section 2.5, the requirements, set by the regulatory efforts that impact the execution of the transition and retransition and their placement in the mission profile, were summarized. This section explores the applicability of the

procedures and the underlying automation in the envisioned mission profile. The provided procedure execution serves as proof of the procedure compliance in the overall aircraft operation and regulatory demands.

The subsequent sections use the terminology that stems from the regulatory effort. The relevant terms and their interpretation were summarized in Section 2.5 of Chapter 2.

4.4.5.1 Take-Off Decision Point Selection

The requirements as to the choice of Take-Off Decision Point (TDP) are summarized in Section 2.5.2. Prior to reaching the TDP, the take-off may be aborted for a number of reasons, which include component faults that may or may not lead to a CFP, a takeover with the Fallback system and more.

The exact placement can be selected freely by the applicant. It must be kept in mind that if the mission is continued past the TDP, then during the initial acceleration, the height of 35 feet above the elevated vertiport altitude h_2 cannot be exceeded. In addition, obstacle clearance of at least 15 feet must be attained. As a consequence, the TDP height is limited to the range of [15, 35] feet above ground level or h_2 for conventional and vertical take-off use-cases respectively. For recollection, the types of take-off scenarios are mentioned in Section 2.5.3. For the sake of simplicity and in the interest of lowering the energy consumption of the powered-lift flight, in the latter case the TDP height is chosen to be 15 feet above h_2 .

4.4.5.2 Specification of the Take-Off Safety Speed

Assuming the TDP is reached and the departure must be performed, the next relevant maneuver is the acceleration to the velocity V_{TOSS} which is introduced previously in Section 2.5.3. A transition during this segment is not possible due to two reasons.

The first discussion point is whether the reconfiguration to wingborne flight can be performed in this segment. Firstly, if the transition were to be performed in the initial take-off phase, this would imply that V_{TOSS} is at least V_{STALL} . This would require a long acceleration distance and if performed in horizontal flight, this would undoubtedly be in collision with the reference volume, mentioned in Section 2.5.2. Otherwise, a climb would be necessary, therefore the range of [15, 35] feet above h_2 would be exceeded. This range is mentioned in the previous section.

The second reason to not perform a transition in the initial departure segment is that manual configuration changes are not permitted. Arguably, in normal operation, the transition with the Nominal system is performed automatically. However, after the TDP, the departure profile must be performed in the event of any single abnormal event. This includes a takeover with the Fallback system, after which the reconfiguration is manual.

This would introduce a certain degree of inconsistency in the mission profile execution because following the takeover, the transition must be executed at a different place in the take-off trajectory.

As a consequence, V_{TOSS} needs to be in the powered-lift flight phase. In order to facilitate better haptic indication as to the necessary throttle level during the acceleration, the throttle detent $\delta_{T,D}$ is used as the V_{TOSS} position. From the derivations of Chapter 3, this means that

$$V_{TOSS} = V_{HOVER}. \quad (4.37)$$

In the Fallback system, the thrust that is equivalent to that speed is mapped to the detent position as well. This was specified in Section 4.3.2.

4.4.5.3 Specification of the Final Take-Off Speed and the Transition

Once V_{TOSS} is reached, the pilot can continue the prescribed climb profile, summarized in Section 2.5.3. This must be performed with V_{TOSS} until the two hundred feet mark is reached. There, horizontal flight is permissible again and the aircraft must accelerate to the speed of V_{FTO} . Here, the transition can be conducted.

If performance of the aircraft allows it, then horizontal flight can be omitted and the transition can be performed while climbing instead. However, in the interest of battery consumption it is advisable to perform the transition during horizontal flight and thus allocate the specific excess power solely to the aircraft acceleration. A dive for transition, on the other hand, needs to be omitted so as to not penalize requirements imposed by AS94900A 2.5.1. After performing the transition, the climb to one thousand feet can be continued at a much more efficient aircraft configuration (wingborne flight). As a consequence, the transition is to be performed at two hundred feet above h_2 and V_{FTO} must be the disengagement speed at the lowest, ideally above V_{SAFE} so as to guarantee ideal obstacle clearance capabilities following the powered-lift disengagement.

4.4.5.4 Specification of the Landing Decision Point, the Retransition and the Landing Reference Speed

A summary of the approach profile is provided in Section 2.5.4. The relevant parameters to specify in the segment are the Landing Decision Point (LDP) and the velocity V_{REF} . Though not explicitly mentioned by the regulator, the execution of the retransition must be performed and as a consequence, the fit of the procedure in the mission profile must be analyzed.

As mention in Section 2.5.4, prior to reaching the LDP the approach can be rejected and a go-around must be performed. After crossing LDP, the landing must be executed. The reasons to abort the approach may be component faults that may or may not lead to a Critical Failure of Performance (CFP), a takeover with the Fallback system and more.

By implication, this means that a retransition mitigation strategy may be the reason for a balked landing. Therefore, it can be concluded that the retransition must be performed prior to reaching the LDP.

The LDP must be crossed with the speed V_{REF} . A consequence of the requirement to perform the retransition prior to the LDP is that V_{REF} is either in the transition/retransition or the hover flight phase. During a balked landing, the regulator prescribes that V_{TOSS} must be regained, after which the take-off profile needs to be initiated. Therefore, it can be concluded that the definition of

$$V_{REF} = V_{HOVER} \quad (4.38)$$

is feasible for two reasons. First, as previously stated, the speed V_{HOVER} is assigned to the throttle inceptor's detent position. Therefore, the pilot has haptic feedback of the correct throttle setting at the crossing of the LDP. Secondly, the deceleration to V_{REF} can be performed shortly before crossing the LDP and prior to that the upper airspeed region of the transition/retransition flight phase can be maintained. In addition, having a Landing Reference Speed at the border of the hover flight phase implies that the distance between LDP and landing point can be kept short. Both above mentioned characteristics reduce the energy consumption of the aircraft.

4.5 Chapter Summary

This chapter presented the Fallback system automation methods that enable the transition from powered-lift to wingborne flight and back of VTOL aircraft. The design and the resulting procedures advance the state of technology in accordance with **Contribution 2**. It accomplished the following targets.

Low-Degree of Automation Transition, Retransition and Takeover Capability

Section 4.2 presented an automation design that together with the laws in Section 2.4.1.2 can facilitate a manual transition and retransition, executed by the operator. Subsequently, Section 4.3.1 demonstrated how a reconfiguration from powered-lift to wingborne flight and back can be executed and a derivation of the transition and retransition procedures was provided. The design ensures a high degree of operator authority.

By design, resilience against component malfunctions and procedure deviations was demonstrated in Section 4.3.4. In addition, the correctness of operation following a fault of the Nominal system was shown in Section 4.4.4. The capability of the low-degree of automation system to perform fixed-wing and powered-lift configurations, its reduced sensor dependency and proper operation following a takeover from the Nominal system allow for correct execution of the fallback principle as defined in Section 1.2.4. This guarantees a fail-safe FCS operation in the event of an erroneous Nominal system.

Lack of Nominal System Implications

Section 4.4.3 demonstrated that the methods and procedures of this chapter take the Nominal system operation and its transition and retransition procedures into account. However, as visible from the above-mentioned section, the Fallback system does not impose additional requirements or restrictions on the Nominal system design and operation. Therefore, in an FCS where both Nominal and Fallback systems can be executed, all favorable Nominal system characteristics in terms of systems safety are retained.

Operator Support

Sections 4.4.2 and 4.4.3 demonstrated that the difference between high- and low-degree of automation operational modes is the shift of the tasks from automation to pilot. The behavior in both Nominal and Fallback system operation is kept consistent with relation to the operator input. In the events where mitigation strategies are required, the harmonization of the transition and retransition procedures ensures a fast and equivalent operator response regardless of the chosen contingency.

The operator awareness is ensured via adequate supply to the indication items that are the same ones as the ones for the Nominal system. This is presented in Section 4.2.4. In addition, procedure and flight-state awareness is facilitated by the utilization of the haptic feedback. This is found in 4.4.1.

Industry Compliance

Section 4.4.5 demonstrated the applicability of the procedures and therefore the underlying automation within the envisioned mission profile from the currently available regulatory effort. It showed that the transition and retransition with both Nominal and Fallback systems can be executed within the take-off and approach maneuvers of the MOC SC-VTOL and fully comply with the imposed requirements.

Chapter 5

Operational Concept Validation During Early Development Stages

Chapter 3 presented the highly-automated operational concept for the transition and retransition. The created procedures and underlying automation provide fully automatic reconfiguration capabilities in the fault-free case. In the event of failures of powered-lift and high-lift systems, the developed solution provides for a non-time critical decision-making process by the operator. At all times, the design enforced a safe flight condition.

Subsequently, Chapter 4 derived the procedures and automation concept of the Fallback system transition and retransition process. The chapter demonstrates that the Fallback automatic functions could perform a takeover from the Nominal system in the event of its failure. The approach ensures an increased operator command authority and provides manual reconfiguration capabilities. At the same time, the Fallback transition and retransition concept guarantees consistency with regards to the Nominal automation concept.

In addition to introducing the automation design, Chapter 4 demonstrates compliance with requirements on the mission profile that are imposed by the regulatory organs. It explains how Nominal and Fallback systems together enable a transition and retransition capability of the system that could fit in the envisioned mission of the MOC SC-VTOL.

The concepts in Chapters 3 and 4 provide innovative solutions to the newly emerged problems associated with lift-to-cruise eVTOL aircraft. Namely, the execution of a transition to wingborne flight and the retransition back to powered-lift flight. However, certain topics and challenges remain open. They are elaborated upon below.

First, compatibility of the above-mentioned procedures with regards to the overall aircraft operational concept is not yet studied. According to [43], the UAM mission-profile and the vehicle behavior are highly aircraft-specific. Due to the novelty of the envisioned lift-to-cruise airframes, ensuring a seamless integration of all procedures in the overall system operation is a highly non-trivial task. Having simplistic and industry-compliant transition and retransition procedures carries little benefit if during the reconfiguration

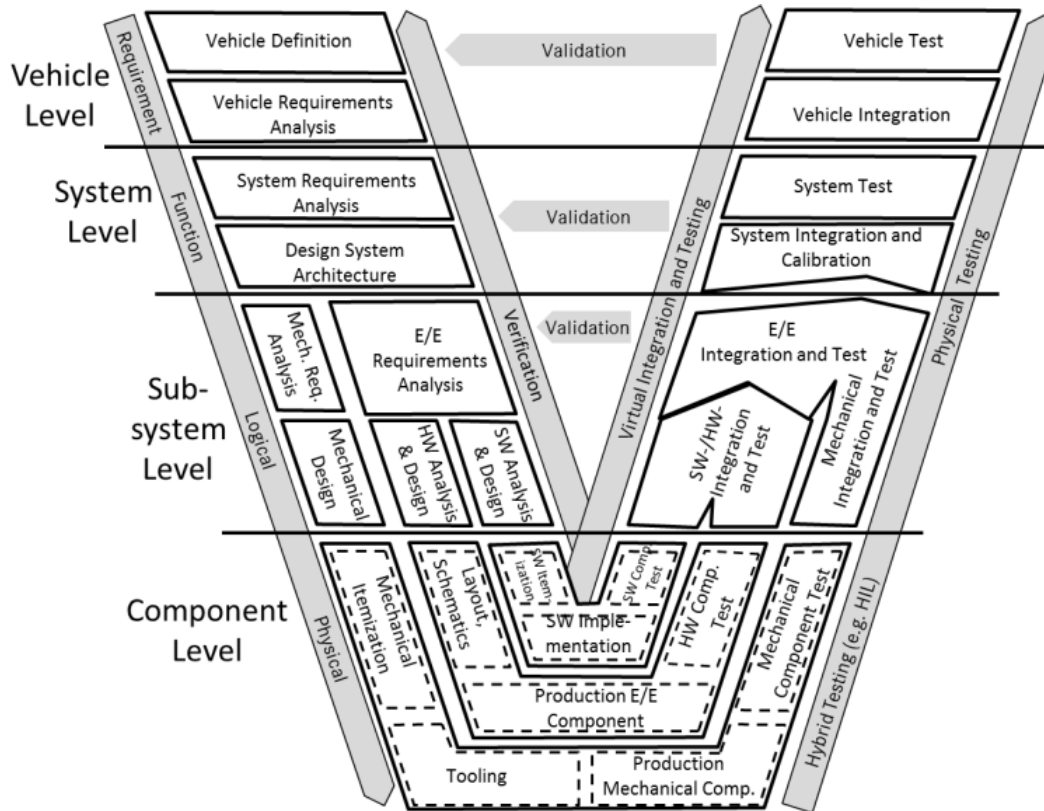


Figure 5.1: V-Model as Found in [11]

the remainder of the aircraft operation is not intuitive. Therefore, the validation of the transition and retransition procedures in the context of the whole aircraft operation is necessary.

Validation activities are prescribed in the aviation industry standards [50, 51] in accordance to the V-Model [11, 125]. An example of the V-Model is provided in Figure 5.1. From the image the importance of an adequate operational concept is visible. According to the process, the ConOps and all system-level requirements are the basis of all development activity and need to be validated using the full system integration. If the envisioned vehicle behavior is unfeasible, then this will only be detected in the final stages of the development cycle. Such a scenario implicates a change in the high-level requirements which can have a high impact on the accumulated product costs [22]. A need for validation methods in the early stages of the development is recognized in [21].

A second challenge that was not yet addressed is the correct integration of all automation functions. Similar to the discussion from above, it must be ensured that the resulting complete system automation is human-centered and that the considerations on the transparency, resilience and flexibility are not negatively impacted. As explained in the previous paragraph, on the one hand, this could be caused by having an inadequate ConOps whereby the operation of the functions is not thought through. On the other hand, such shortcomings could be driven by the interactions between the automation

functions themselves. The transition and retransition functionality is just one of the elements involved in the very complex system. As a consequence, the operator workload may be increased due to different automatic functions interfering with one-another. Mode awareness may be lost and many more. A need for an environment, in which the automation concept can be simulated and validated prior to the full functional implementation is needed.

This chapter of the thesis addresses the above-mentioned challenges and provides a method of modeling the aircraft operational concept in the beginning phases of the product life-cycle. The resulting behavioral specification model recreates the automation functions in a simplified manner and enables the simulation of the aircraft operation prior to the full functional development. Therefore, it allows for efficient validation of the aircraft ConOps and automation concept. Fast adaptations in the events where changes to the specification are necessary is possible. It thereby advances the state of technology in accordance with **Contribution 3**.

The focus of the chapter is on the demonstrating that the interactions between the derived transition and retransition functions, found in Chapters 3 and 4, and the envisioned operation of the aircraft found in Section 1.1.4 of Chapter 1 can be reproduced and thus their plausibility can be validated. The developed method offers the possibility of reproducing all aspects of the aircraft operational concept.

The chapter is structured as follows. Designing a specification model prior to the functional development implicates that a set of assumptions must be met. In addition, a degree of abstraction and simplification must be pursued so as to facilitate a low-degree of dependence of the system architecture and to allow for fast and efficient modeling. In Section 5.1 an overview of the behavioral specification design method is provided, explaining where assumptions and simplifications are met and where a high-degree of modeling fidelity is maintained.

In order to illustrate the method, a behavioral specification model for the aircraft of Section 1.1.4 of Chapter 1 is created. All subsequent sections following Section 5.1 present different aspects of the proposed methods using the behavioral specification model of this aircraft as an example. In Section 5.3, the architecture of the behavioral specification model is presented. It describes the design patterns utilized and how the previously introduced assumptions and simplifications are exactly considered into the design. It demonstrates under which conditions and circumstances the automation concepts of Chapters 3 and 4 are executed.

The behavioral specification modeling is application-driven in terms workflow and must therefore consider and incorporate a development process. Section 5.2 introduces the file structure, the utilization of different repositories and provides an overview of the development effort and workflow. In addition, aspects such as change management are discussed.

This thesis emphasizes on the automation transition and retransition process with both Nominal and Fallback systems. As a consequence, the chapter continues to present the parts of the behavioral specification model which directly concern the methods that achieve **Contribution 1** and **Contribution 2** of the thesis. In Section 5.4, the logical activities and operations that are common to both Nominal and Fallback automation are presented. Those include the processing of the operator input and the generation of the logical variables, required for the transition and retransition automation that originate from the pilot intentions. In addition, this includes the operation of the haptic feedback that was previously explained in Section 2.4.3.1.

Next the individual behavioral specification modeling of the transition and retransition automation of Nominal and Fallback system are presented. Section 5.5 shows how the high-degree of automation method of Chapter 3 is realized within the model. The management of the LTUs and the high-lift system is presented. This is followed by Section 5.6, in which the same is performed for the Fallback system methods previously derived in Chapter 4. There the focus is on the modeling of the correct initialization following a takeover from the Nominal system and on the management of the control mode.

The chapter is concluded with Section 5.7 where the achieved contributions are summarized. The section furthermore shows how the behavioral specification model for the aircraft in Section 1.1.4 of Chapter 1 is utilized within the activities of the TUM Institute of Flight System Dynamics.

5.1 Method Description

The section lists the topics considered in the developed method. As previously discussed, in order to create a high-level behavioral specification model, the proposed method must be predicated on certain simplifications and assumptions. Those are necessary due to the high-degree of aircraft abstraction in the beginning stages of the functional development but also in order to reduce the behavioral specification modeling time and effort. This section presents the task breakdown involved in the creation of the behavioral specification model and in addition defines clear objectives for the provided solution.

Section 5.1.1 lists all activities and methods that are unarguably necessary in the scope of full functional development but are omitted in the behavioral specification modeling. Clear argumentation for the reason of the elimination of these tasks is provided.

As discussed in detail in Section 2.2.1 of Chapter 2, a major design aspect of the automation functions are considerations, directly related to the system architecture and operation. Section 5.1.2 analyzes the automation activities, associated with the system design operation and provides an overview of all tasks that are reproduced within the behavioral specification model. A degree of modeling abstraction is decided on. The latter facilitates a reduced development effort. The degree of fidelity and made assumptions are listed in that section.

The level of modeling fidelity is a recurring topic in this chapter. The degree of abstraction which certain functions are developed is of significant importance. The two properties are the central element that describe how representative the resulting functions with respect to the final functional design are. Section 5.1.3 summarizes and groups the different functions according to their degree of modeling fidelity and abstraction.

5.1.1 Degree of Rapid Prototyping

Avoiding full functional development carries both benefits and shortcomings. Reducing the complexity of certain functions by assumptions and averting robustness considerations allows for a time-efficient modeling of the system response. However, minimizing the effort just for the sake of expediting results leads to the danger of oversimplification and thereby unfeasible capabilities of the function. This section lists activities that could be completely omitted without sacrificing the system response characteristics.

The developed method must pursue a high-level of abstraction but at the same time aims to guarantee a realistic behavioral modeling. All produced functions that are attributable to the FCS design must fulfill their intended operation using data that can be supplied by surrounding systems and sensors later in the actual application. However, certain robustness considerations are omitted.

Namely, measures such as anti-aliasing, filtering out process noise and others methods that require knowledge of the providing sensors' characteristics are omitted completely for the sake of simplicity. They are necessary to ensure the system's robustness but do not influence the end behavior significantly.

The integration of system architecture components involves tedious tasks that carry little benefit when it comes to the integrated system behavior. Such functionalities include considerations with respect to the operation of low-level software drivers, the transmission protocols and more. Omitting the generation of bitmasks or processing of integrity data and instead sending the raw data has no impact on the high-level system behavior. Scaling of variables that is associated with data transmission leads to precision loss but the effect is negligible for the purposes of the methods here.¹

Knowing the input data in a idealized manner implies that assumptions, such as threshold magnitudes and confirmation times, could in be avoided. However, provisions for such elements must be made and parameterized in order to enable a realistic response. Most failure detection mechanisms - wherever necessary - are avoided for the same reasons. Instead, the behavior of such functions can be assumed to be known. One exception are functional monitors. As explained later on, they are developed with a high degree of rigor and fidelity.

¹Delays due to sensor processing and transmission change the closed-loop response. However, arguments as to why this is permissible in the considerations here are available in [8].

All other known system behavior that is not attributable to the FCS operation but has an influence on it can be abstracted to the highest degree possible in order to reduce the development effort. Examples of items that fall into this category is the operation of the electrical system, the processing of cockpit items, such as button and switches and more.

5.1.2 Degree of System Architecture Independence

One big factor that drives the development effort are considerations with regards to the components involved in the control of the aircraft. Aspects such as robustness measures against sensor dirt effects are mentioned in the previous section. However, other topics imposed by the system architecture influence the aircraft operation greatly and must be included in the specification modeling. However, they can be reproduced with significant reduction of development effort and at the same time still facilitate a realistic behavioral specification. This is only possible if a certain degree of abstraction is pursued. This section summarizes these considerations.

In very early stages of the product life-cycle, it may be unknown what exact components are involved in the system architecture design. However, for the purposes of the behavioral specification, only their role needs to be known. Thereby, the developed framework can initially model the component's response and later on expand it to enable a greater fidelity. For example, it may be known that power is supplied to the avionics after an operator input via the cockpit. However, the exact mechanics of this power supply may be unclear at this stage of development. If related to the operational concept, initially provisions can be made by modeling the power supply via one input item until the electrical system is specified. Thereafter, this behavioral specification can be expanded to account that different parts of the avionic components are powered via multiple input items and that the status of the power needs to be fed back to cockpit indications. This also allows for testing that the prescribed management of the avionics by the operator is plausible and intuitive. In the cases where it is not, this method allows for efficient changes to the specification.

Another topic attributed to the system architecture that during functional development requires a great amount of resources has to do with the redundancy management. In order to guarantee fault-tolerant properties, component redundancy is introduced in the architecture of an FCS [126]. This implies that safety-critical signals, required for proper FCS execution, are available from multiple physical entities. This is necessary in order to guarantee the availability and the detection of potential failures of the data. The process of selecting a signal from redundant sources is referred to as voting [127].

Another redundancy measure is the replication of individual flight control algorithms on several physical instances [128]. This is done in order to guarantee the availability of the algorithms in the event of component failures. The replication implies that processes within

the FCS become distributed and require software provisions that introduce significant complexity to the design [129]. To reach decisions, the components of the FCS require consensus and agreement protocols [130] and need to be resilient to failures [131].

For this purpose, the framework makes the following two simplifications. Firstly, voting mechanisms are omitted completely in the behavioral specification design. The model instead relies on the raw data from the simulation and assumes that the voting - if necessary - has been performed. Whether signal redundancy is required is a product of the safety analysis and the applied voting techniques are dependent of the number of available sources. However, this is not in the scope of the behavioral model specification and is instead subject to later development effort. Wherever applicable, faults in the voting mechanisms are simulated instead.

Secondly, the behavioral specification omits the known decentralization of the flight control algorithms and instead uses centralized algorithm design. Such an algorithm prescribes how the future decentralized one is supposed to respond. Doing this, on the one hand, alleviates the necessity to know the physical allocation of the provided functions. On the other hand, algorithms, such as command selection and consensus are significantly simplified and in some instances not even necessary. As explained later, the proposed method of algorithm centralization aids in the later functional allocation to system components.

5.1.3 Simulation Capabilities and Tools

The previous sections discussed multiple simplifications that are made within the behavioral specification model. This section discusses the capabilities of the developed solution. For the sake of clarity, Table 5.1 lists the methods in the model according their level of fidelity and degree of abstraction. In addition, this section summarizes the constructs utilized in order to reproduce the envisioned aircraft mission profile and operational concept fully.

At the lowest levels of fidelity and highest degree of abstraction is the processing of the signals. Those include the error injection, checking of the integrity of the signals and the voting (or signal selection). The time necessary to detect failures relies heavily on the sensors used and the voting mechanisms depend on the criticality of a sensor error and the sensor redundancy. As previously mentioned, the voting is omitted completely. The behavioral specification model includes an error injection functionality that halts the supply of a particular signal. Depending on the type of error, the last value before injection is retained or an unfeasible value is fed to the software components. The detection is modeled by a Confirmation Counter as per Equation 2.12. Thus, the notion of sensors is omitted completely. Instead, incoming data is split into signal types (rotation rates, kinematic velocities, etc.) and their availability. The time to detect erroneous signal sources can initially be assumed and later on modified depending on the characteristics of the system.

Table 5.1: *Overview of Behavioral Specification Model Method Abstraction*

Method	Response	Fidelity	Abstraction
Data Supply	Signals are clean. Precision loss not considered. Signal selection implicitly assumed. Error Detection time (or lack thereof) is predefined. Erroneous data supply is injected.	Low	High
Signal Selection	Voting not modeled at all. Errors covered by error injection of data supply.	Low	High
Input Data Processing	Robustness measures considered for further design stages. Data processing consolidated wherever possible.	Medium	High
Electrical System	Fidelity depends on stage of development. System response modeled, but mechanisms of power supply not considered.	-	High
Component Physical Behavior	Fidelity depends on stage of development. Faults injected. System response modeled.	-	High
Command Selection	Algorithms centralized. Desired system response specified.	High	High
Control Concept	DRM method utilized. Closed-loop system response representative to end design.	High	High
Functional Monitoring	Checks equivalent to end design.	High	Low
Law Automation	Functions equivalent to end design.	High	Low

The constructs of above allow to model the behavior of the incoming data in the presence of malfunctions that lead to unavailable signal types. This enables the simulation and validation of the system response in the event of failures.

Due to the distribution and replication of the functions onto different physical flight control computers, in the actual FCS certain activities are the joint decision of all involved components. Such a function is for example the algorithm that chooses the system in command (Nominal or Fallback) or some parts of the generated indication data. Opting for modeling such distributed decision-making processes as centralized reduces the modeling effort significantly.

Thus, such FCS functionality that is either decentralized among the FCS components is reconstructed with a greater level of fidelity while achieving a very high level of modeling abstraction. The behavioral specification does not consider the number of function replications or the physical allocation to components. As mentioned earlier, however, this reduces the development effort while at the same time retaining a feasible system response.

Such an approach is pursued for so-called “shared” functions. Those are for example the aircraft behavior on ground or the processing the pilot inceptors. Such activities must be performed by all control concepts within FCS. In the behavioral specification model, these processes are consolidated.

Other methods that are of higher fidelity are ones that are explicitly necessary, but cannot be recreated realistically to their full extent due to requirement on explicit system architecture knowledge. Examples of such methods include the error mitigation in the cases of a flap runaway. Clearly, mechanisms to counteract this are necessary but a higher level of abstraction is required prior to knowing the specifics of the underlying system.

Lastly, the methods with a fidelity level close to the end-design are the ones, associated with the FCS operation during flight that are attributable to the different systems - in this thesis to the Nominal and Fallback systems. They include the automation functions and the closed-loop response. Examples of the former are the automation concepts of Chapters 3 and 4 and more. The closed-loop response is modeled using the so-called Design Reference Model (DRM) method. A comprehensive explanation on the DRM abstraction is available in [8]. The important aspects of the method are highlighted.

The DRM is a method that enables the recreation of the closed-loop system response for a control law. The approach takes the system under control’s kinetic capacity into account and thereby guarantees a physically feasible behavior. It makes use of a simplified model of the plant but at the same time allows for a highly abstracted system description. The specifics of the control concept implementation and the process of the DRM design are not in the scope of this thesis. Due to their impact on the behavioral specification model, the DRM method is examined here.

Firstly, the simplified plant does not model the effectors as physical systems but rather observes the force and moment production capacity of all effectors. Thus, the behavioral specification model needs to recreate meaningful control surface and powered-lift data

such as deflections or RPM. Fortunately, this data does not impact the DRM but is rather necessary in order to facilitate the operator awareness as to the aircraft state via the cockpit indications. Thus, for the behavioral specification model the indication data is generated fully decoupled from the closed-loop DRM operation.

Another aspect of the DRM operation is the correct selection of plant mode. The simplified plant of the DRM requires discrete information of the current flight state. This is necessary in order to schedule the force and moment production rate based on the currently utilized effectors. Such data must be provided by the behavioral specification model.

The DRM can be split into control concept and plant dynamics. One important feature of the former is that it normally includes no states, such as Simulink “integrator” or “delay” blocks. This is, on the one hand, enabled by omitting methods, such filters for signal conditioning. On the other hand, by design all necessary states related to the flight state are delegated to the plant dynamics. In the DRM method, the plant model states are utilized directly. However, while in command, the control concept relies on inputs from the automation functions in order to guarantee correctness of the execution in dependence of the current flight state. As a consequence, the control concept does not require additional initialization considerations apart from the correct data supply from the automation.

This concludes the summary of the methods utilized within the behavioral specification model. The next sections demonstrate how they are structured and designed to recreate the complete aircraft operation.

5.2 Data and Functional Management

The behavioral specification model is one of the initial phases in the development process. While the DRM concept enables the validation of the control concept, the behavioral specification model utilizes the DRM and can be used for validation of the operational concept and thereby the correctness of the automation.

A major aspect in every development process is the management of the data structures and the allocation of the functions to the individual development spaces. Having to utilize the results of the Nominal and Fallback DRMs, the file structure of the behavioral specification model needs to ensure compatibility between the different processes. This section elaborates on how this is achieved.

This section is composed of two sections. In Section 5.2.1, the file structure of the exemplary behavioral specification model is illustrated. This is highly related to the repository structure, which is necessary for version management. Hence, this is explained as well. The section demonstrates where the different automation functions are stored and provides explanations for the motivation behind the placement.

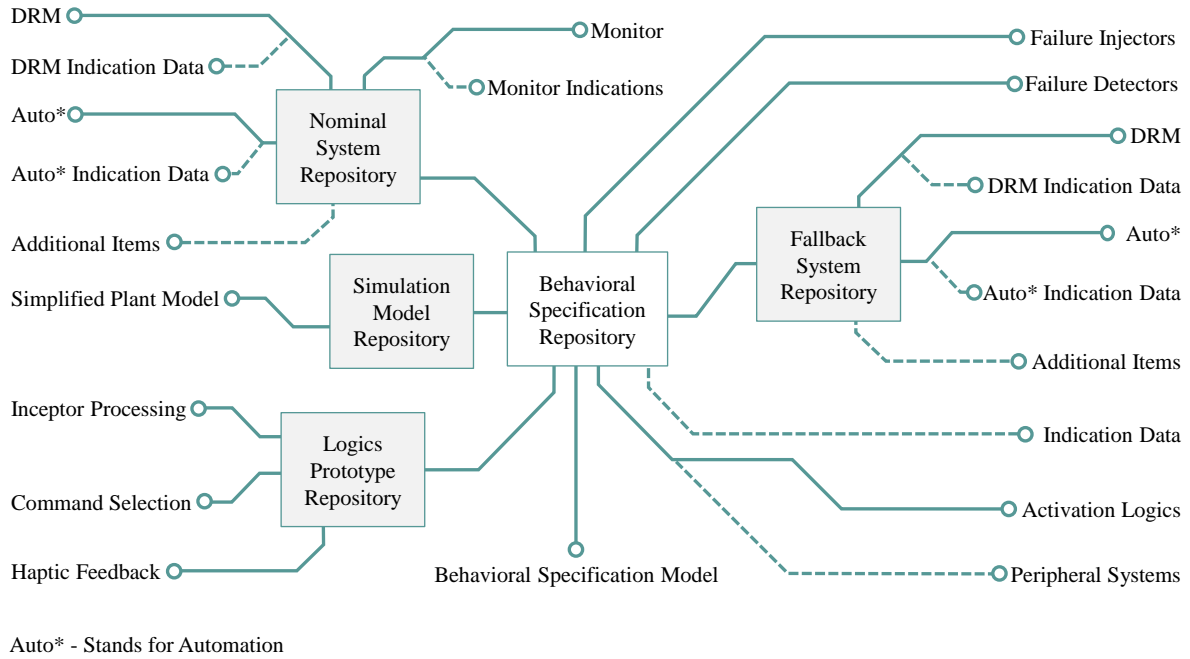


Figure 5.2: *Repository and File Structure of the Exemplary Behavioral Specification*

The section is concluded with Section 5.2.2 which demonstrates the development process of the automation functions. It focuses on the topic of change management, especially for the case of functions that are highly dependent on the interactions between the individual software components.

5.2.1 Repository and File Structure

Figure 5.2 visualizes the models and repositories involved in an exemplary behavioral specification model. The top-level repository is depicted in white, whereas the remainder of the repositories are submodules of that repository. The main components within each module are listed next to them.

The provided example is the behavioral specification file structure for the aircraft of Section 1.1.4. It is composed of two control concepts and hence DRM designs - the Nominal and Fallback. Parameters, necessary for all functions are kept in Data Dictionaries [132].

The Nominal system includes a functional monitor, while the Fallback system does not require such a feature. This is evident in Figure 5.2 and in order to verify the monitoring concept as previously mentioned in Section 5.1.1, this function is included in the corresponding repository. The functional monitor is not in the scope of this thesis, but allows for runtime assurance of the Nominal control concept in an independent and dissimilar manner.

In Figure 5.2, the abbreviation “Auto” is used to summarize the automation function design for the individual DRM modules. As visible from the figure, each automation function is part of the repositories of the systems and therefore developed within the

corresponding process. This structuring allow for the verification and validation of the closed-loop design of DRM together with its automation in a thorough manner prior to the system integration in the behavioral model. Among many others, the automation function modules of the Nominal and Fallback control concepts include the functionality that is attributable to the transition and retransition previously presented in Chapters 3 and 4.

A consequence of this file allocation is that the automation functions within the Nominal and Fallback system repositories are solely responsible for the operation of the control concept while it is engaged. Therefore, all additional functionality, such as the engagement of the control concept, the interaction concept harmonization, all ground procedures and others are allocated to the behavioral specification file structure. They are what is referred to as shared in Section 5.1.3. This carries the advantage that the maintenance, associated with changes to these concepts is carried out in one location and can be kept consistent for both Nominal and Fallback systems.

The behavioral specification model, depicted in Figure 5.2 integrates all models and enables the complete simulation of the aircraft operation. It must be noted that both Nominal and Fallback system repositories utilize the simplified plant that stems from a separate repository. For the sake of readability this is omitted in Figure 5.2, where only one repository instance is referenced. Prior to integration it must therefore be ensured that all such shared repositories are on the same stage. The simplified plant repository is one example, but such considerations also apply to common conventions, such as mappings of control inceptors and many others.

In Figure 5.2 certain structures are depicted with dashed lines. Those are, on the one hand, multiple items that are omitted for the sake of readability. For example all additional functions, such as altitude hold or altitude protection automation functions for the Nominal system are contained under the “Additional Items” category of Figure 5.2. On the other hand, dashed lines are used to depict application-specific items, such as the indication data generation that is necessary for cockpit indications. The last category are constructs of highly reduced fidelity and specific to the system architecture that could be included later in the behavioral specification model design once their role and operation is better known.

Lastly, the “Logics Prototype Repository” visible in Figure 5.2 contains functions that are shared within different stages of the development process. The mentioned function are the basis for design of higher-fidelity and code-compliant modules. Examples include the haptic feedback behavior or the processing and conditioning of the pilot control inceptors. In addition, the behavioral specification models of these functions can be utilized if necessary in later stages of control concept development. Therefore, a separate repository is deemed meaningful in order to have access to the functions without having to include all other items, associated with the behavioral specification repository.

5.2.2 Change Management

During the functional testing and validation of the integrated system behavior, different improvement potential was pinpointed. This included missing functionalities or inadequate interaction between functional elements. The initially high likelihood of wrong function specifications is mainly driven by the novelty of the underlying airframe, its control and automation concept and ConOps. This is especially applicable to the interaction concept, namely to the design of the indication items. It was found that items that facilitate adequate, ergonomic and intuitive operator support and enable situational awareness require multiple validation cycles.

The development of the behavioral specification and of its corresponding elements is therefore an iterative process, in which an update to the specification and subsequent renewed validation need to be supported by the functional distribution within the data structure. Upon discovery of necessary modifications, certain deficiencies can directly be delegated and alleviated in the corresponding submodules of the behavioral specification. These can for example be changes necessary in the automation of solely the Nominal system due to changes of the dynamics of the LTUs in the simplified plant. Another example includes the inclusion of altitude protection functions that were not envision in the beginning of the development.

However, certain issues arise whenever improvement potential is observed in interactions between components of different development paths. For example, during testing of the integrated system, it was noticed that with the specified pilot reaction times in [122, 133], in certain situations the aircraft safe envelope was exceeded following a takeover from the Nominal system by the Fallback system. Namely, after the reaction time, the operator was incapable of stabilizing the system in time.

Prior to the integration, this phenomenon could not be detected or even reproduced in the standalone Fallback system development due to the lack of aircraft behavior with the Nominal system in that environment. As a consequence, the issue can only be solved efficiently in the behavioral specification model where both systems are present.

The developed framework allows for adequate flexibility in the design to address and alleviate such issues in an efficient manner. The above-mentioned problem was discovered and a solution for it was developed by Daniel Gierszewski of TUM Institute of Flight System Dynamics and is not in the scope of this thesis. Here, the utilized workflow is of importance. It is inspired from the agile development processes [55] and is depicted with Figure 5.3. This cycle only tackles the method of alleviating such potential issues and does not deal with the modification of the high-level requirements. They follow formal processes recognized in the aviation industry are not in the scope of this thesis.

The change cycle begins with the identification of the improvement point. It refers to the observation of the issue and gaining understanding of the underlying reason of the effect. Typically, this is performed during validation.

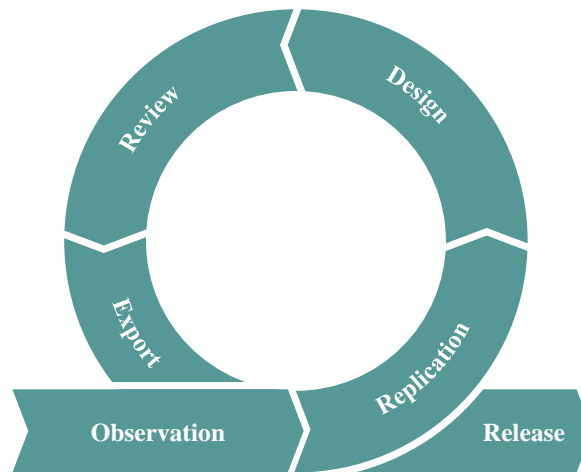


Figure 5.3: *Development Cycle for Functions that Tackle System Interactions*

Logically, this activity is followed by the effort to reproduce the issue in a deterministic manner. In particular, a test case needs to be formally defined, in which the unwanted effect is replicated. This is used not only as a basis for the design of the solution. In addition, the test case can be reused from then on to ensure that the problematic effect does not manifest at later points during development.

Next, a fix for the observed problematic interaction is designed. This is performed within the behavioral specification framework. By doing this, the needed additional functionality can be specified in detail and a placement within the necessary system can be decided.

The design imposes a specific solution that needs to be allocated to the particular system. In the example from above, the reaction of the Fallback system in the event of a takeover needs to be modified. The inclusion of this proposal in the design of the Fallback system requires a review from the involved designers in order to guarantee that other functionality is not negatively impacted or compromised.

Once the proposed solution is accepted, in the next step the design is migrated from behavioral specification model into the necessary system file structure. This is performed via a separate branch in the corresponding repository. Because the solution is already implemented, the duration in which the branch is open is kept low and it does not hinder the remaining designers. Subsequently, a new system release is prepared, which is the basis for further validation effort.

This method of identification of missing functionality and subsequent supplementation within the behavioral specification model was first applied and from then on utilized heavily in the development of cockpit indication behavior.

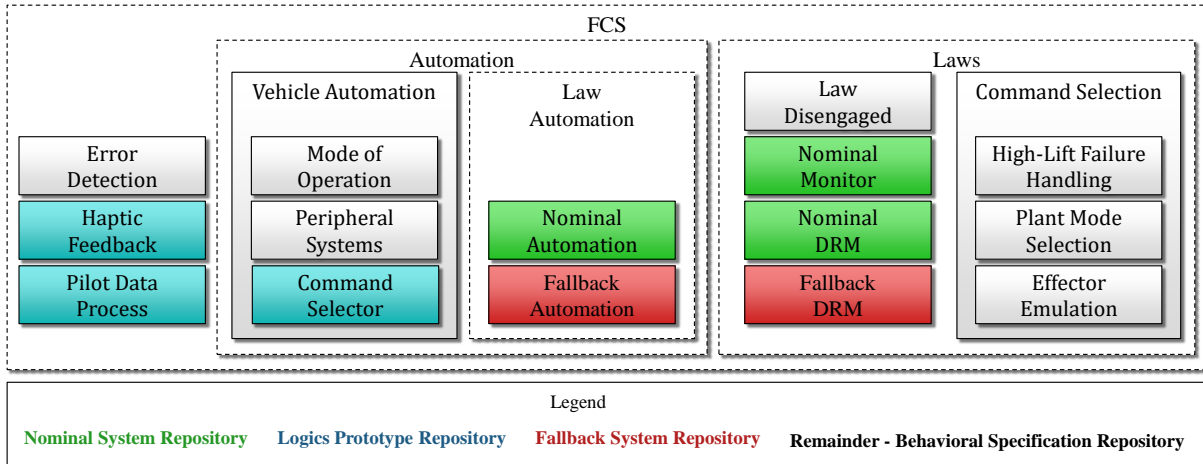


Figure 5.4: Overview of the Model Architecture

5.3 Model Architecture

This section presents the FCS behavioral specification model architecture. Figure 5.4 depicts the layout of the deployed models. For the sake of readability, in the figure the data flow of the involved modules and their interactions are omitted. In addition, not depicted in the figure is that the FCS interacts with the simplified plant model in a closed-loop manner directly without the use of sensor models. Thus, the outputs of the plant are inputs to the FCS and vice-versa.

The color coding in Figure 5.4 is in accordance with the origin of the functions in terms of repository. The repositories are already mentioned in Figure 5.2 and Section 5.2. All blocks without color are contained in the repository of the behavioral specification model. The section breaks down the architecture into several topics that are listed in the following sections. In addition, Appendix F provides images of the behavioral specification model for the aircraft of Section 1.1.4.

5.3.1 Utilization of Simulink Libraries

First, the main referencing constructs are explained. The standard workflow for embedded software design at TUM-FSD utilizes model references. In the behavioral specification model this construct is avoided. The reason is as follows.

Model references are useful when designing software, intended for flight due to advantages in the code generation. However, this has the potential to introduce artificial algebraic loops [134–136] due to the incapability to split the functions within the model references according to more suitable execution orders. Therefore, additional Simulink delay blocks are necessary to mitigate this issue. Instead of utilizing this method, here the functional elements are maintained in Simulink library blocks where inlining [137] of the elements is explicitly disabled.

By this means, the problem with artificially created algebraic loops is alleviated. At the same time, version and change management of the functions is maintained in a similar fashion as with model references. An additional benefit observed is that the compilation time is reduced compared to the use of model references.

5.3.2 Error Detection

As depicted in Figure 5.4 for rapid prototyping and evaluation, the error detection capability is consolidated within one element of the FCS and is not physically placed at each instance of the control system (Nominal or Fallback). This, on the one hand, aids in the management and troubleshooting of the functions due to their integration into one functional element. On the other hand, this reduces the dependency of the framework on the system architecture.

The error detection is highly simplified as previously discussed in Section 5.1.2. As mentioned there, for each signal source, error injection signals are provided as inputs to the system in order to simulate the system's response for component malfunctions.

An example of the error detection functions is provided in Figure F.1 found in the Appendix. For each signal source, a flag is raised to *true* a specific number of cycles after the corresponding error injection signal has also been set to *true*. This is realized using a Confirmation Counter with the error injection signal as an input. This is in addition performed for each control instance within the FCS. The behavioral specification model is therefore capable of first reproducing the reaction during an undetected erroneous of a particular signal source. After the specified number of cycles is exceeded, a detection of the error is simulated and the subsequent system action can be produced.

5.3.3 Integration Models

The elements in Figure 5.4 marked with dashed lines represent integration models that consolidate elements of similar functionality or ones that have high mutual dependency. For example, the Nominal and Fallback automation modules can be found under the "Law Automation" integration model. Similarly, vehicle and law automation form the "Automation" integration model. They are depicted in Figures F.2 and F.3 respectively.

The difference between the integration modules and all other modules within the behavioral specification is that the integration models do not include any functionality and are required for better structuring, readability and testability. Apart from the main functions, the integration models include the following functional elements that are not depicted in Figure 5.4 for the sake of readability.

The first element included within the confines of the integration model are the error injection modules. Those are tailored to each included model and are found immediately before and after the main functional elements. In the error injection models, the supplied data is manipulated depending on the error type. Thus using the error injectors before the

main function would depict the system response for erroneous inputs of the functionality. Similarly, the error injection found immediately after the function can depict a malfunction of the functional component itself due to a hardware failure and - in the case of the Nominal system - a design failure as well.

The next element within the integration models are interface modules. As visible in Figure 5.4, a majority of the models originate from different repositories that share very little dependency. Apart from the simplified plant outputs, the models of those create their own input and output data structures. The information, originating from the different systems required by the mentioned models, is therefore consolidated and prepared in the interface modules.

The last functional element within the integration models are ones that process information that has high dependency on modules of similar functionality. This can for example be the takeover initialization or inputs that facilitate the procedure harmonization found in Sections 4.2.3 and 4.4.2 of Chapter 4 respectively. An example of an integration model is provided in Figure F.4.

5.3.4 Vehicle Automation

A central element and the core of the logic within the behavioral specification model is the “Vehicle Automation” visible in Figure 5.4. This model fulfills the overall coordination of every key task within the aircraft operation. For the current example, it is depicted in Figure 5.5. As visible in the image, it is composed of multiple levels of parallel sequential logic modules, the majority of which are Finite-State Automata. It is modeled in a centralized manner to alleviate the implementation complexity as previously mentioned in Section 5.1.2.

Among others, the management and emulation of all peripheral system is allocated within the vehicle automation functionality. Depending on the inputs from the operator and the injected errors, it determines the operational state of the aircraft. This also includes the operation on the ground and the activation of the laws. The currently executed mode is thereby a task of the vehicle automation’s function, referred to as “Mode of Operation”, explained in the next section.

5.3.4.1 Mode of Operation

As visible in Figure 5.5, the mode of operation contains multiple states and is composed of two levels. It is implemented according to the workflow found in [77]. Based on the current state of the peripheral systems and the detected errors, the availability of the different control modes and their subfunctions is evaluated. As visible in Figure 5.5, the first level of the mode of operation is responsible to specify the current system in command. The system in command is selected manually by the operator via dedicated input items which are not in the scope of this thesis. Automatic selection due to safety requirements is also

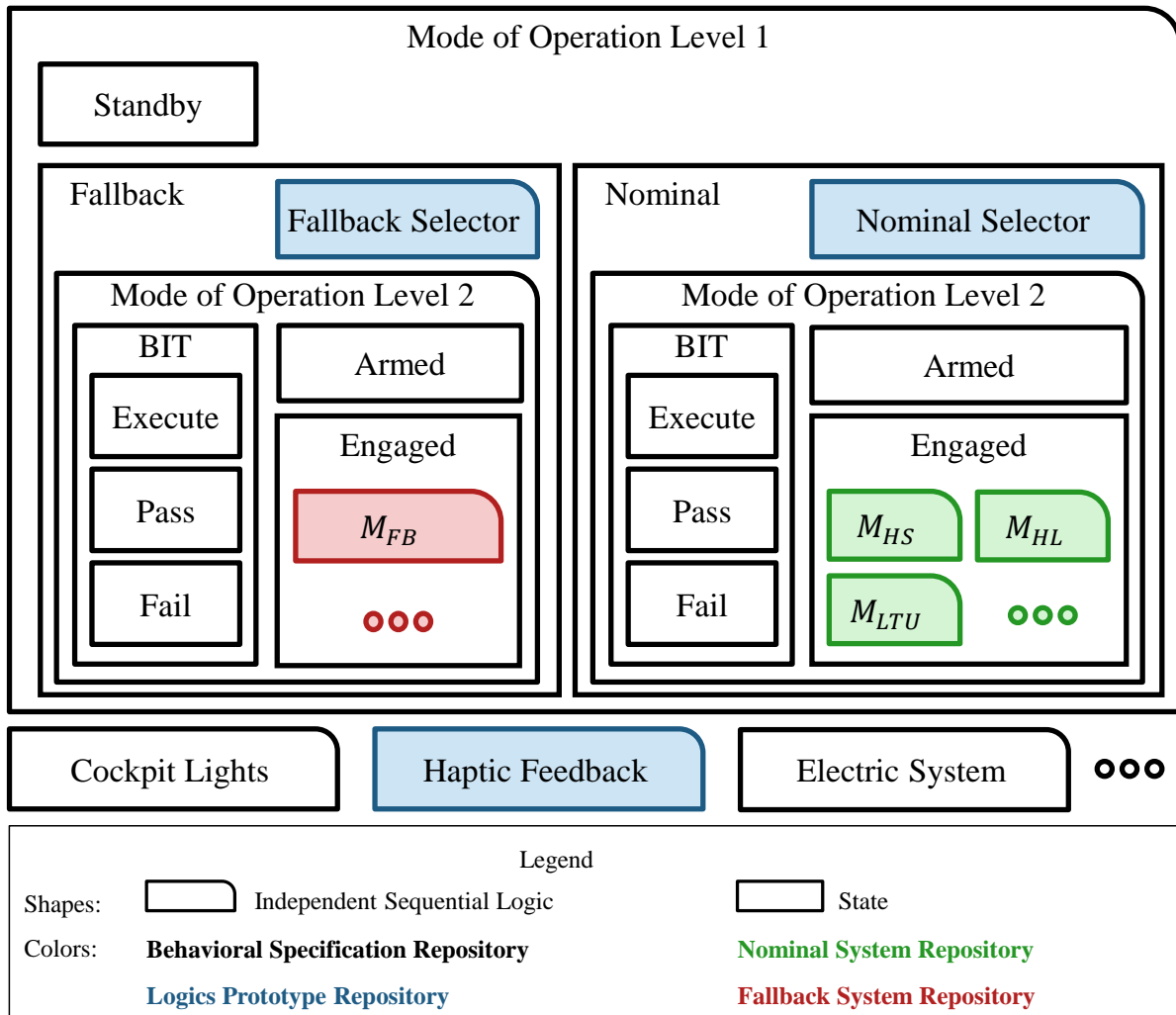


Figure 5.5: Behavioral Specification Vehicle Automation Architecture

modeled based on the signal availability. This is calculated within the vehicle automation using the error detection data. For the specification of the currently engaged system, the vehicle automation element relies on the command selector function, explained in the next section.

Whenever a system is in command, then the second layer of the mode of operation determines the FCS state of automation. The current example distinguishes between three main modes, namely “Armed”, “Built-In Test (BIT)” and “Engaged”. The flow from Armed to Engaged via BIT is the activation procedure of the control concept and the flow from Engaged to Armed follows the deactivation procedure. In both Armed and BIT state, the operator control inceptor input does not induce any change in effector commands, whereas in the Engaged state, the deflections of the pilot inceptors are fully utilized by the corresponding control concept.

In the Armed state, the selected control concept is available and both distributed and traction propulsion systems are fully disengaged. In the background, the operator is executing the prescribed checklists and activating the various systems. This is processed by the peripheral system emulation, found in Figure 5.4. Thereby, different electrical components are turned on and configured, the cockpit indications are supplied with the required data and more. This information is also fed to the mode of operator.

When the conditions are met and the appropriate operator input is registered, the system transitions into the BIT mode. In this state, the distributed and traction propulsion is engaged and the correctness of the activation is checked. This greatly mitigates the potential of in-flight failures [124] and additionally facilitates a smoother activation of the selected control mode due to the effector preactivation.

Provided the BIT checks pass, the operator is capable of engaging the selected control concept. At this point, the law-specific automation is enabled. The modules stem from the corresponding repositories and include the transition and retransition elements as described in Chapters 3 and 4 for Nominal and Fallback system operation respectively.

It must be mentioned that with the exception of the control mode selection, the activation and deactivation procedures are identical. In Figure 5.5 it is visible that the second level of the mode of operation is identical with the exception of the law-specific automation within the engaged state. However, as evident in Figure 5.4, those are instead found in the “Law Automation” module. Therefore, the vehicle automation actually utilizes one instance of the second level of the mode of operation. Figure 5.5 depicts it twice for the sake of readability.

5.3.4.2 Command Selection and Control Concept Replication

The vehicle automation’s mode of operation module specifies the system in command. In this example, this could be the Nominal or the Fallback system. In addition, when a power off of the avionics is simulated, none can be selected. Function replication is a term which implies that multiple instances of a particular control concept are available within

the FCS. For instance, in the behavioral specification model here, the Fallback system would have to be available in at least two instances in order to guarantee the availability of the takeover function.

If the vehicle automation selects the system in command, the “Command Selector” visible in Figure 5.4 is the centralized algorithm that is utilized for function replication to specify which instance of the particular system is selected. The benefits of opting for a centralized algorithm within the behavioral specification model was already mentioned in Section 5.1.2.

This module implicitly requires knowledge of the number of replications of each control concept and therefore carries a large degree of design decisions in that regard. In addition, the method of selection is application specific. However, although the algorithm must account for the number of control concept instances, it requires no knowledge with regards to the physical allocation of the instances to flight control computers. It therefore guarantees a high-degree of independence with respect to the system architecture.

It must be noted that in very early stages of the aircraft development where the number of instances is not known at all, this module could be omitted completely. This is because the system in command evaluation is not performed here, but in the mode of operation module instead.

Another detail with regards to this module is that the number of replications for each system are assumed. However, no assumption as to the physical allocation is made. This makes the algorithm largely agnostic to the system architecture.

5.3.4.3 Law Automation and Law Utilization

As discussed previously in Section 5.2.1, the two control concepts are available from two separate repository structures. This includes their corresponding automation modules. Among others, they are responsible for automating the transition and retransition as described in Chapters 3 and 4 for Nominal and Fallback system operation respectively.

As visible in Figure 5.4 and previously discussed, control algorithms and automation are physically separated for testability purposes. In addition to this, for each function - automation, law and monitor (if applicable) - only one library block is utilized, regardless of the number of function replication which facilitates a degree of independence of the system architecture as discussed previously in Section 5.3.4.2. In the event of a switch from one control instance to another (for example from Nominal system one to Nominal system two), proper initialization is guaranteed by DRM design as mentioned in Section 5.1.3.

The decision-making modules within the law automation are allocated to enabled subsystems. Those subsystems are activated whenever the control mode is engaged and selected. This enables the execution of the law-specific automation only when necessary and guarantees coherence with the depiction in Figure 5.5. Figure F.5 demonstrates this for the Nominal system and this is done for the Fallback system in an identical manner.

The Nominal and Fallback DRM algorithms describe the closed-loop control response during flight. For this reason, in addition to the two DRMs, the behavior of the aircraft on the ground is included with a separate model within the Law integration model of Figure 5.4 called “Law Disengaged” which is allocated in the behavioral specification model file structure. This includes the reaction of the system when completely disarmed and during the engagement process of each control concept. It is visible in Figure F.2.

Lastly, the outputs of the correct system are rooted to the simplified plant model via the “Command Selection” function. In addition, this module handles the system response when failures are detected. Because the simplified plant does not include effectors but only force and moment production capabilities, this element additionally provides a simplistic effector emulation which is necessary for the cockpit indications.

5.4 Haptic Feedback Automation and Pilot Input Processing

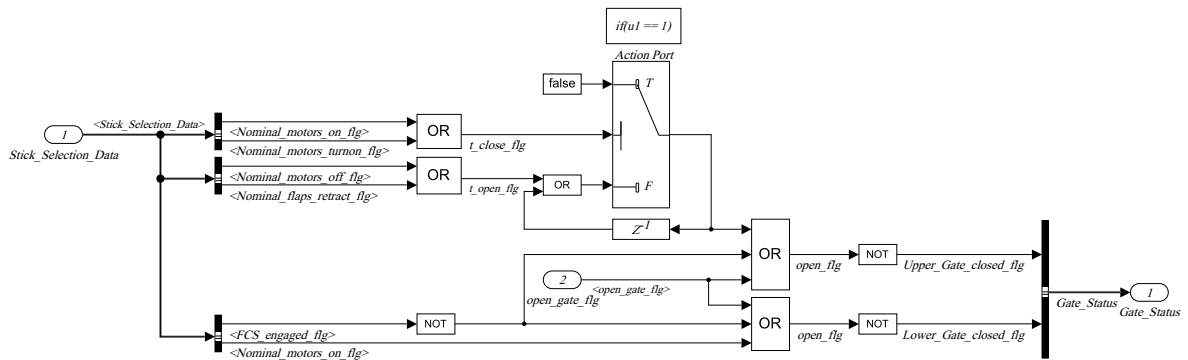
This section presents the behavioral specification modeling of the haptic feedback, supplied to the operator via the throttle control inceptor. In addition, the input processing of the pilot input that is shared among Nominal and Fallback system is discussed. As explained in Section 5.2.1, this information is processed within the Logics Prototype Repository.

For recollection, the throttle control inceptor includes two barrier elements that can limit the entry of the inceptor into the different regions. This was explained in Section 2.4.3.1. The operation of the barriers within the scope of the transition and retransition procedure harmonization between Nominal and Fallback systems was prescribed in Section 4.4.1.

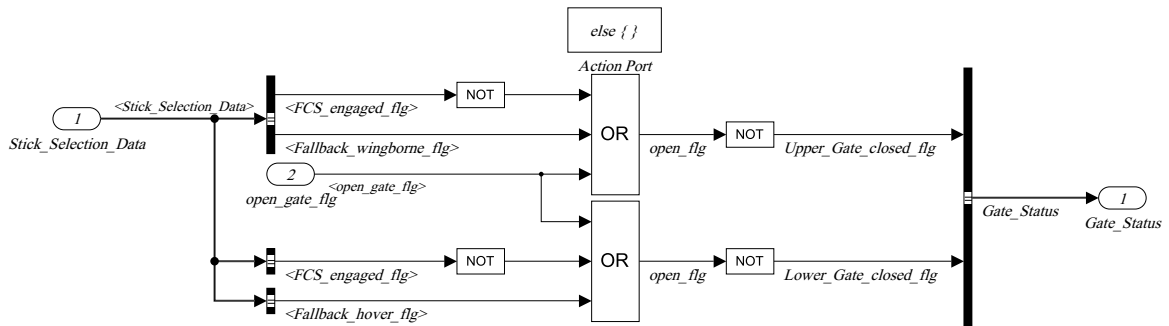
More precisely, the decision with relation to the barrier status was summarized in Table 4.7. Figure 5.6 illustrates how this behavior is realized within the behavioral specification model. The two subfigures are part of an if-clause, where a check of the current system in command is performed. The upper evaluation is executed whenever the Nominal system is selected, whereas the lower is chosen whenever this is not the case. The command evaluation is contained in the variable $u1$ contained in the if-clause and visible in the upper block of Figure 5.6.

From both graphs in Figure 5.6 it is visible that the constraint to lift both barriers when $OPEN_{GATE}$ is *true* is maintained. Though according to Section 2.4.3.1 this is realized mechanically independent of the logic, for the sake of completeness in the behavioral specification, this is simulated as well. Otherwise, whenever $OPEN_{GATE}$ is *false*, the prescribed behavior in Table 4.7 is maintained by the remainder of the functions.

The validation activities of the operational concept discovered that the operator may deviate from the transition and retransition procedures and overstep the allowed region, dictating the barrier function operation via $OPEN_{GATE}$. The subsequent release of $OPEN_{GATE}$ to *false* prohibited the operator from returning to the intended throttle



(a) Gate Behavior when the Nominal System is in Command



(b) Barrier Behavior when the Nominal System is not in Command

Figure 5.6: Barrier Behavior

region. For this, additional robustness features are implemented within the behavioral specification. For the sake of readability, these considerations are omitted in this thesis and are instead in the scope of further publications.

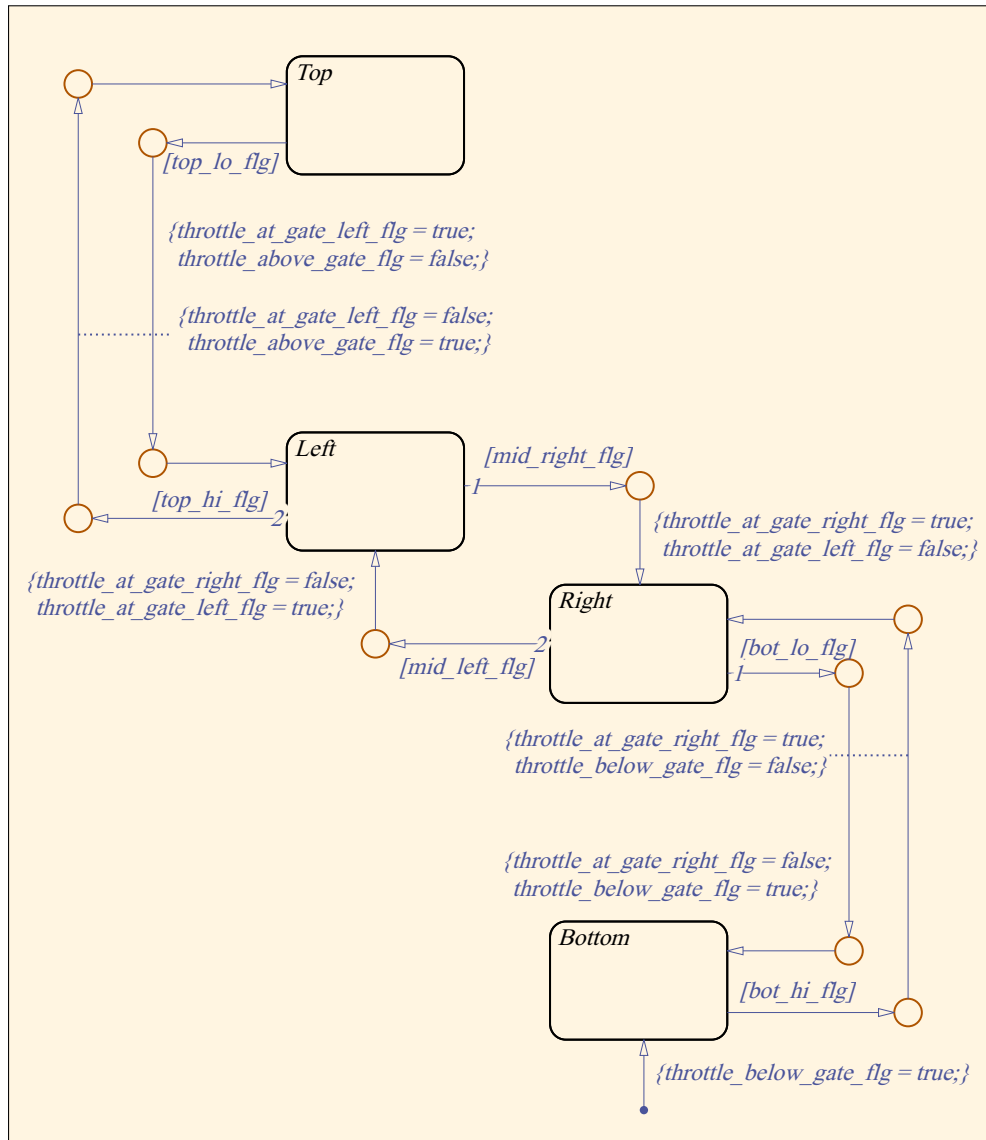
Another central element of the functions, shared among Nominal and Fallback systems is the processing of the operator input from the throttle command inceptor. This involves the evaluation of the throttle position with respect to the inceptor range divisions, mentioned in Section 2.4.3.1, and the operator desired mode of operation. The methods applied to facilitate this are illustrated in Figure 5.7. The upper graph depicts the State Machine used to analyze the current throttle position. The parameters for the evaluation are chosen such that robustness against sensor noise is ensured.

The resulting State Machine outputs are unambiguously assignable to the regions, defined in Section 2.4.3.1. Namely, the throttle inceptor is in the powered-lift region whenever *throttle_below_gate_flg* or *throttle_at_gate_right_flg* are *true*. An additional check is performed using a “Compare to Constant” block to differentiate between the regions \mathbb{H} and \mathbb{T} but is omitted here for the sake of readability. Similarly, the inceptor is positioned in the wingborne region \mathbb{W} if either *throttle_above_gate_flg* or *throttle_at_gate_left_flg* is *true*. Lastly, the divisions \mathbb{R} and \mathbb{L} apply for *true* conditions of *throttle_at_gate_right_flg* and *throttle_at_gate_left_flg* respectively.

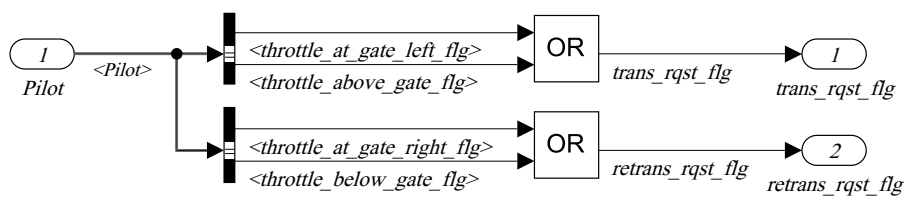
Another observation is that the state transition from one state to the other and vice-versa is done using different variables. The difference in the values is small but is necessary for robustness considerations. The provision ensures that the state does not frequently change due to sensor noise whenever the control inceptor is in the vicinity of the region borders.

In the upper illustration of Figure 5.7 it can be noticed that the inceptor is initially always assumed to be in below the gate. In the cases where the simulation is initialized and the physical throttle inceptor not in this region, it can take up to three simulation cycles to arrive at the correct division. This would be the case if the throttle inceptor is above the gate in the wingborne region. No robustness against such events is built in as the State Machine corrections are magnitudes faster than the activation of the FCS according to the procedures. As a consequence, the correct position evaluation occurs before any event, associated with the activation procedures.

The lower portion of Figure 5.7 visualizes the evaluation of the pilot intentions with regards to the transition and retransition. These values are used by the Decision-Making processes of both Nominal and Fallback systems and are presented in Section 3.4.1.1 and 4.2.1 respectively. More precisely, the evaluation depicted in the lower part of Figure 5.7 implements Equations 3.3 and 3.4 for the Nominal system whereas the Fallback system utilizes the upper evaluation as per Equation 4.5.



(a) Pilot Input Processing



(b) Pilot Flight Phase Requests

Figure 5.7: Pilot Input Processing

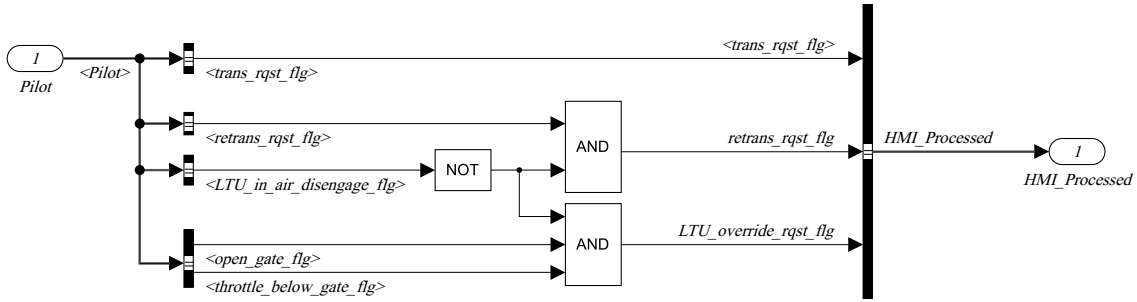


Figure 5.8: Processing of the Human-Machine-Interface

5.5 Nominal Automation

This section presents the behavioral specification modeling of the Nominal system automation. As per Section 5.2, it is allocated to the Nominal System Repository and as per Section 5.3 it operates within the Law Automation integration module.

The Nominal system operational concept includes multiple high-degree of automation capabilities, such as the transition and retransition capability, the management of altitude protections, altitude hold and more. The emphasis in this section is on the core modules, utilized to produce a high-fidelity behavior of the automation concepts in Chapter 3.

The concepts found in Chapter 3 include the operation of the LTUs, found in Section 5.5.1. As discussed in Section 2.4.1.1, the Nominal system control algorithms require knowledge of the flight phase in the powered-lift flight mode. The automation module, responsible for this is discussed in Section 5.5.2.

Because the exemplary vehicle in this thesis in Section 1.1.4 includes a high-lift system, the automation of the flap operation with regards to the transition and retransition is considered in Section 5.5.1 and the high-lift system management itself is presented in 5.5.3. For the sake of consistency, the contents of the above-mentioned sections are organized in a similar manner as the structure of Chapter 3. First the Decision-Atomics of the individual functions are presented and are followed by the Decision-Making modules. The Atomics make up the basic relationships, utilized by the Decision-Making.

5.5.1 Powered-Lift System Operation

As found in Section 3.4.1.1, the Decision-Atomics of the automation that dictates the LTU operation using the State Machine M_{LTU} begins with the processing of the operator input via the HMI. The pilot actions communicate the crew intentions to the automation and

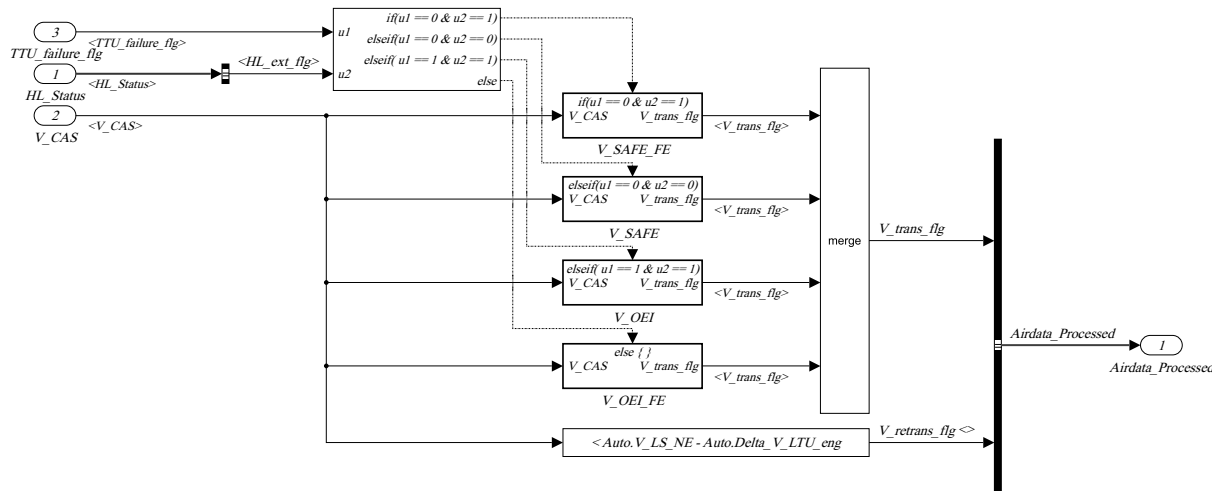


Figure 5.9: Processing of the Airdata

the implementation of the processing is depicted in Figure 5.8. As visible in the figure, the input variable $trans_{rqst}$ that is calculated as per Equation 3.3 is taken as-is from the processing previously depicted in Figure 5.7 found in Section 5.4.

For $retrans_{rqst}$, the information is taken in a similar manner. However, an additional condition is introduced, summarized in the signal $LTU_in_air_disengage_flg$. This variable is processed from an additional operator input, necessary in abnormal events where the powered-lift system should be prohibited following the transition into wingborne flight. It is used to communicate to the automation that the engagement of the LTUs should not be executed despite the movement of the throttle inceptor into the transition/retransition command region. Thus, the usage of this input item enables the wingborne landing of the aircraft. The input item is utilized within a separate aircraft procedure and both input item and procedure are not in the scope of this thesis.

The last pilot input processing has to deal with the generation of the input variable $LTUengage_{rqst}$. Apart from the dependency on $LTU_in_air_disengage_flg$ that was explained in the previous paragraph, the variable generation is in accordance with Equation 4.34. For recollection, the computation is conducted so as to guarantee consistency in the retransition procedures among Nominal and Fallback systems and was derived previously in Section 4.4.2. The evaluation of the position with relation to the throttle inceptor gate that is necessary for the variable generation is realized using the logical operation, introduced previously in Section 5.4.

The State Machine M_{LTU} that is responsible for the management of the powered-lift system requires knowledge of the current flight state. More precisely, the disengagement of the LTUs has to occur in a permissible airspeed range where stall is mitigated and obstacle clearance is facilitated. Likewise, the engagement of the LTUs must be performed such that no structural damage can occur. These considerations are analyzed thoroughly and can be found in Chapter 3. The airdata processing of the Decision-Atoms of M_{LTU} is depicted in Figure 5.9 and is in accordance with Sections 3.4.1.1 and 3.4.3.2.

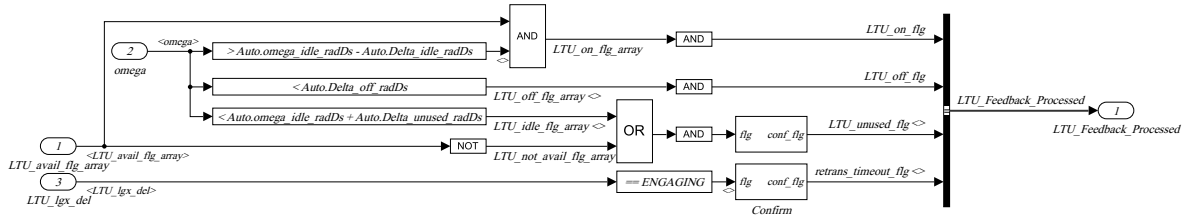


Figure 5.10: Processing of the LTU Feedback

In the current example of the behavioral specification model, the aircraft is equipped with a high-lift system. Therefore, the evaluation of the variable V_{trans} is conducted as per Equation 3.53 and is the first signal, visible in 5.9. In the cases, where the aircraft does not have flaps, this can be performed using Equation 3.9 instead. As visible in Figure 5.9, the choice of computation of V_{trans} is implemented using an if-clause, whereby the different inequalities in Equation 3.53 are within “If Action Subsystems”. For the sake of readability, they are not depicted in separate images. The information, necessary for correct operation of the if-clause stems from the error detection and the flap feedback information and the origins of the data is not depicted further for the sake of readability.

The second signal, generated as depicted in Figure 5.9 is the variable $V_{retrans}$. Using it, the system guarantees that the activation of the powered-lift system causes no structural damage to the airframe. The model implements the check in accordance with Equation 3.10, found in Section 3.4.1.1 of Chapter 3.

The Decision-Atomics module of the powered-lift system automation module further needs to evaluate the state of the LTUs. This is necessary in order to determine the lack of powered-lift system usage prior to the initiation of the LTU deactivation. In addition, the correctness of engagement and disengagement needs to be evaluated by the software in order to confirm the successful start or end of the transition and retransition processes respectively. This is discussed at length in Chapter 3 and the implementation of this functionality within the behavioral specification model is depicted in Figure 5.10.

As visible from Figure 5.10, the feedback and integrity of each LTU is processed. Thereby, the automation evaluates the current usage of every LTU. More precisely, the implementation visible in Figure 5.10 generates the input variables LTU_{ON} , LTU_{OFF} and LTU_{UNUSED} in accordance with Equations 3.13, 3.14 and 3.15 respectively. The latter equation implements a Confirmation Counter, previously derived in Section 2.2.3.4. The inputs of the Confirmation Counter is, on the one hand, the implementation of Equation 3.16 as required in Section 3.4.1.1. On the other hand, the threshold is a parameter, found in the “Confirm” masked subsystem, visible in Figure 5.10.

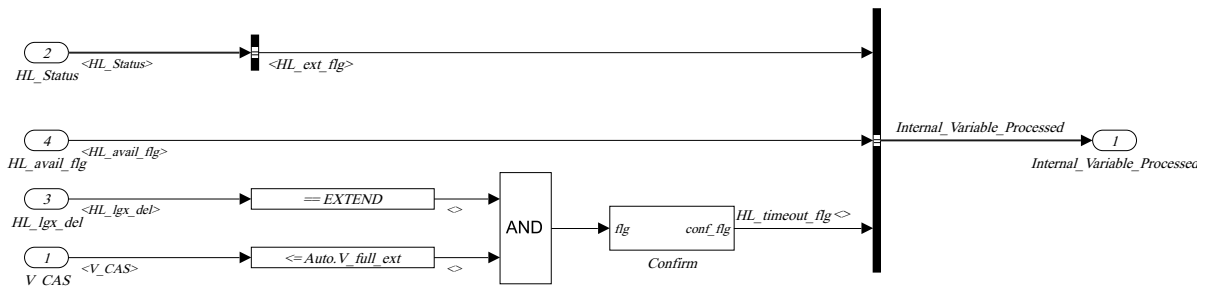


Figure 5.11: *Powered-Lift Automation Provisions for High-Lift System Operation*

The last signal, generated as per Figure 5.10 implements Equation 3.17 of Chapter 3. For recollection, this variable becomes *true* if the engagement process takes unexpectedly long. This evaluation is facilitated by the deterministic activation time of the LTUs, achieved via a RPM command ramp-up through the Nominal system’s control allocation. The duration of the activation is hence known.

For recollection, the reason why the timeout may occur is a malfunction of the powered-lift system that leads to the impossibility of LTU_{ON} as per Equations 3.13 to become *true*. Via *retrans_timeout*, the automation sends the necessary for the provision of a warning to the operator that formally marks the start of the retransition mitigation strategies. This is discussed at length in Chapter 3.

The last portion of the Decision-Atomics functionality of the behavioral specification model tackles the effects the high-lift system has on the operation of the powered-lift automation. As previously mentioned in Chapter 3, in the cases where the aircraft is not equipped with flaps, this can be omitted. Thereby, the supplementation of the logic, found in 3.4.3.2 can be omitted. For the current example specification, high-lift system operation is considered and is depicted in Figure 5.11.

For recollection, in order to maintain the execution order of the two system (powered-lift and flaps) during retransition, the powered-lift system activation is preceded by the extension of the high-lift system. Therefore, the first signal found in Figure 5.11 is utilized for the normal case, in which the automation of the powered-lift system waits for the high-lift system extension. The information stems from the flap feedback and the origins of the data are not depicted further for the sake of readability. The computation of the signal is done in accordance with Equation 3.51.

The next two signals of Figure 5.11 are used in abnormal events, in which the flaps malfunction. This enables the activation of the powered-lift system despite the failure to extend the flaps. The first item originates from the failure detection functions of the behavioral specification model and covers a detected erroneous flap operation. Because the time and conditions of extension are known, the last signal in Figure 5.11 implement a functional evaluation of an abnormal scenario in the events of a undetected erroneous flap operation. They implement the timeout of Equation 3.52.

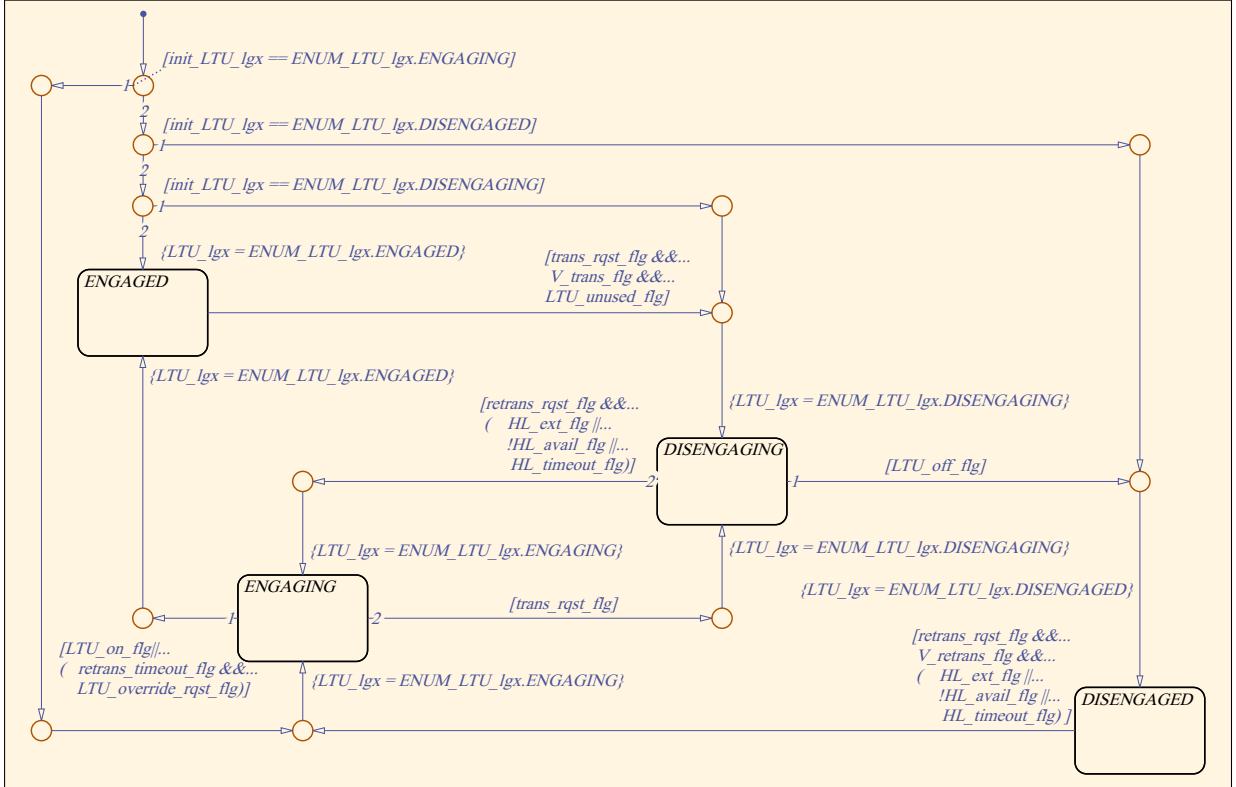
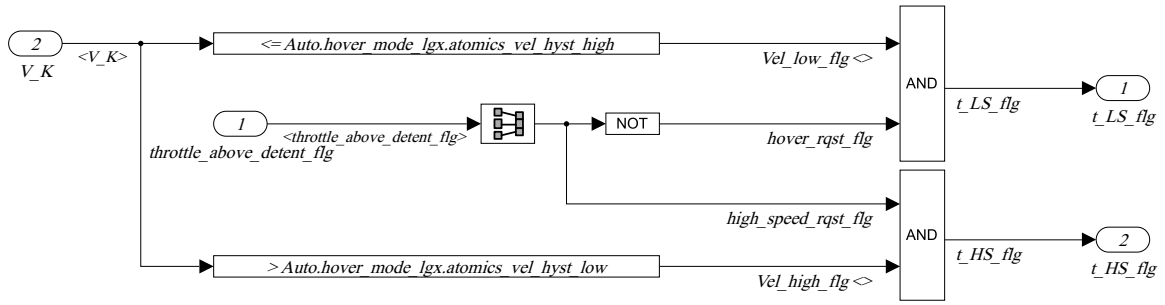


Figure 5.12: M_{LTU} Implementation

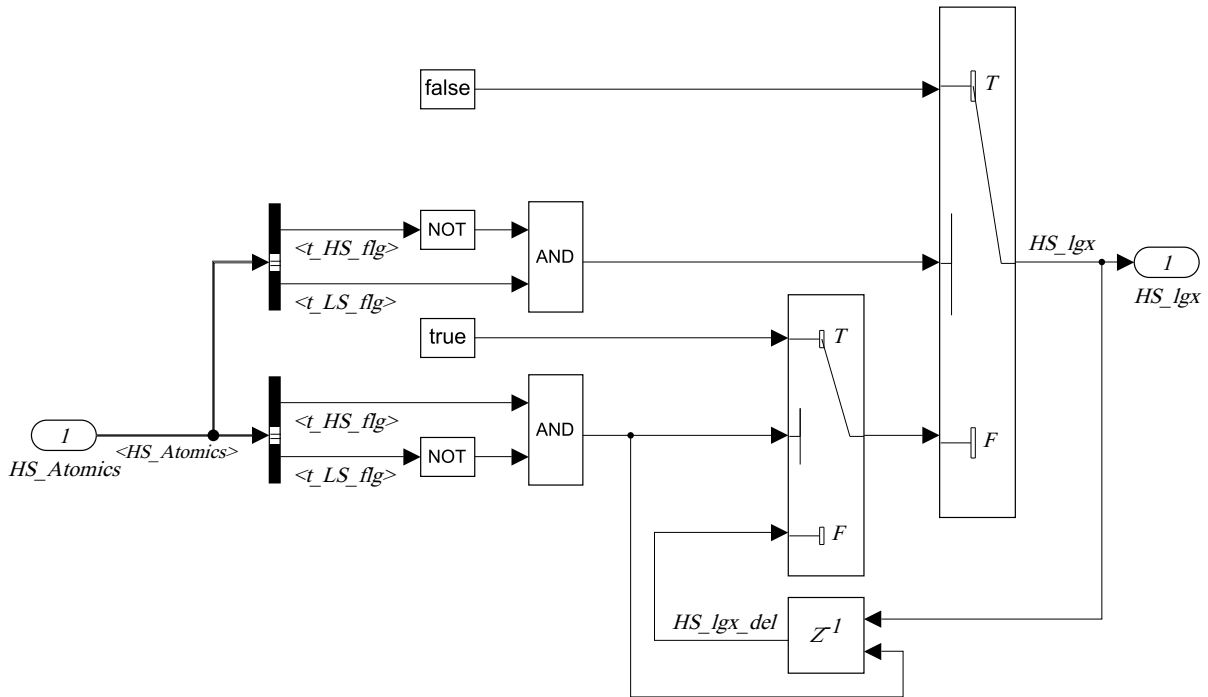
This concludes the Decision-Atomics of the Nominal system powered-lift automation module. The information from the Decision-Atomics is passed on to the Decision-Making, which implements the State Machine M_{LTU} . The chart is found in Figure 5.12 and follows the derivations, found in Sections 3.4.1.2 and 3.4.3 of Chapter 3.

The provided chart in Figure 5.12 implements a variable starting state. This is in order to allow for an in-flight activation of the Nominal system. Thereby, the evaluation, found in $init_LTU_lgx$ facilitates a correct initial state with regards to the powered-lift automation and depends on the status of the previous system in command. The possibility for an in-flight switch to the Nominal system is in the scope of separate procedure definitions and hence the evaluation is not in the scope of this thesis. For the sake of simplicity, here it can be assumed that $init_LTU_lgx = Engaged$, facilitating the starting state, found in Equation 3.2 found in Chapter 3.

Apart from the initial state specification, the chart exactly follows the derivations of Sections 3.4.1.2 and 3.4.3. The transition conditions for normal and abnormal entry to the powered-lift mode, i.e. the transition from *Engaging* to *Engaged* as per Equations 3.25 and 3.25 respectively are established with a logical “or” as evident from Figure 5.12. Furthermore, the chart considers the provisions for high-lift system operation as per Equations 3.57 and 3.58 that cause the transitions from *Disengaged* or *Disengaging* to



(a) Powered-Lift Mode Decision-Atomics



(b) Powered-Lift Mode Decision-Making

Figure 5.13: Powered-Lift Mode Selection

Engaging respectively. Provided the system is not equipped with a high-lift system, the bracketed contents in the transition conditions in the chart of Figure 5.12 can simply be removed.

The output of M_{LTU} 's implementation is the state, denoted with LTU_lgx . It is passed to the automation's Decision-Execution together with all other sequential and combinational logic used for the generation of the data, required by the surrounding systems within the behavioral specification model.

5.5.2 Powered-Lift Mode Selection of the Nominal System Law

For recollection, the Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics requires explicit knowledge of the flight state with regards to the aircraft kinematic speed. More precisely, when the control inceptor is in the hover region \mathbb{H} , the aircraft ground speed needs to be controlled in order to enable high-precision landing, take-off and near-ground maneuvering. The differentiation between the aircraft flight states is established by the automation as previously presented in Equation 3.59 of Chapter 3. For this, the state of the the State Machine M_{LTU} and an additional automation logic is used. The latter is s_{HS} and was presented in Section 3.4.1.2. The implementation of this logic within the behavioral specification model is visualized in Figure 5.13.

Subfigure 5.13a demonstrates the Decision-Atomics generation of s_{HS} . The two signals are used to control the state of the Latch and are performed in accordance with Equations 3.33 and 3.32. There, the evaluation of the kinematic speed is as per Equations 3.12 and 3.11 respectively.

For recollection, the comparison needs to be done with relation to the upper kinematic speed boundary V_{HOVER} . However, in Subfigure 5.13a, an additional robustness criteria is added. It is necessary to ensure that state changes and frequent mode switches are avoided. Hence, a hysteresis is made with regards to the kinematic speed as visible in the “compare to constant” elements of Subfigure 5.13a. The same is performed for the pilot input, captured in the *throttle_above_detent_flg* by means of a “relay” in the pilot data process, found in the common functions within the behavioral specification model. A depiction has been omitted for the sake of readability.

Subfigure 5.13b demonstrates the sequential logic that drives the state s_{HS} . It is done in accordance with Equation 3.34 and implements the Latch method as found in Equation 2.9 in Chapter 2. One addition with regards to the Latch definition in Equation 2.9 is the provision for a variable starting state, visible in Subfigure 5.13b. This is in order to allow for an in-flight activation of the Nominal system. The evaluation, found in *init_LTU_lgx*, facilitates a correct initial state with regards to the powered-lift automation and depends on the status of the previous system in command. The possibility for an in-flight switch to the Nominal system is in a scope of separate procedure definitions and hence the evaluation is not in the scope of this thesis.

5.5.3 High-Lift System Operation

The powered-lift system automation module presented in Section 5.5.1 demonstrated the provisions that consider the operation of the high-lift system. In this section, the automation of the flaps is presented. It follows the methods, derived in Section 3.4.2 of Chapter 3.

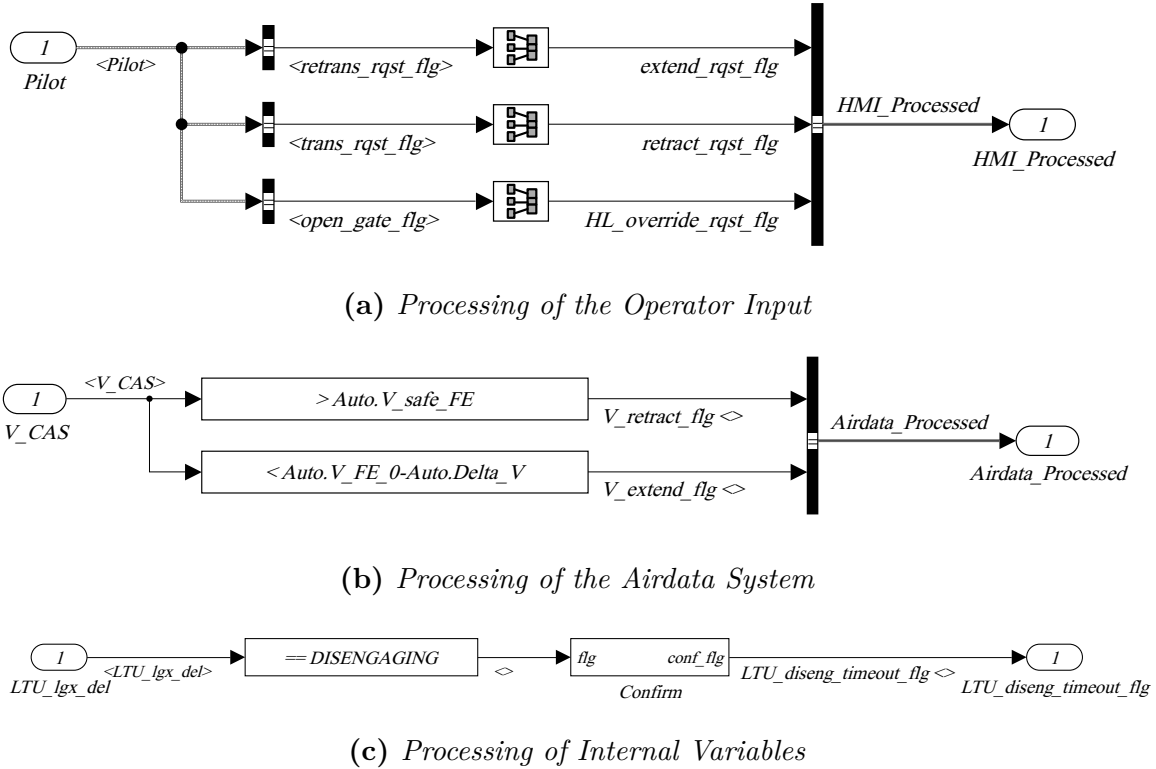
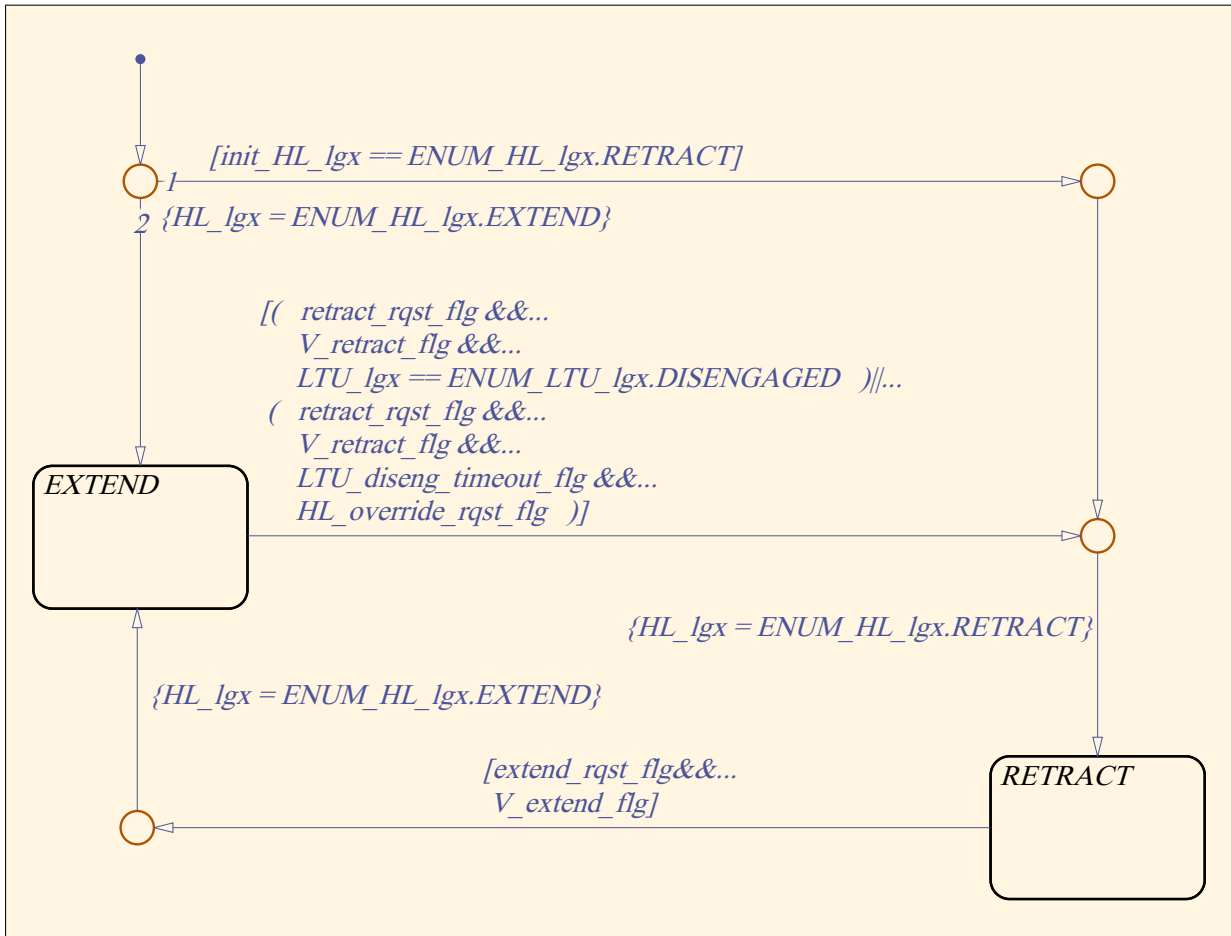


Figure 5.14: *The Decision-Atoms of the High-Lift System Automation*

The Decision-Atoms module of the High-Lift system automation is visible in Figure 5.14. The figure is organized as the individual parts of Section 3.4.2.1 in Chapter 3 and therefore divided by the signal source.

Subfigure 5.14a evaluates the pilot inputs via the HMI. For recollection, they communicate the desired mode of operation to the automation module. As visible in the figure, the first two input variables are calculated as per Equations 3.40 and 3.41 and are taken as-is from the processing previously depicted in Figure 5.7 found in Section 5.4. The last input - $HL_{override_rqst}$ - is necessary in the abnormal event where the LTU disengagement fails and a mitigation is in effect. The strategy is explained in detail in Chapter 3. The origin of the signal is the basis for the Nominal and Fallback system procedure harmonization and is derived in Equation 4.33 in Chapter 4. In addition to this, it is coupled with a temporal check, explained later in this section.

The processing of the operator input is followed by the evaluation of the flight conditions for the high-lift system operation. This is visualized in Figure 5.14b. The two boolean values are calculated as per Equations 3.42 and 3.43. For recollection, the latter ensures that the extension of the high-lift system can only commence when the structural integrity of the aircraft is ensured. Prior to retraction (i.e. in the probable proximity to the ground), the former check ensures obstacle avoidance until the final take-off configuration is initiated.

Figure 5.15: M_{HL} Implementation

Lastly, Subfigure 5.14 depicts the internal variable processing. The variable is needed in the event where a functional and undetected failure in an LTU is in effect because of which the disengagement of the powered-lift system is no longer possible. The timeout becomes *true* if the disengagement duration exceeds the known LTU ramp down command by the control allocation and achieves two properties. First, it produces a warning via the HMI and informs the operator that a mitigation strategy must be initiated. Secondly, it allow the automation to transition to the state *Retract* if requested by the operator via $HL_{override_rqst}$ explained in the previous paragraphs of this section.

This concludes the Decision-Atomics of the Nominal system high-lift automation module. The information from the Decision-Atomics is passed on to the Decision-Making, which implements the State Machine M_{HL} . The chart is found in Figure 5.15 and follows the derivations, found in Section 3.4.2.2 of Chapter 3.

The provided chart in Figure 5.15 implements a variable starting state. This is in order to allow for an in-flight activation of the Nominal system. Thereby, the evaluation, found in $init_HL_lgx$ facilitates a correct initial state with regards to the high-lift automation and depends on the status of the previous system in command. The possibility for an in-flight switch to the Nominal system is in a scope of separate procedure definitions and

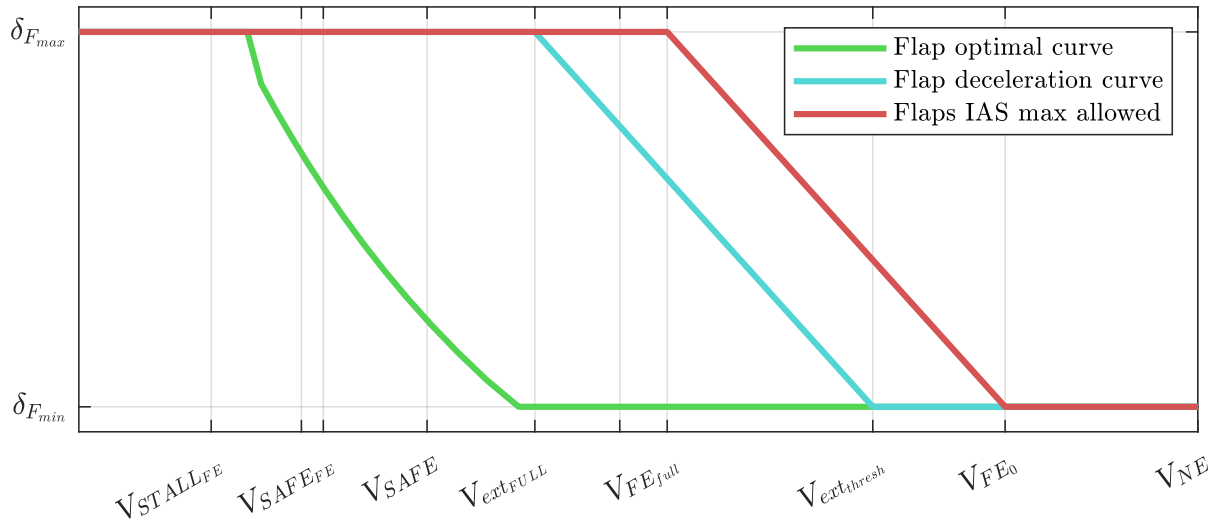


Figure 5.16: *High-Lift System Scheduling over the Airspeed and Automation Mode*

hence the evaluation is not in the scope of this thesis. For the sake of simplicity, here it can be assumed that $init_HL_lgx = Extend$, facilitating the starting state, found in Equation 3.36 found in Chapter 3.

Apart from the variable starting state provisions, the implementation visible in Figure 5.15 implements the transition conditions exactly as derived in Equations 3.46, 3.48 and 3.50. The normal and abnormal transition conditions that cause the change of the state from *Extend* to *Retract* are summarized in a logical “or”. It must be noted that the logical operation can be optimized. For the behavioral specification model this is not performed for the sake of readability and consistency with Equations 3.46 and 3.50. However, later on for application in flight software, such optimizations must be conducted.

The output of the chart HL_lgx specifies the state of M_{HL} . It is passed to the automation’s Decision-Execution and together with all other sequential and combinational logic used for the generation of the data, required by the surrounding systems within the behavioral specification model. Of particular interest are the commands, supplied by the Nominal system to the high-lift system. Depending on the state of the automation, these commands differ as previously explained in Section 3.4.4.3.

The scheduling of the flap commands is depicted in Figure 5.16. The important velocities are denoted on the x-Axis. The never exceed speed as a function of the high-lift system deflection is depicted in red. Whenever the state of the automation is *Extend*, then the function, depicted in blue in Figure 5.16 is tracked and the corresponding command value is forwarded to the high-lift system. This corresponds to the derivations, found in Equation 3.70. Thereby, the drag is maximized, allowing for a faster retransition. At the same time, a buffer from the never exceed speed is maintained in order to ensure the structural integrity in the cases of short-term disturbances that may cause the airspeed to

increase abruptly. In a similar manner, the state of the automation *Retract* forwards the mapping, visualized in green in Figure 5.16. For recollection, this mapping minimizes the drag and is computed in preprocessing in accordance with Equation 3.68.

In terms of implementation, the command scheduling is realized with a switch that chooses between the two different command mappings. The mappings are implemented by means of lookup tables that accept the aerodynamic velocity as an input. This part of the Decision-Execution is omitted here for the sake of readability.

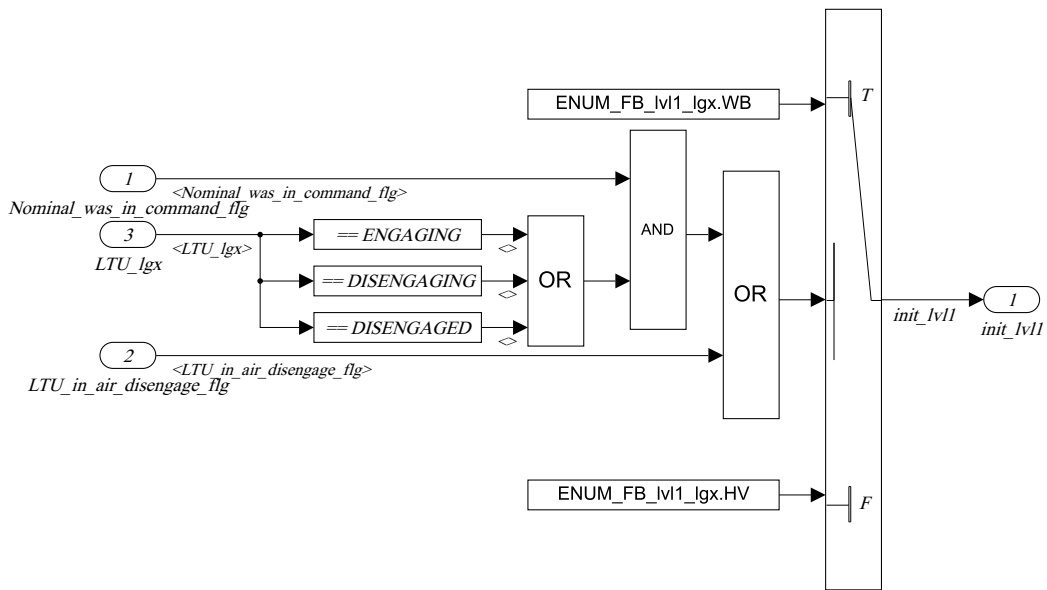
In addition to this, the Decision-Execution of the Nominal system automation's behavioral specification model includes coordination and data supply to the law and control allocation, the scheduling of the airspeed limits and the signal generation for the cockpit indication items. The considerations for each of the above-mentioned topics was discussed in Sections 3.4.4.1, 3.4.4.2 and 3.4.4.4 of Chapter 3 respectively. The implementation methods are, however, omitted here in the interest of readability due to the high interface dependency on the surrounding modules and additional procedural considerations that are outside the scope of this thesis.

5.6 Fallback Automation

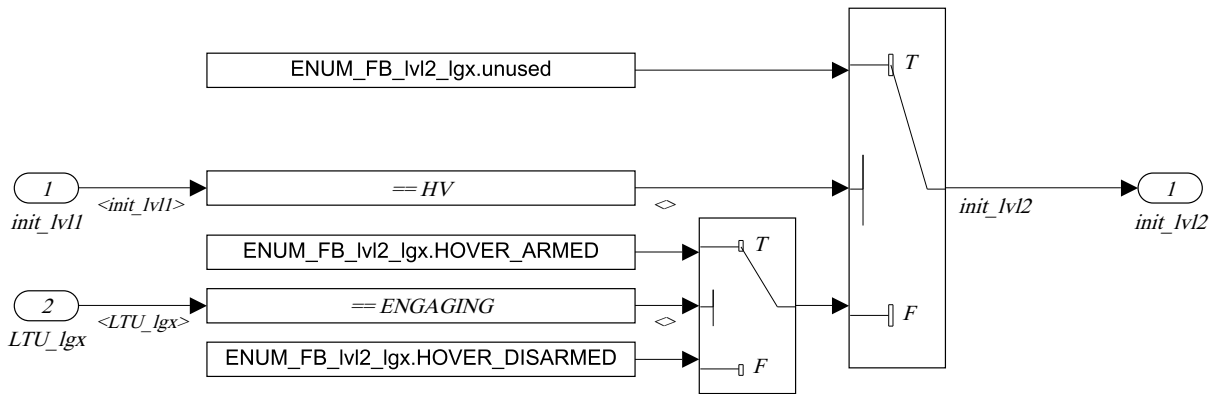
Section 5.5 presented the the behavioral specification modeling of the Nominal system automation. This section does the same for the Fallback system. As per Section 5.2, it is allocated to the Fallback System Repository and as per Section 5.3 it operates within the Law Automation integration module.

Among others, the Fallback system automation includes the management of the Fallback control concept by the pilot. Thereby, the powered-lift system operation is specified. In terms of the transition and retransition, the behavioral specification model of the Fallback system automation follows the concepts, derived in Chapter 4. It thereby assures consistency in the procedures during transition and retransition with both Fallback and Nominal system. Furthermore, a capability for correct state initialization following a takeover is provisioned.

This section focuses on the main tasks of the automation and their implementation. It is organized as follows. Section 5.6.1 demonstrates the implementation of the *Initialize* function, responsible in ensuring starting state correctness following a reversion to the Fallback system due to a failure in the Nominal system. This is done in accordance with Section 4.2.3 of Chapter 4. Next, the Fallback system automation's Decision-Atomics module is presented. This is done in Section 5.6.2 and is consistent with the derivations of Section 4.2.2 of Chapter 4. This is followed by Section 5.6.3, in which the Simulink chart that implements the Fallback State Machine M_{FB} is demonstrated.



(a) First Level Starting State Selection



(b) Second Level Starting State Selection

Figure 5.17: Initialize Function Implementation

5.6.1 Takeover Starting State Calculation

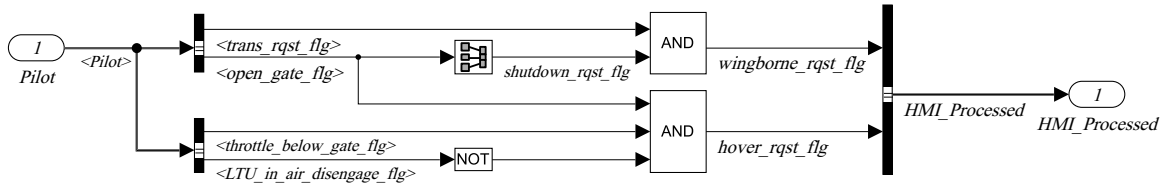
In order to facilitate a takeover with the Fallback system in the correct automation mode, in Section 4.2.3 of Chapter 4, the function *Initialize* was introduced. This specifies the starting states of the Fallback system's State Machine M_{FB} . This is done in accordance with Equation 4.18, found in Chapter 4. The implementation of the *Initialize* function is illustrated with Figure 5.17.

For recollection, from a theoretical standpoint, a State Machine may only have one starting state or state tuple. Because the State Machine M_{FB} is part of a much larger automation module, however, this automaton is actually enabled only when the Fallback system is in command. This was explained in more detail previously in this chapter in Section 5.3.4.1 and visualized with Figure 5.5. Therefore, a variable initial state of this particular state machine is allowed. This is explained in more detail in Appendix D.

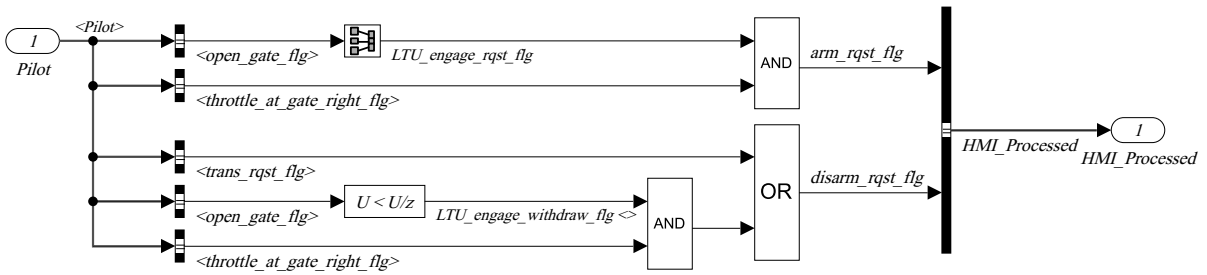
The implementation in this exemplary behavioral specification model follows the derivations, found in Table 4.3. Thereby, Subfigure 5.17a illustrates how the selection of the first starting state is evaluated. The output corresponds to the third column of the specification in Table 4.3.

As visible in the Subfigure 5.17a, apart from the known dependency on the state of Nominal system automation s_{LTU} and the evaluation whether that system is in command, an additional condition is introduced, summarized in the signal $LTU_in_air_disengage_flg$. This variable is an additional operator input, necessary in abnormal events where the powered-lift system should be prohibited following the transition into wingborne flight. It is used to communicate to the automation that the engagement of the LTUs should not be executed despite the movement of the throttle inceptor into the transition/retransition command region. Thus, the usage of this input item enables the wingborne landing of the aircraft. The input item is utilized within a separate aircraft procedure and both input item and procedure are not in the scope of this thesis.

Similarly, Subfigure 5.17b demonstrates the second starting state selection. The output corresponds to the fourth column of the specification in Table 4.3. Comparing the implementation in the subfigure and the table contents, it is visible that the evaluation is performed differently. For example, Table 4.3 defines the conditions, under which the starting state $s_{FB|2}$ may be *unused* using the state s_{LTU} whenever the Nominal system is operational. In the provided implementation this is omitted as it is performed previously for the starting state of the first level $init_lvl1$ in an identical manner. Therefore, this variable is utilized instead for the sake of simplicity.



(a) First Level Decision-Atomics: Pilot Input Processing



(b) Second Level Decision-Atomics: Pilot Input Processing

Figure 5.18: Fallback System Control Mode Selection Decision-Atomics

5.6.2 Decision-Atomics

This section presents the Decision-Atomics of the Fallback system automation’s behavioral specification model. It follows the derivations found in Section 4.2.1 of Chapter 4. For recollection, one main objective of the Fallback automation concept is to ensure maximum operator authority throughout the aircraft flight envelope. This is because following a takeover the operator alone bears the responsibility of maintaining safe flight.

However, this also allows for robust automation design due to the possibility to reduce the amount of sensor information necessary for the automation operation. As evident in Section 4.2.1, the scope of the Decision-Atomics of the Fallback concept is solely the processing of the crew input via the Human-Machine-Interface. The implementation of the Decision-Atomics of the Fallback automation behavioral specification is depicted in Figure 5.18.

As evident in the two subfigures, the behavioral specification modeling of the Decision-Atomics distinguishes between two separate signal processing elements. For recollection, the derivations in Section 4.2.1 do not differentiate with respect to the layer of State Machine. However, the implementation methods of multilevel Finite-State Automata requires the separation of the individual layers as prescribed in [77, 83]. Therefore, from

an implementation point of view it is beneficial to further split the signal according to the separate levels. Subfigure 5.18a hence provides the Decision-Atomics for the first level of M_{FB} , whereas Subfigure 5.18b does so for the second level.

The computation for the wingborne control mode activation command $wingborne_{rqst}$ in Subfigure 5.18a follows Equation 4.6. In the equation, the input variable $trans_{rqst}$ that is calculated as per Equation 4.5 is taken as-is from the processing previously depicted in Figure 5.7 found in Section 5.4. For recollection, the variable $shutdown_{rqst}$ is chosen such that harmonization between the procedures between Nominal and Fallback system is achieved. It is hence chosen in accordance with Equation 4.33.

The same harmonization considerations apply to $hover_{rqst}$. It is therefore computed in accordance with Equation 4.35. However, as evident in Subfigure 5.18a, an additional condition is introduced, summarized in the signal $LTU_in_air_disengage_flg$. This variable is an additional operator input, necessary in abnormal events where the powered-lift system should be prohibited following the transition into wingborne flight. It is used to communicate to the automation that the engagement of the LTUs should not be executed despite the movement of the throttle inceptor into the transition/retransition command region. Thus, the usage of this input item enables the wingborne landing of the aircraft. The input item is utilized within a separate aircraft procedure and both input item and procedure are not in the scope of this thesis.

The Decision-Atomics of the second level of the State Machine presented in Figure 5.18b generates the remaining two signals that are required by the charts. More precisely, the variable that triggers the activation of the LTUs to the idle setting arm_{rqst} is computed as per Equation 4.7. It utilizes the processing previously depicted in Figure 5.7 found in Section 5.4. In addition, it includes the harmonization considerations, included with $LTUengage_{rqst}$ and found in Equation 4.34. For recollection, the last variable - $disarm_{rqst}$ - is necessary in the event of mitigation strategies. This was previously explained in detail in Chapter 4. The computation of this signal follows Equation 4.8, in which $LTUengage_{withdraw}$ is in accordance with the considerations, derived with Equation 4.36.

This concludes the Decision-Atomics of the Fallback system control mode selection. The information from the Decision-Atomics is passed on to the Decision-Making, which implements the State Machine M_{FB} . The State Machine architecture is presented in the next section and follows the derivations, found in Section 4.2.2 of Chapter 4.

5.6.3 Decision-Making

The Fallback automation fulfills multiple tasks. Among others, it is responsible for the management of the law. This includes specifying the control mode in terms of level of automation but also in terms of allowed effector usage and flight state. The latter includes considerations on the transition and retransition from powered-lift to wingborne mode and

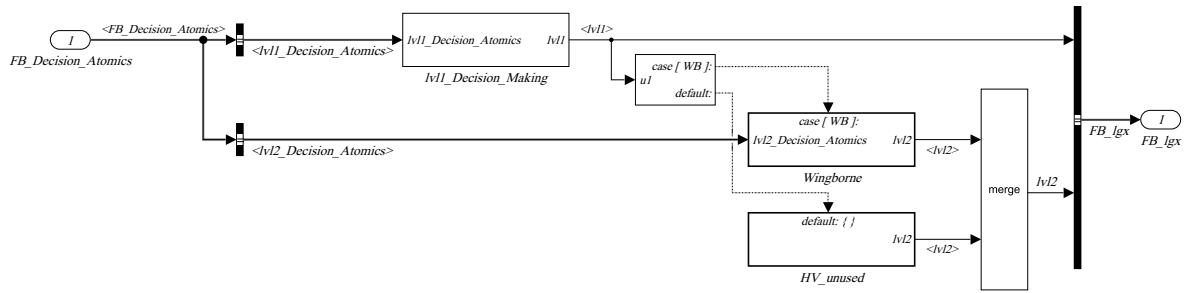


Figure 5.19: *The Multi-Level Finite State Machine Architecture that Implements M_{FB}*

back. The methods for this were in the scope of this thesis and presented in Chapter 4. This section presents the Decision-Making process of the exemplary behavioral specification model.

For recollection, the State Machine used for the transition and retransition with the Fallback system was presented in Section 4.2.2 of Chapter 4. As explained there, the Mealy machine used contains two levels. In order to produce an implementation solution that adheres to the the architecture presented in Section 4.2.2, the workflow derived in [77, 83] is utilized and visualized in Figure 5.19.

The methods in [77, 83] use a mixture of Simulink and Stateflow constructs that allow for the design of Finite-State Automata that comply with rigorous demands for high-integrity software as found in [82] and more. It must be noted that the automation behavioral specification model need not be developed with the workflow, derived in [77, 83]. However, this method is nonetheless utilized due to the easier ability to transfer the design later on for embedded software applications where demands such as code compliance are relevant.

In Figure 5.19 the first level of the State Machine is located within the subsystem on the left hand side. This automaton is explained later on in this section. The correct chart for the second level is called using a switch-case based on the state of the first level. For the sake of readability, here this selection is on the same level unlike in the methods, introduced in [77, 83]. In addition, a bus creator is used instead of a bus assignment for the same reason. The inputs fed to both first and second level of the State Machine originate from the Decision-Atomics as evident from the figure.

The first level of the State Machine that implements M_{FB} can be seen in Figure 5.20. From an architectural point of view, it resembles the depiction in Figure 4.1 of Chapter 4. The starting state selection in accordance with the initialization function *Initialize* as described previously in Section 5.6.1. It is implemented with the top-most nodes in the figure.

One used signal that was not explained previously is *force_lv11* that is of the same data type as the chart enumeration type. This signal is used for on-ground operation prior to the engagement of the law. It is used to communicate to the automation the

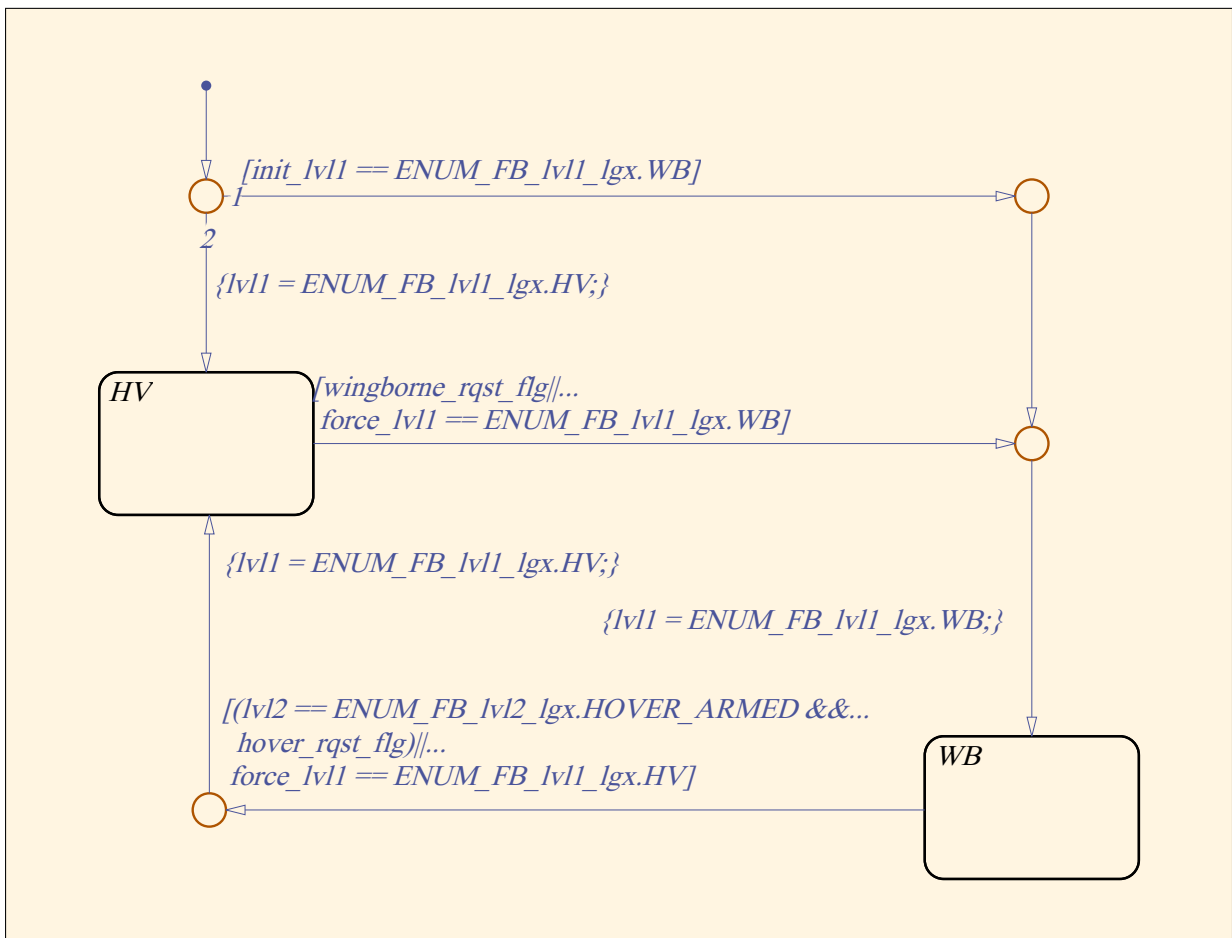


Figure 5.20: *Fallback Automation State Machine First Level*

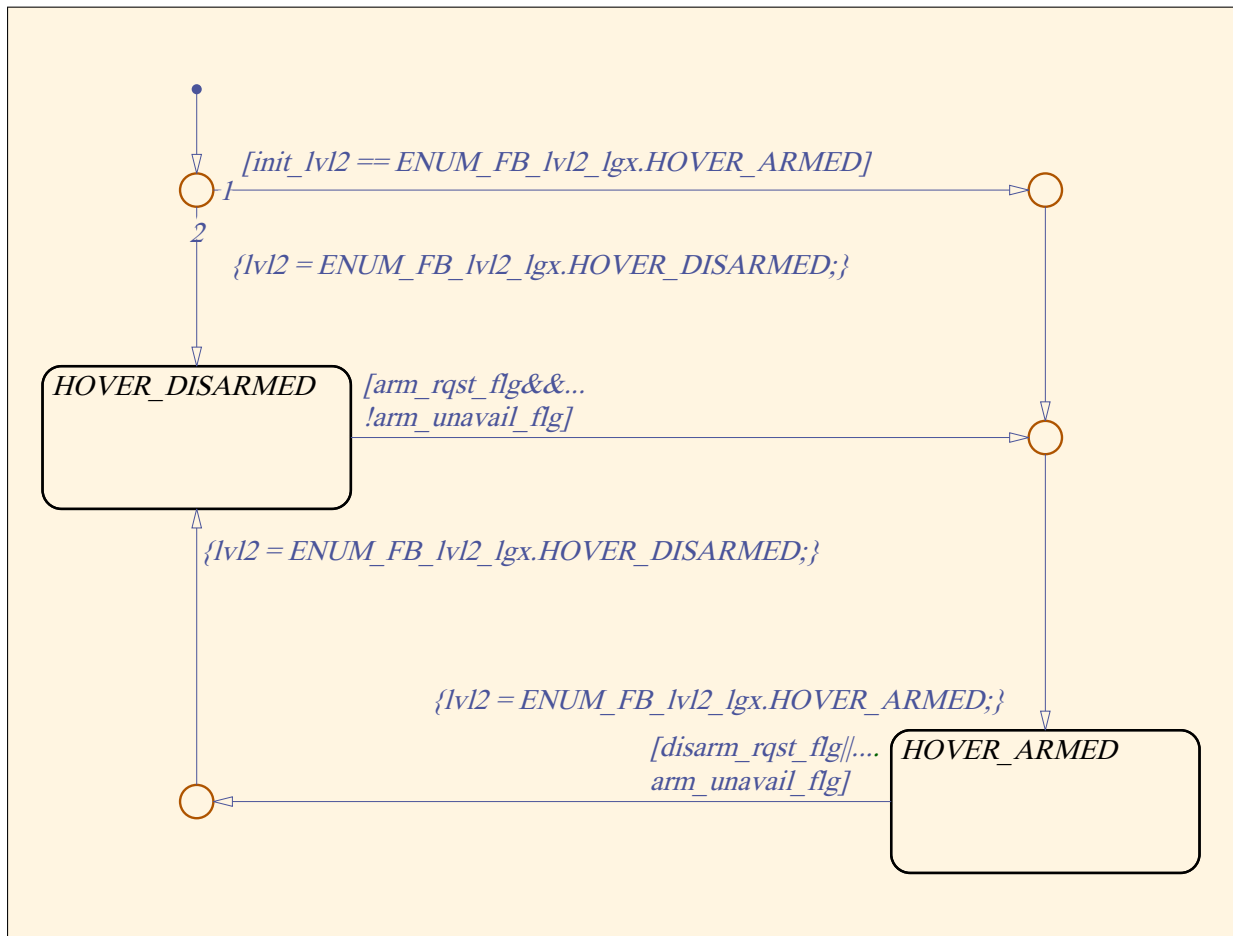


Figure 5.21: *Fallback Automation State Machine Second Level*

way the control surfaces need to be utilized by the Fallback DRM. This allows for the implementation of a direct law while the control mode is not fully engaged and is therefore part of procedures that are related to preflight checklists. It is therefore not in the scope of this thesis.

Apart from *force_lv11*, the mechanics of the chart follow the ones, explained in Section 4.2.2 of Chapter 4. The input data originates from the Decision-Atomics previously introduced in Section 5.6.2. What must be noted here is that the condition that implicates the transition from *Wingborne* to *Hover* includes the state of the second level of the State Machine. This differs from the previously derived transition condition found in Equation 4.12. The additional utilization of the state of the second level is necessary due to the implementation method itself. As visible in both Equation 4.12 and Figure 4.1, the transition from *Wingborne* to *Hover* is only permissible if $s_{FB|2}$ is *HoverArmed*. This condition must be explicitly modeled in the chart found in Figure 5.20.

The state first level State Machine is passed to the switch case block which enables two switch case action subsystems as seen in Figure 5.19. Provided the system is in the *Hover* state, then for the second level the state *unused* is assigned as per the lower subsystem

visible in the figure. This is consistent with the assignments found in Table 4.1 and Figure 4.1 of Chapter 4. Otherwise, for the state *Wingborne*, the second level State Machine is called and its implementation is depicted in Figure 5.21.

The signal *arm_avail_flg* seen in Figure 5.21 is a variable resulting from multiple pilot inputs, necessary in abnormal events where the powered-lift system should be prohibited following the transition into wingborne flight. It is used to communicate to the automation that the engagement of the LTUs should not be executed despite the movement of the throttle inceptor into the transition/retransition command region. Thus, the usage of this input item enables the wingborne landing of the aircraft. The input item is utilized within a separate aircraft procedure and both input item and procedure are not in the scope of this thesis.

Similar to the chart for the first level, the starting state selection is in accordance with the initialization function *Initialize* as described previously in Section 5.6.1. It is implemented with the top-most nodes in the figure. Apart from *arm_avail_flg*, the mechanics of the chart follow the ones, explained in Section 4.2.2 of Chapter 4. The input data originates from the Decision-Atomics previously introduced in Section 5.6.2.

As explained previously in Chapter 4, the operation of the high-lift system is within the responsibility of the pilot. However, the Fallback system includes a protection function that retracts the flaps in the event where a critical airspeed is exceeded. This was done according to Equation 4.23. The scheduling of the command upper limit of the high-lift system with regards to the airspeed is as per Equation 4.21. For the sake of simplicity, the linear interpolation of the last line in Equation 4.21 is chosen to be the same as for the drag maximization with Nominal system as depicted previously in blue in Figure 5.16. Thereby, the same buffer that ensures the structural limit speed is not exceeded due to short-term disturbances.

In addition to this, the Decision-Execution of the Fallback system automation's behavioral specification model includes coordination and data supply to the law and control allocation, the scheduling of the airspeed limits and the signal generation for the cockpit indication items. The considerations for each of the above-mentioned topics was discussed in Section 4.2.4 of Chapter 4. The implementation methods are however omitted here in the interest of readability due to the high interface dependency on the surrounding modules and additional procedural considerations that are outside the scope of this thesis.

5.7 Conclusion

This chapter presented a methodology for the validation of automation function behavior within the scope of the whole aircraft operation. The proposed solution is largely system architecture-agnostic and can therefore be applied in early stages of the product development cycle. This is made possible using the so-called behavioral specification model presented in this chapter. The specification model recreates the automation functions

in a highly simplified but representative environment and enables the simulation of the aircraft operation prior to the full functional development. Therefore, it allows for efficient validation of the aircraft ConOps and for fast adaptations in the events where changes to the specification are necessary. It thereby advances the state of technology in accordance with **Contribution 3**.

The behavioral specification model was utilized for multiple design and validation efforts at the TUM Institute of Flight System Dynamics. Among others, the method proved useful in identifying missing functionality and pinpoint improvement potential in both control and operational concept. In the next paragraphs selected examples of this are provided.

Takeover Analysis

In order to reproduce executable control concepts within the behavioral specification model, the DRM method is utilized. How this construct is embedded into the behavioral specification model is explained previously in Section 5.1.3.

In addition to this, the analysis and subsequent functional partition of the DRM method into control part and simplified plant modeling allowed for the integration of multiple control concepts within a single behavioral specification model. As a consequence, the inclusion of additional automation functions could be included, with which a change between the control concepts during flight is facilitated. Section 5.6.1 of this chapter provided an example of how this can be achieved.

The above-mentioned property and the correct variable initialization of the DRM method allow for concepts such as a takeover due to the fallback principle to be tested. Clearly, a switch from one control concept to another introduces a transient in the system response. The magnitude of such transients is especially exacerbated whenever this occurs without the pilots request but automatically instead. This is due to the delayed operator reaction following a change of control concept. Such reaction times are mentioned in [122, 133] and could be significant if the current attention of the pilot is directed elsewhere (for example during cruise flight).

The behavioral specification was used at TUM-FSD for the analysis of the system response following a takeover. During the evaluation of the transients it was discovered that under certain situations the stabilization of the aircraft following a fault in the Nominal system was impossible without the use of proper operator input filtering. The analysis and the solution for the mentioned issues were performed by Daniel Gierszewski of TUM Institute of Flight System Dynamics.

Functional Monitor Design

The high-degree of automation for the Nominal system implies high complexity in the deployed functions. In addition, the functional design relies heavily on aircraft parameters that are prone to uncertainties. As a consequence, a failure in the Nominal system may be caused due to undiscovered failures in the design.

In order to address this, for the exemplary aircraft found in Section 1.1.4 of Chapter 1 the so-called functional monitor is under development at TUM-FSD. The functional monitor offers checks of the correctness of the Nominal system execution by independent evaluation of the pilot intentions, the aircraft response and the state of the automation. It therefore provides runtime assurance capabilities to the Nominal system operational concept. This was explained briefly in Section 5.2.1.

The developed behavioral specification model was used for the conceptual design of the functional monitor for the Nominal system. Analysis and testing of the aircraft response and the utilization of the failure injections enabled the selection of the main error detection mechanisms. The development of the functional monitor is performed by Hannes Hofsäß of the TUM Institute of Flight System Dynamics.

Cockpit Indications

The behavioral specification model covers the complete aircraft operation. This is, on the one hand, facilitated by the incorporation of automation functions that enable the execution of the aircraft operational concept. On the other hand, the inclusion of the DRM method into the behavioral model design allows for a representative description of the control concept and handling qualities. As per Section 5.1.3, both are modeled with a relatively high degree of fidelity.

This capability has allowed for the validation of the interaction concept between law and automation on one side and the pilot on the other. The system feedback is communicated to the operator via cockpit indication. Therefore, the indication items could be tested and validated for their intuitiveness and readability.

In addition to this, the simulation of the behavioral specification model allowed for the identification of additional useful indication items that would otherwise have been discovered much later during the functional development. Among others, those include the warning and cautions associated with the transition and retransition that were mentioned in Chapters 3 and 4.

Pilot-In-The-Loop Testing

At the TUM Institute of Flight System Dynamics, an eVTOL simulator is under development. The purpose of the simulator is to conduct Pilot-In-The-Loop validation of both operational and control concept in a representative environment. In that regard, the behavioral specification model presented here provided an efficient solution in the

commissioning of the eVTOL cockpit simulator at TUM-FSD. The reason for this was the possibility to rapidly reproduce the aircraft operational concept with little dependency on the system architecture and prior to the full functional development.

The different stages of the developed cockpit are visible in Figure 5.22, where the initial design can be found on the upper left and the current state of technology - on the bottom. Each development stage included altered operational concepts or additional functionality and capability. Therefore, the high degree of rapid prototype of the behavioral specification model proved useful for the fast commissioning of the simulators.

So far the TUM-FSD eVTOL simulator and therefore the behavioral specification model has been used at the institute for the conduction of mission task elements that evaluate the aircraft handling qualities and the operational concept. In addition the specification model has been used for the creation of pilot checklists. They can be found in Appendix G.

Formulation of Automation Function Requirements

The artifacts of the behavioral specification model can be utilized outside the scope of the validation and simulation activities. For example, the resulting automation sequential logic found in Figure 5.5 is the product of the functional decomposition of the automation tasks and is utilized for the design of the software architecture of the automatic functions.

Section 5.1.2 mentioned the explicit centralization of otherwise distributed algorithms for the sake of work effort reduction. These centralized functions are the basis for the requirement specification of the later centralized function design. The decentralized algorithms are designed and tested using their centralized specification.

Transition and Retransition Automation

The exemplary behavioral specification model of this thesis includes the application of derived methods from Chapters 3 and 4 of the thesis and therefore the transition and retransition procedures and the underlying automation. They were presented in Sections 5.5 and 5.6 respectively, where Section 5.4 demonstrated the common items shared among them. As discussed in Section 5.1.3, these functionalities are modeled at a high level of fidelity.

Thus, the behavioral specification model can be used for the validation of the transition and retransition solutions developed in this thesis. The next chapter provides simulation results of namely these validation efforts.



Figure 5.22: *The Development Stages of the eVTOL Simulator of TUM-FSD*

Chapter 6

Simulation Results

The framework presented in Chapter 5 is utilized for the validation of the closed-loop aircraft behavior. This chapter summarizes the simulation results of selected scenarios using the behavioral specification. Thereby, the chapter focuses on the methods responsible for the automation of the aircraft transition and retransition, developed in Chapters 3 and 4 of this thesis.

A mission profile that conforms with the SC-VTOL regulatory effort is utilized for the generation of all results in this chapter. Figure 6.1 visualizes the mission profile. The results, found in the graph, are from the flight using the Nominal system. During flight with the Fallback system, the results are similar. The different segments of the mission are numbered in Figure 6.1.

The depicted mission begins with the activation of the FCS and is followed by a vertical take-off until the TDP at $20ft$. Subsequently, the aircraft is accelerated until V_{TOSS} at that height. Upon reaching the desired speed, a climb is initiated in accordance with segment one of the SC-VTOL profile which continues until $200ft$ as prescribed by the standard. There, a transition is performed as suggested in Section 4.4.5. This is denoted with a change in color coding in Figure 6.1, whereas the colors of the different flight phases can be seen in the legend.

Upon reaching wingborne flight - depicted in green - the second segment of the mission profile is initiated. As seen in both Chapter 3 and 4, after achieving wingborne flight the high-lift system is retracted. Apart from that, no change in the automation methods in those chapters is observed. For this reason and for the sake of visibility in Figure 6.1, the second segment here is concluded at $400ft$ instead of at the prescribed $1000ft$.

The mission profile in Figure 6.1 includes a short cruise flight upon concluding the take-off trajectory. This is followed by an approach at the prescribed sink rate by the MOC SC-VTOL. At a height of approximately $200ft$, the sink rate is stopped in order to perform the retransition which is denoted with orange Figure 6.1. Thereupon, the approach and vertical landing are conducted in the powered-lift mode. It can be noticed

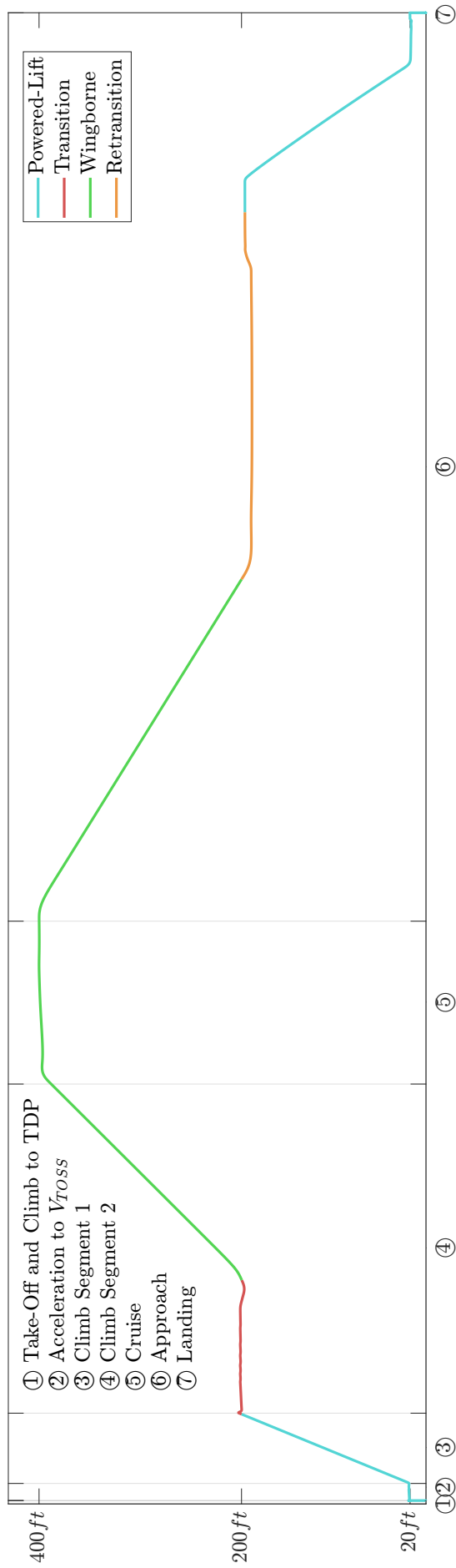


Figure 6.1: *Mission Profile*

that the transition requires noticeably less distance to execute than the retransition. This is due to the aerodynamic efficiency of the aircraft and the high available specific excess power of the traction system.

For the execution of the mission profile, an autopilot is implemented that is meant to substitute the operator input. The autopilot includes actions following different component errors. The implementation of the autopilot is not in the scope of this chapter. Instead, the aim of the chapter is to visualize the automation mechanisms of Chapters 3 and 4 within the whole system operation.

This chapter is organized as follows. The transition and retransition processes and automation mechanisms for the Nominal system are examined first. This section covers the fault-free scenarios. Section 6.1 presents the results of the validation effort. The focus is on the three core functionalities of the automation - the operation of the powered-lift system, the high-lift system and the management of the airspeed protections. In addition, Section 6.1 examines the possibility of a takeover by the Fallback system. It demonstrates the correctness of Fallback system initialization. Next, in Section 6.2 the Fallback system transition and retransition in the failure-free case is examined.

Abnormal scenarios are examined in Section 6.3 for both Nominal and Fallback systems. The simulation results focus on the pilot actions and the automation response. For the Nominal system, the airspeed protection management is examined in order to prove that safe flight can be maintained during the prolonged reconfiguration processes.

The chapter is concluded with Section 6.4, where the findings are summarized. Due to confidentiality, all aircraft performance and configuration parameters are not provided. Those include stall speeds, structural limit speeds, LTU operation ranges and more. Therefore, in the simulation results all graphs utilize the symbols used throughout this thesis instead of the corresponding numerical values.

6.1 Fault-Free Nominal System Transition and Retransition

This section provides simulation results of the Nominal system automation operation during transition and retransition. Section 6.1.1 illustrates the time history of the relevant signals during the reconfiguration to wingborne flight and Section 6.1.2 does so for the retransition to powered-lift flight.

The provided graphs focus on the methods, presented in Chapter 3. Namely, they demonstrate the operation of the distributed hover propulsion system, the high-lift scheduling and the airspeed protection settings during the different system modes.

6.1.1 Transition

As discussed in Section 3.5.1.1, the transition automation first disengages the LTUs and thereupon proceeds to retract the high-lift system. During every point in the process, the airspeed protections are scheduled in order to ensure a safe envelope. This section begins with a discussion about the management of the powered-lift system. The results of the simulation are found in Figure 6.2.

As previously summarized in Table 4.8 of Chapter 4, the transition process is initiated by accelerating the aircraft via the throttle control inceptor. According to the procedure in the table, this can be done by moving the stick to the gate but it should remain in the transition region. This is depicted in the upmost graph of Figure 6.2. In the plot, the outcome of the pilot input processing found in Figure 5.7 is presented. Namely, initially the pilot moves the throttle lever from the detent to the right portion of the gate (i.e. $\delta_T \in \mathbb{R}$).

Thereupon, the aircraft accelerates. This is visible from the second graph of Figure 6.2 where the calibrated airspeed is visualized. According to Table 4.8, once the stall speed is exceeded - in this case $V_{STALL_{FE}}$ due to the fully extended high-lift system - the pilot is allowed to move the throttle into the wingborne region. The movement is captured by the operator input processing found in Figure 5.8 and is visible in the top graph in Plot 6.2. In accordance with Equation 3.3, this is the request to transition to wingborne flight. The moment when the variable $trans_{rqst}$ becomes *true* for the first time is also depicted in the plot.

According to Equation 3.19, three conditions need to be satisfied in order to initiate the LTU disengagement. The first is that wingborne flight is required by the operator. This is fulfilled by the *true* evaluation of $trans_{rqst}$. In addition, the disengagement speed needs to be reached. This is described in Equation 3.53 and in this example implemented as depicted in Figure 5.9. Due to the lack of error in the high-lift system or the transition units, the disengagement speed is calculated to be $V_{SAFE_{FE}}$. Exceeding this speed is captured by the input variable V_{trans} of the state machine M_{LTU} as per Equation 3.7. The moment where this condition applies is visible in the second graph of Figure 6.2.

Lastly, in order to shut down the LTUs, the control allocation must not actively utilize them for force and moment production. This evaluation is in accordance with the Confirmation Counter of Equation 3.15, whereby the condition for starting the timer is as per Equation 3.14. In this example, the implementation of the check is provided in Figure 5.10. The counter start conditions are visible on the y -Axis of the third graph in Figure 6.2. There, the maximum RPM of all LTUs is plotted for the sake of visibility. The moment where the State Machine input LTU_{UNUSED} is valid is marked on the x -Axis of the same graph.

Once all three conditions - $trans_{rqst}$, V_{trans} and LTU_{UNUSED} - are satisfied, the State Machine M_{LTU} transitions to the state *Disengaging*. This is in accordance with Equation 3.18, implemented in Figure 5.12. This is visible in the bottom graph of Figure 6.2. In

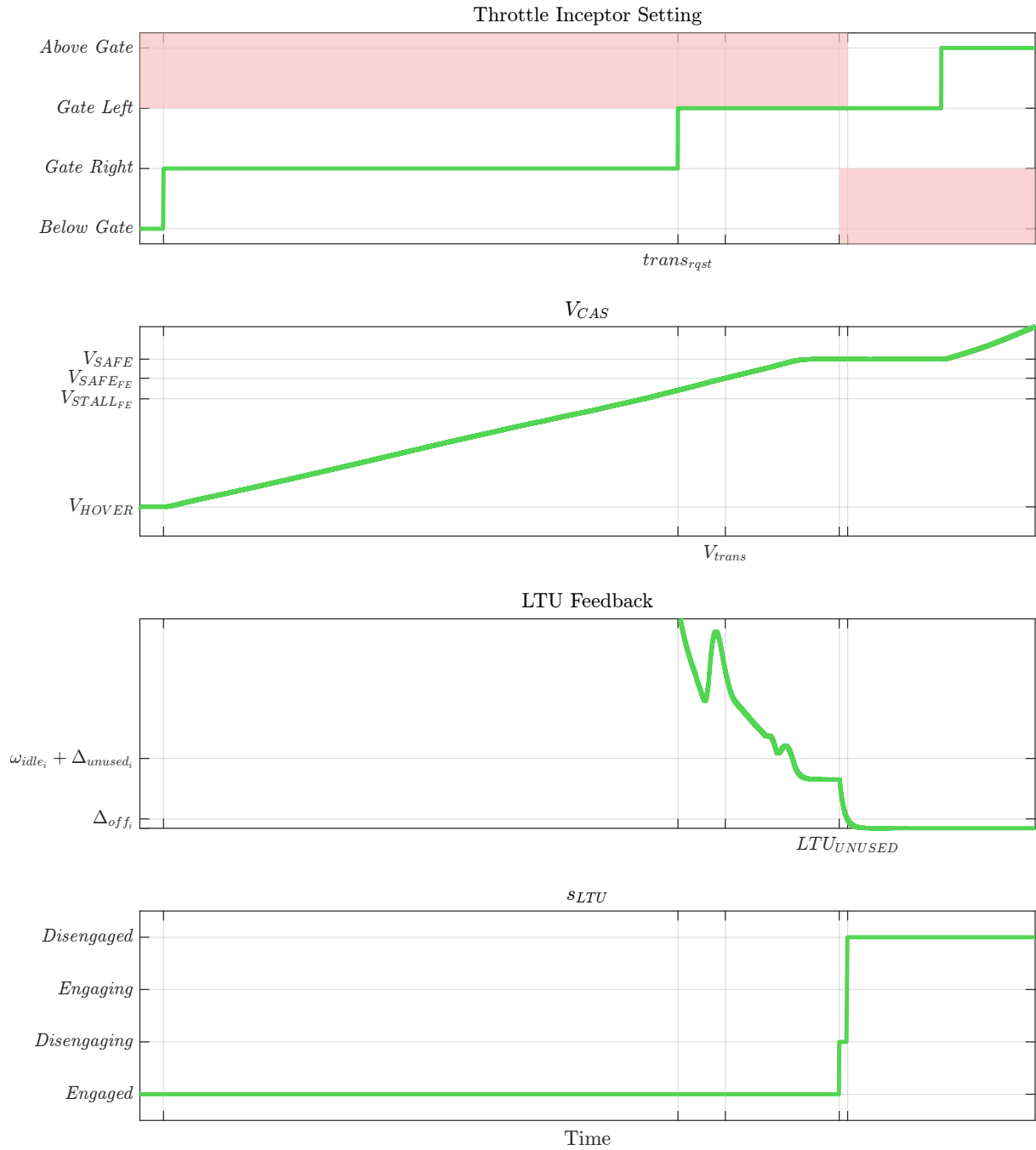


Figure 6.2: *Nominal System Powered-Lift Management Simulation Results During Transition*

this state the control allocation ramps down the LTUs to a halt. The ramp down is visible in the third plot in Figure 6.2. Once all units are evaluated to be off as per Equation 3.21 and performed as per Figure 5.10, the system transitions to the state *Disengaged* as visible in the lowest graph.

In terms of haptic feedback, the management of the barriers is as per Table 4.7 and is implemented as visible in Figure 5.6. The barrier operation during the transition is depicted in the upper graph of Figure 6.2, whereby the red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region. It is therefore visible that the powered-lift regions where low airspeeds can be commanded are inaccessible once the disengagement has been initiated. Similarly, the wingborne regions can be entered only once the correct shutdown of the LTUs has been confirmed.

Next, the management of the flaps by the high-degree of automation during transition is examined. The simulation results are available in Figure 6.3. As described previously in Chapter 3, the retraction of the flaps occurs after entering wingborne flight. By implication, this is executed after the disengagement of the LTUs. For convenience, the graphs in Figure 6.3 do not focus on the whole transition as the high-lift system management does not take any actions in the initial portion of the procedure. Instead, the graphs are zoomed to the portion, where changes in the high-lift command scheduling are observed.

As explained previously in this section, according to the transition procedure of Table 4.8, the operator moves the throttle into the wingborne region upon exceeding the stall speed. According to Equation 3.40, this processed by the automation as a request to retract the flaps. This is implemented as per Figure 5.14a and therefore the variable $retract_{rqst}$ is evaluated as *true*. The moment this occurs is indicated in the upper graph in Figure 6.3.

As visible in Equation 3.46, the retraction of the flaps in the fault-free case is initiated if three conditions are satisfied. More precisely, the change of operation needs to be commanded by the operator which is confirmed via $retract_{rqst}$. In addition to this, a safe speed needs to be reached. This is monitored via Equation 3.42 that is implemented as visible in Figure 5.14b. The moment this occurs can be seen in the second graph of Figure 6.3.

The last condition is that the LTUs are fully disengaged. For this, the state of M_{LTU} is used as visible in Equation 3.46. Once s_{LTU} is *Disengaged*, then the transition function found in Equation 3.45 is triggered. This function is implemented as per Figure 5.15. Therefore, the State Machine M_{HL} transitions to the state *Retract* as visible the last graph of Figure 6.3.

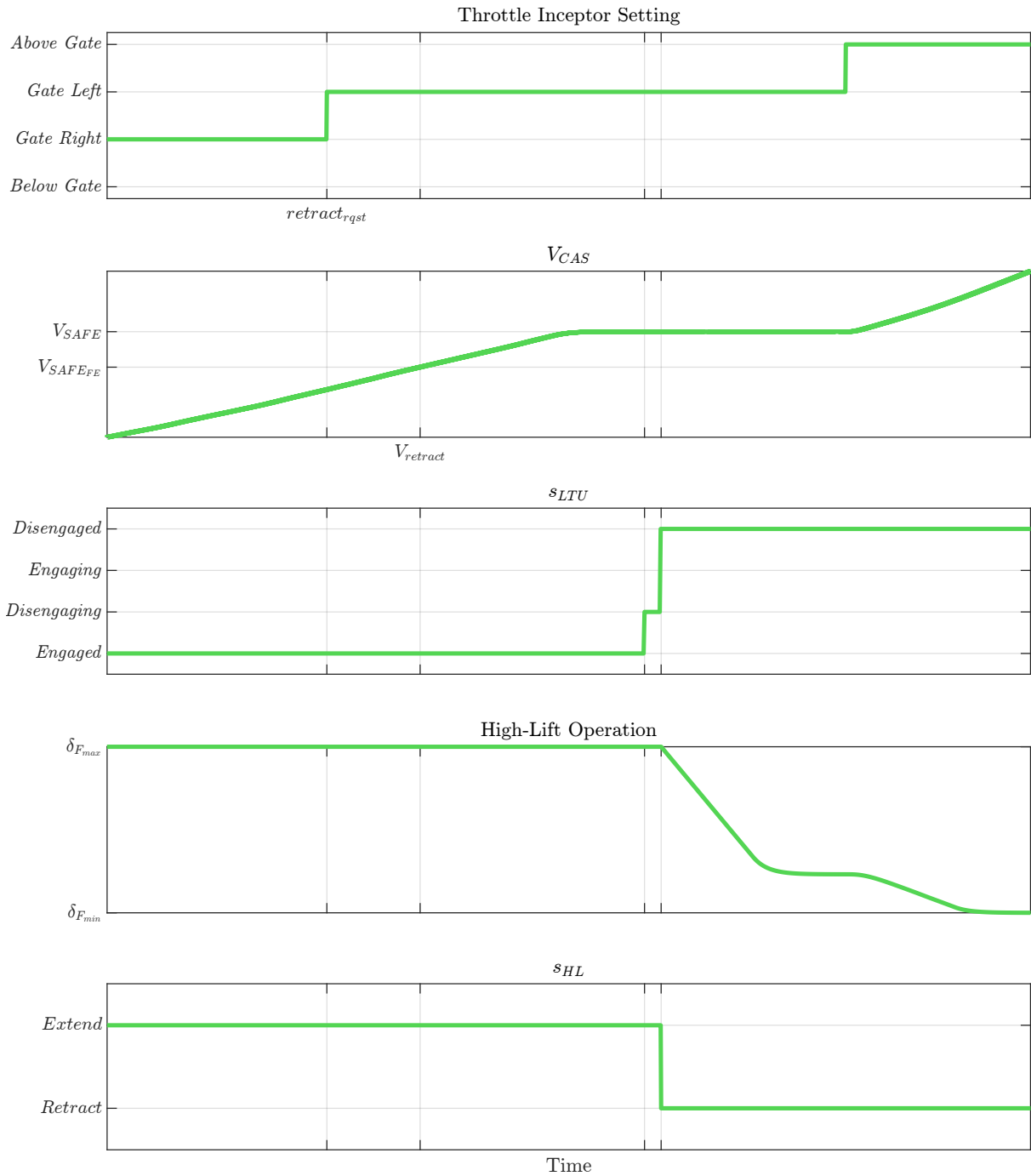


Figure 6.3: Nominal System High-Lift Management Simulation Results During Transition

As previously explained in Section 3.4.4.3 of Chapter 3, the transition to this state changes the high-lift system management to deflect the flaps such that the drag is minimized. This is in accordance to the scheduling found in Figure 5.16. As visible in the third graph of Figure 6.3, the flap deflection changes in accordance with the scheduling after the transition has been completed.

Lastly, the scheduling of the airspeed protections during transition needs to be observed. Figure 6.4 provides the simulation results for this function operation. The underspeed and overspeed protections are scheduled in accordance with Tables 3.5 and 3.6 respectively. The values they assume are visible in the lowest graph of Figure 6.4, whereby underspeed and overspeed protection are depicted with orange and red respectively.

As visible from Table 3.5, the underspeed protection is not applicable whenever the system is in powered-lift mode of operation. The reason for this is that stall is not hazardous when utilizing the LTUs for force and moment production. Therefore, while the state of M_{LTU} is *Engaged*, the lower airspeed protection does not exist. This is visible in Figure 6.4. After the disengagement process is initiated (i.e. s_{LTU} is not *Engaged*), then the lower airspeed is dependent on the status of the high-lift. Therefore, once movement outside the extended position is detected, the upper airspeed limit changes from the initial value of $V_{SAFE_{FE}}$ to V_{SAFE} .

As per Table 3.6, initially, the upper airspeed protection value is at $V_{LS_{NE}}$ in order to prevent structural damage. Once the LTUs are confirmed to be disengaged, the upper airspeed limit is relaxed and is dependent on the high-lift system deployment. This is in accordance with the lookup values depicted previously in Figure 5.16 of Chapter 5. As the retracted high-lift position is confirmed, the upper airspeed limit is increased further to V_{NE} as per Table 3.6.

6.1.2 Retransition

Section 3.5.1.2 of Chapter 3 describes the high-degree of automation behavior during retransition. This section provides simulation results that support those claims. As defined in Section 3.5.1.2, the retransition sequence is to first deploy the high-lift system and thereupon engage the LTUs. The operation of the flaps during the retransition process is visible in Figure 6.5.

According to the procedure, summarized in Table 4.9 of Chapter 4, a deceleration of the aircraft is initiated by the crew by moving the throttle inceptor into the gate. This is visible in the upper graph of Figure 6.5. In the plot, the outcome of the pilot input processing found in Figure 5.7 is presented. Namely, the pilot moves the throttle lever to the left portion of the gate (i.e. $\delta_T \in \mathbb{L}$).

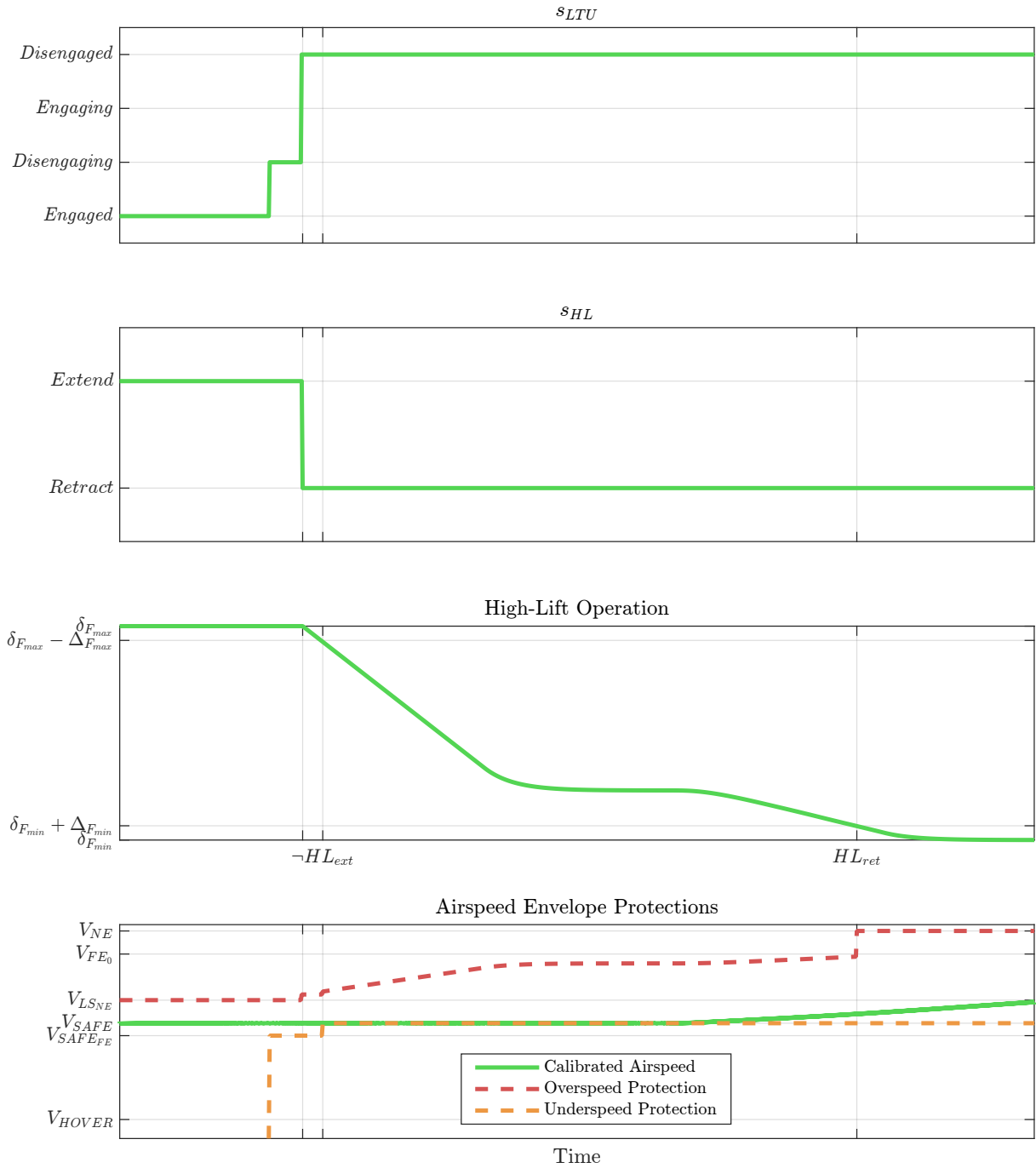


Figure 6.4: Nominal System Airspeed Protection Management Simulation Results During Transition

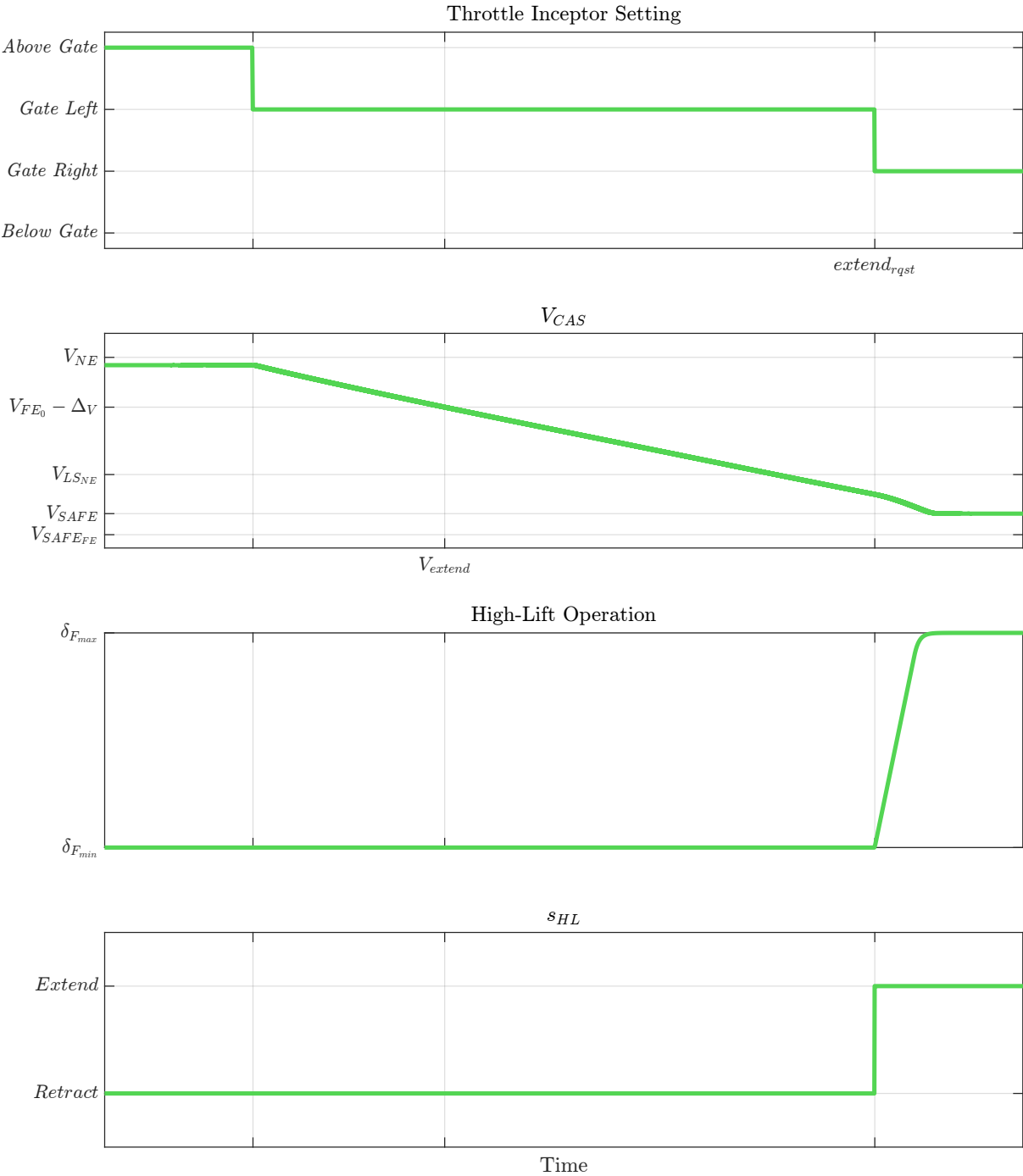


Figure 6.5: Nominal System High-Lift Management Simulation Results During Retransition

As the aircraft decelerates and the airspeed goes below the structural limit speed of V_{LSNE} , according to Table 4.9 the operator is required to deflect the control inceptor into the transition region, i.e. $\delta_T \in \mathbb{R}$. The upper graph of Figure 6.5 depicts the moment in which this occurs via the variable $extend_{rqst}$. For recollection, this is calculated as per Equation 3.41 and implemented as depicted in Figures 5.14a and 5.7.

According to 3.48, this is one of two conditions that need to be satisfied in order to initiate the flap deployment. Apart from this, the aircraft needs to fly at an airspeed lower than the extension speed. This speed is defined in 3.43 and the evaluation is found in Figure 5.14b. The moment this condition is satisfied is depicted in the second graph of Figure 6.5.

Fulfilling both conditions implicates that the transition function found in Equation 3.47 is triggered. This function is implemented as per Figure 5.15. Therefore, the State Machine M_{HL} transitions to the state *Extend* as visible in the last graph of Figure 6.5. Thereupon, the flaps are extracted via the scheduling previously explained in Section 3.4.4.3 of Chapter 3 and visualized in Figure 5.16 of Chapter 5 for the current application. The response of the high-lift system operation is visible in the third graph of Figure 6.5.

According to Section 3.5.1.2 of Chapter 3, once the flap deployment is initiated, the engagement of the LTUs is expected. Figure 6.6 demonstrates the system management of the distributed propulsion during the retransition process.

Equation 3.57 defines the transition conditions to initiate the engagement process. According to the equation in the fault-free case applicable here, the engagement process may initiate if three conditions are met. Firstly, the aircraft must be in an airspeed envelope in which no structural damage may ensue if the LTUs were to engage. This is captured by the variable $V_{retrans}$ as per Equation 3.8. In the current implementation, this is realized as depicted in Figure 5.9. The event is captured in Figure 6.6 in the third graph.

The next condition is that the retransition is requested by the operator. According to Equation 3.4, this is applicable whenever the throttle inceptor is outside the wingborne division of the gate. Therefore, deflecting the level into the right portion of the gate communicates to the automation that the flight phase is required by the operator. In the upper graph of Figure 6.6, the outcome of the pilot input processing found in Figure 5.7 is presented. In the plot, $retrans_{rqst}$ denotes the event when this is registered by the automation as per implementation depicted in Figure 5.8.

The last condition is that the high-lift deployment has concluded fully. This is realized in accordance with Equation 3.51. As previously explained during the simulation result presentation of the high-lift management, the deflection of the inceptor to the right portion of the gate initiates the deployment. The movement of the flaps is denoted in the second graph of Figure 6.6. The moment when the high-lift system is close to reaching the fully deployed setting is captured via HL_{ext} on the graph as per Equation 3.51. The generation of this variable is found in Figure 5.11.

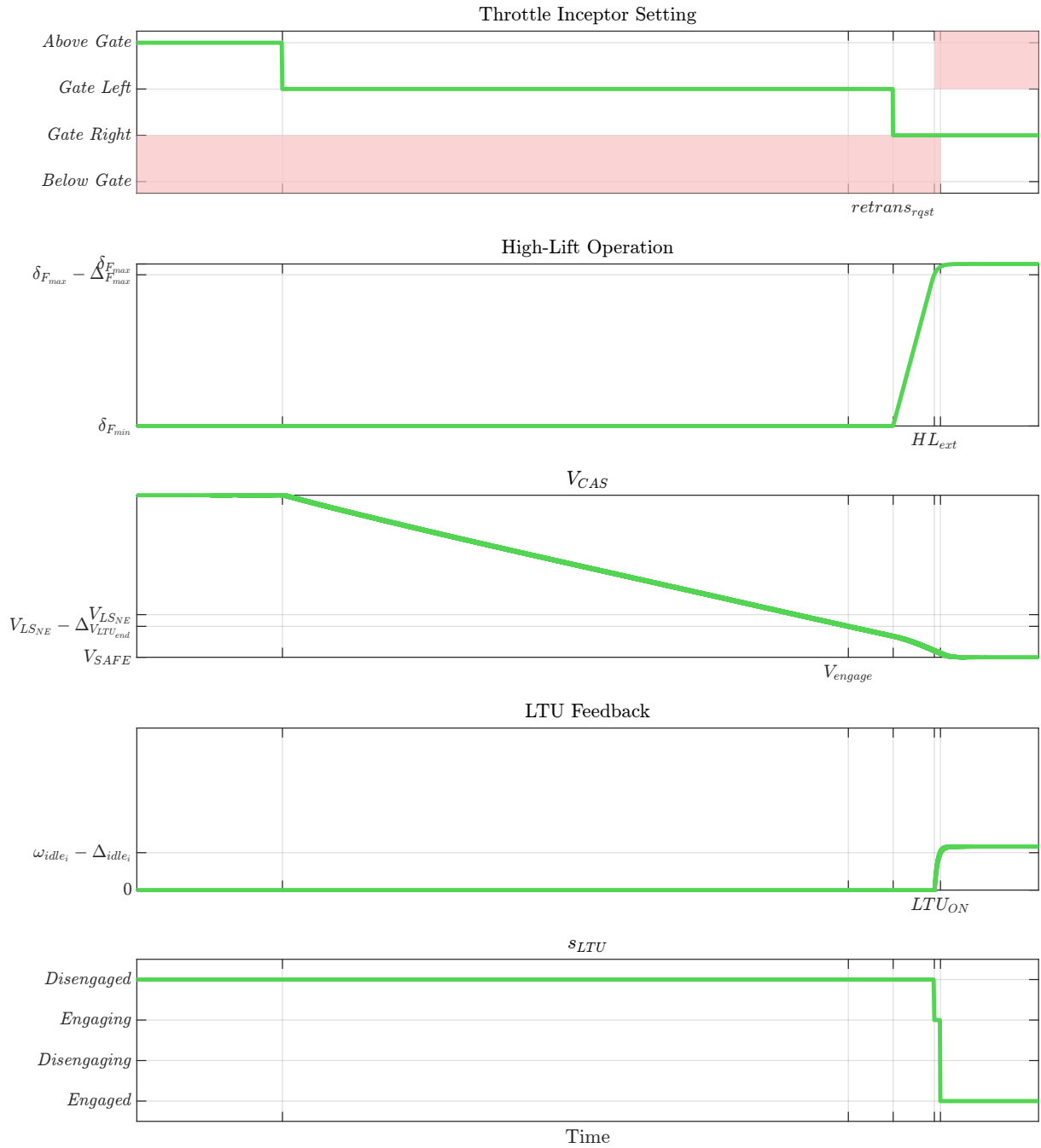


Figure 6.6: Nominal System Powered-Lift Management Simulation Results During Retransition

Once all three conditions are fulfilled, then the transition condition of Equation 3.57 is evaluated to be true. As per Equation 3.54, this forces M_{LTU} to transition to *Engaging*. This activity is implemented as found in Figure 5.12. The state change is visible in the last graph of Figure 6.6. As explained in Section 3.4.4.1 of Chapter 3, in this state the control allocation produces an LTU command of idle RPM. This allows the automation to check for the correct operation of the distributed propulsion prior to entry into powered-lift flight.

The idle RPM command to the LTUs is visible in the third graph of Figure 6.6. There the lowest RPM response of all LTUs is depicted, on the x -Axis and the threshold after which the units are evaluated to be correctly engaged is visualized. This is in accordance with Equation 3.13. In the implementation example, this check is performed as per Figure 5.10, whereby the moment after which the signal is valid can be seen on the y -Axis of the third graph of Figure 6.6. Once this value is exceeded, the condition found in Equation 3.25 is fulfilled. This triggers the transition condition of M_{LTU} to the state *Engaged* as visible in the lowest plot in Figure 6.6.

In terms of haptic feedback, the management of the barriers is as per Table 4.7 and is implemented as visible in Figure 5.6. The barrier operation during the retransition is depicted in the upper graph of Figure 6.2, whereby the red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region. It is therefore visible that the wingborne regions where high airspeeds can be commanded are inaccessible once the disengagement has been initiated. Similarly, the powered-lift regions can be entered only once the correct engagement of the LTUs has been confirmed.

Lastly, the operation of the airspeed protections during the retransition process is of interest. Figure 6.7 provides the simulation results for this function operation. The underspeed and overspeed protections are scheduled in accordance with Tables 3.5 and 3.6 respectively. The values they assume are visible in the lowest graph of Figure 6.7, whereby underspeed and overspeed protection are depicted with orange and red respectively.

As visible in Table 3.6, the scheduling of the overspeed protection depends on the status of the high-lift and powered-lift systems. Prior to the deployment of the flaps, the upper limit that can be reached by the aircraft is the structural limit speed V_{NE} . However, as the high-lift system starts extending and the deflection is measurable by the automation, then the limit is set in accordance with the velocity V_{FE} as per Table 3.6. This is done using the values depicted in Figure 5.16. The trigger for the switch can be seen in the third graph of Figure 6.7.

As soon as the engagement of the LTUs is initiated, Table 3.6 prescribes that the overspeed protection should further be limited to V_{LSNE} . This is visible in the last graph of Figure 6.7.

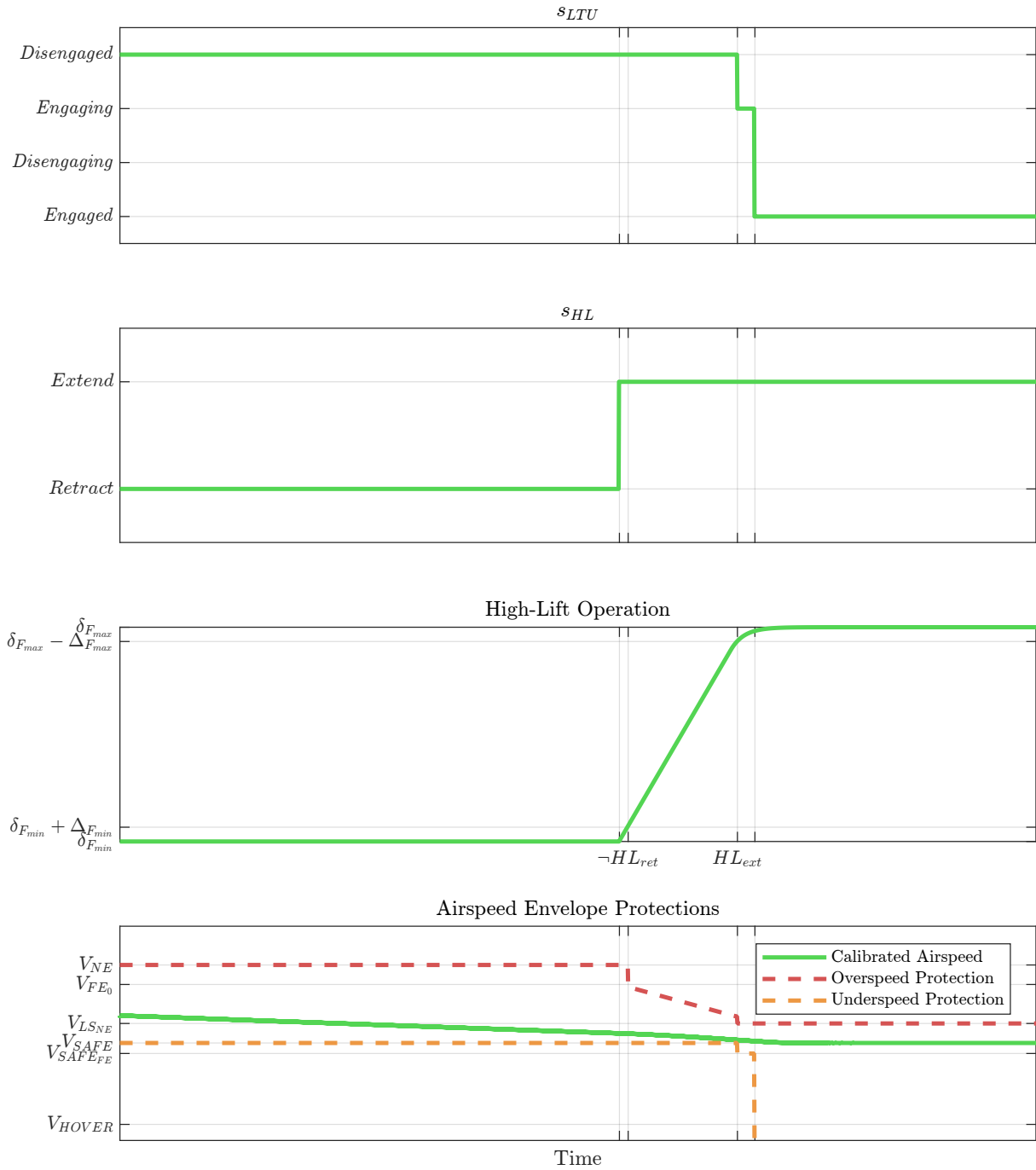


Figure 6.7: Nominal System Airspeed Protection Management Simulation Results During Retransition

According to Table 3.5, while the high-lift system is not fully extended, the underspeed protection limit is set to be V_{SAFE} . The lowest graph of Figure 6.7 confirms that this is the case. The limit can only be relaxed to $V_{SAFE_{FE}}$ once the full deployment is confirmed by the automation. This is realized in accordance with Equation 3.51. The moment when the high-lift system is close to reaching the fully deployed setting is captured via HL_{ext} on the graph as per Equation 3.51. The generation of this variable is found in Figure 5.11.

Once the engagement of the LTUs has been confirmed by the automation, the lower airspeed protection is fully lifted as per Table 3.5. This is visible in the last graph of Figure 6.7 following the state transition of M_{LTU} to *Engaged*.

6.1.3 Takeover State Evaluation During Nominal System Operation

In the previous two sections, the normal transition and retransition using the Nominal system was examined. As discussed in Section 4.2.3 of Chapter 4, the Fallback system constantly monitors the progress of the automation. Using the Nominal system automation states, the Fallback system is capable to initialize its State Machines such that a correct takeover can be performed at any point during the transition and retransition. For recollection, a takeover can be initiated manually on pilot demand or automatically in the case where an error in the Nominal system is detected. This section examines the correctness of initial state evaluation of the Fallback system.

As defined in Table 4.3 and Equation 4.18, the evaluation of the initial states following a takeover is done algebraically using the high-degree of automation state s_{LTU} . In the current example, this specification is implemented as found in Figure 5.17. Simulation results that illustrate the operation of the *Initialize* function are visible in Figure 6.8.

The left and right half of the results visible in Figure 6.8 originate from the transition and retransition process with the Nominal system respectively. Those were examined in more detail previously in Sections 6.1.1 and 6.1.2 respectively.

In the first row, the airspeed is plotted for reference during both transition and retransition process. The state s_{LTU} is available on the next row found in the figure. Here, the Nominal system is in command, therefore this information is not depicted for the sake of visibility. The next two rows depict the outcome of the implementation of the *Initialize* function found in Figure 5.17. Namely, the third plot shows the output of the first level selection found in Figure 5.17a and the last plot illustrates the outcome of the second level as per Figure 5.17b. It is visible that the specification available in Table 4.3 is fulfilled.

For the evaluation to be correct, it is visible that the information, fed to the Fallback system needs to be feasible. As previously mentioned in Section 4.2.3, this property is satisfied by a functional monitor that ensures the state transitions of the Nominal system are executed in the appropriate manner and therefore taking the last valid value suffices to ensure correctness of the Fallback state selection.

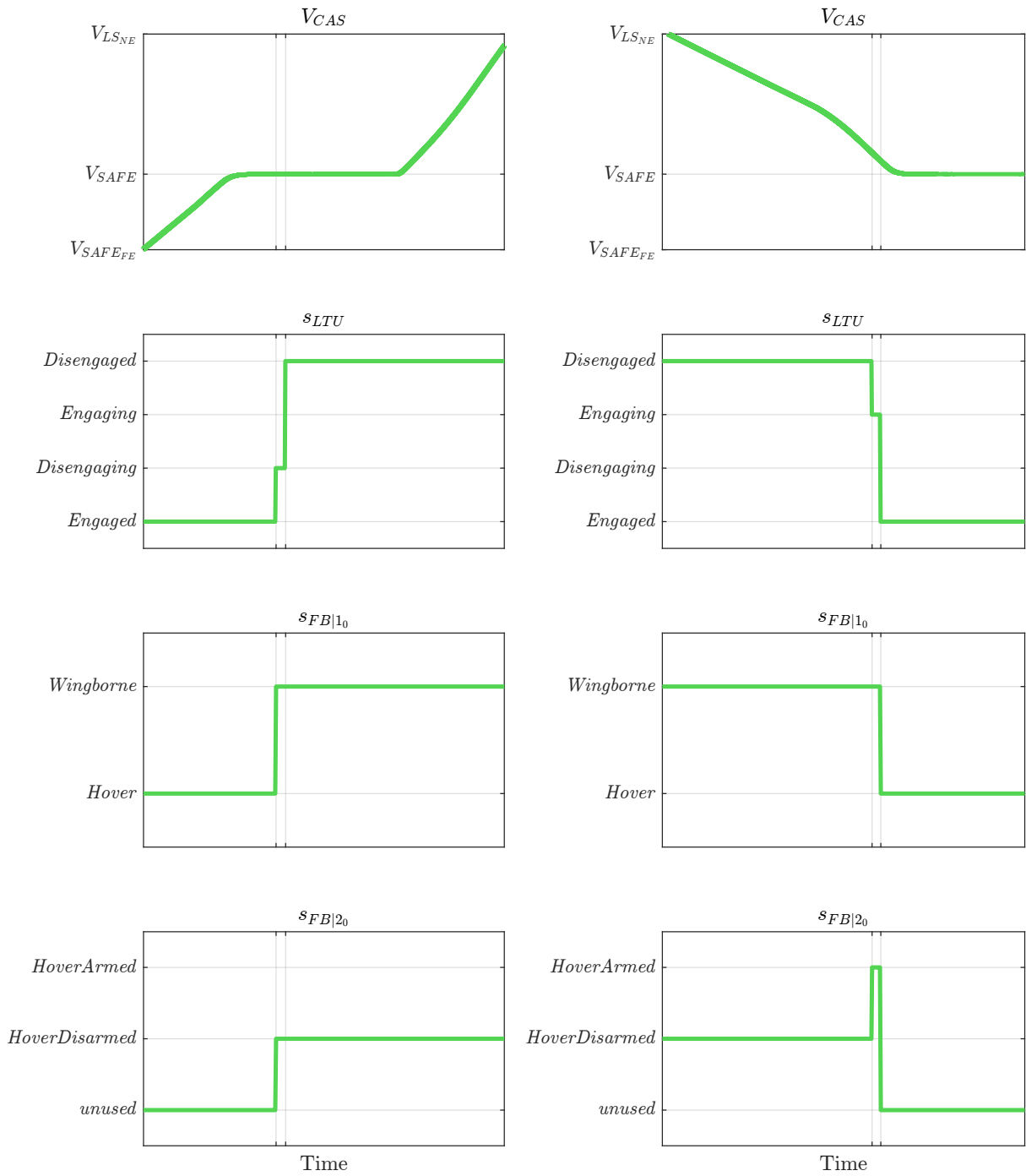


Figure 6.8: Results of the Fallback System Initial State Evaluation Following a Potential Takeover During Nominal System Flight. Left: Evaluation During Transition. Right: Evaluation During Retransition

The methods, provided in this thesis ensure the correctness of the Fallback system's state initialization following a takeover. The aircraft response that is caused by the takeover and the functional monitor that may initiate the reversion to the Fallback system are in the scope of further publications at the TUM Institute of Flight System Dynamics.

6.2 Fault-Free Fallback System Transition and Retransition

This section provides simulation results Fallback system automation mechanisms during transition and retransition. The transition process is explained first. Results of the performed simulation are available in Figure 6.9.

For recollection, the procedure from the perspective of the automated system is explained in Section 4.3.1.1. According to Table 4.8, the transition starts with an acceleration command by the operator by deflecting the throttle inceptor towards the end of the transition region (i.e. in the right portion of the gate $\delta_T \in \mathbb{R}$). This is depicted in the first graph of Figure 6.9. The acceleration of the vehicle is apparent when examining the airspeed visible in the second graph of the figure.

In accordance with the harmonized procedures of Table 4.8 found in Chapter 4, the pilot is expected to wait until the aircraft exceeds the stall speed. Upon reaching such higher dynamic pressures, the operator is required to deflect the throttle into the wingborne region, i.e in the left portion of the barrier. This is visible that in this simulation the pilot performs this action well above the stall speed, namely when exceeding the stall speed $V_{SAFE_{FE}}$.

While accelerating further, the operator is expected to monitor the LTU revolution rates. Whenever the RPM of all LTUs is deemed to be low enough, then the disengagement process can be initiated by the pilot. The third graph illustrates the LTU usage by the control law and allocation. For the sake of visibility, the maximum RPM is visualized.

Once the operator confirms that the LTU RPM commands are below the predefined threshold, then the disengagement of the powered-lift system must be initiated. According to the State Machine M_{FB} introduced in Chapter 4, this is done in accordance to the condition found in Equation 4.10. There, it is visible that the operator is capable of requesting the Fallback system's wingborne mode and therefore the disengagement of the LTUs when two properties are satisfied.

Namely, the control inceptor needs to be in the wingborne division. This is performed by the operator in accordance to the procedure found in Table 4.8 whenever the stall speed is exceeded. In addition, a confirmation needs to be communicated to the automation via the $shutdown_{rqst}$ variable, which according to Equation 4.33 needs to be performed via the input item $OPEN_{GATE}$. The check for both conditions is conducted in the implementation example as per Figure 5.18a, whereby some of the signals originate as found in Figure 5.7.

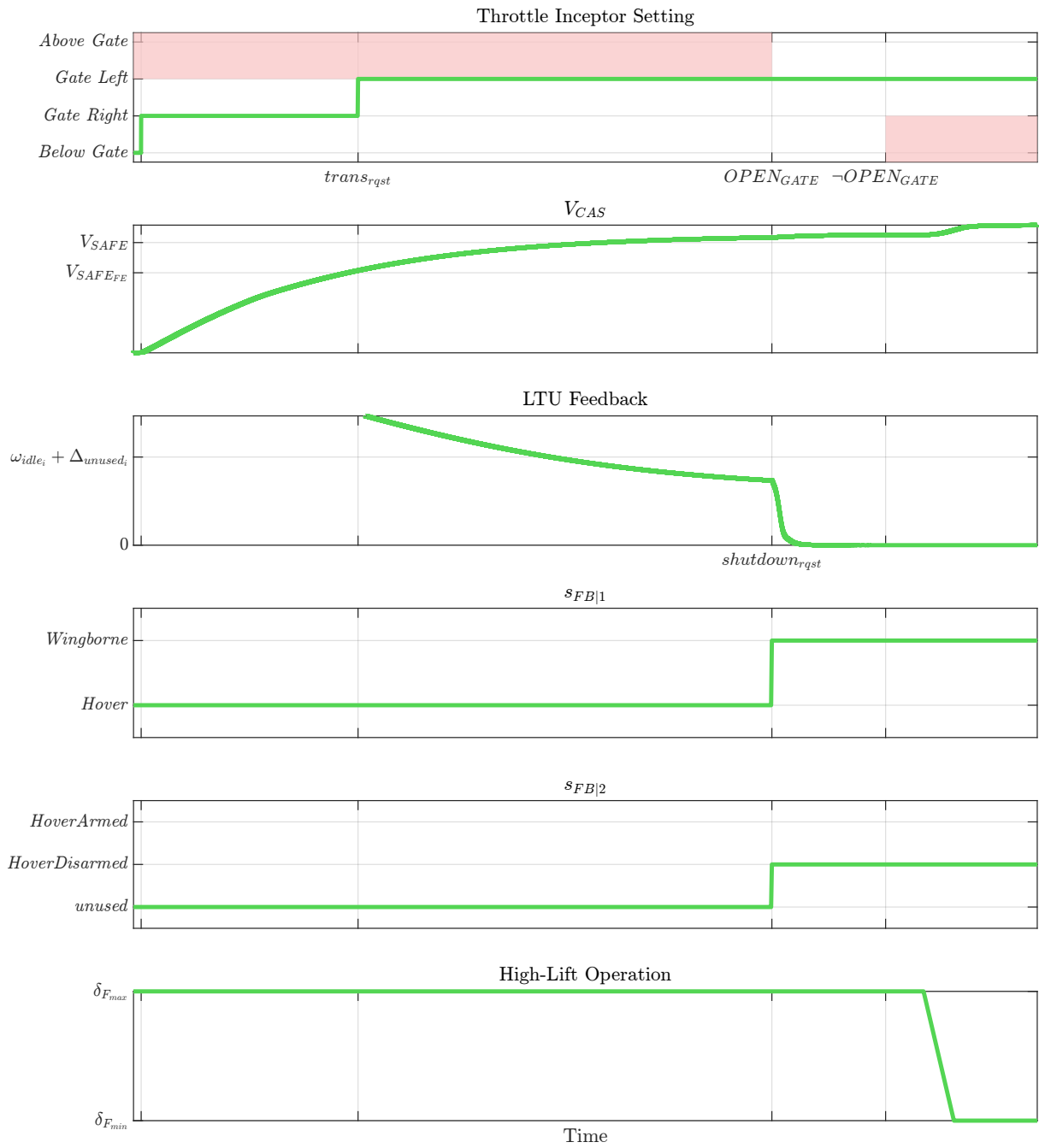


Figure 6.9: Fallback System Automation Operation During Transition

The exact moment during the transition in which the operator does both actions is visible in Figure 6.9. Namely, $trans_{rqst}$ indicates the movement of the throttle to the wingborne division and is indicated on the first graph. The moment, in which $OPEN_{GATE}$ is applied and thus $shutdown_{rqst}$ is fulfilled is on the first and third graphs respectively. Therefore, the transition function of Equation 4.9 causes the State Machine M_{FB} to transition to the states $\{Wingborne, HoverDisarmed\}$. For the example provided here, this transition is implemented as found in Figures 5.20 and 5.21. As explained previously in Section 4.2.4, this engages the wingborne mode of the Fallback law and forces the control allocation to ramp down the RPM commands to the LTUs until they come to a halt. The latter is visible in the third graph of Figure 6.9.

In accordance to the procedure, found in Table 4.8, the operator is then required to manually retract the high-lift system. This action is visible in the last graph of Figure 6.9. For recollection, with the Nominal system in this envelope the airspeed is tracked. In contrast, the Fallback system has a traction thrust command. The retraction of the flaps causes a reduction of the aircraft drag. This leads to an excess of available power which causes an acceleration of the vehicle. This is visible in the second graph of Figure 6.9.

In terms of haptic feedback, the management of the barriers is as per Table 4.7 and is implemented as visible in Figure 5.6. The barrier operation during the transition is depicted in the upper graph of Figure 6.9, whereby the red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region. It is therefore visible that initially the wingborne region where high airspeed can be commanded are inaccessible. This limitation is lifted with the $shutdown_{rqst}$ as visible in the figure. Because this input variable is generated using the operator input $OPEN_{GATE}$, both barriers are lifted as this point. The reason for this is that $OPEN_{GATE}$ lifts both barriers regardless of the automation output as previously explained in Section 2.4.3.1. The subsequent release of $OPEN_{GATE}$ as visible in the first graph of Figure 6.9 allows for the activation of the lower barrier as per Table 4.7 due to the Fallback automation's state transition to the wingborne mode. This prohibits the command of low traction thrust commands.

When performing the transition with the Nominal system, the operator sends clear commands to the automation, whereby the reconfiguration is performed fully automatically by the system. It thus requires less crew involvement. As visible in Figure 6.9, the transition with the Fallback system is a collaboration by human and machine. The reconfiguration to wingborne flight is the responsibility of the operator and the pilot commands are used to both control the aircraft states but are in addition processed by the automation. The system thereby provides the pilot with the required response. Whenever the conditions are suitable, the operator proceeds to command the entry into wingborne mode which is provided by the automation accordingly.

The retransition process with the Fallback system is examined next. The results of the performed maneuver are available in Figure 6.10. In the paragraphs below, the results are elaborated upon.

In accordance with the derivations from Table 4.9, the start of the retransition process is with the deceleration of the aircraft that is driven by the operator thrust demand. The pilot namely is expected to deflect the throttle inceptor to the lowest portion of the wingborne division (i.e. $\delta_T \in \mathbb{L}$). This action is visible in the first graph of Figure 6.10.

As soon as the speed of V_{FE} is crossed, the deployment of the high-lift system can be initiated. In this particular scenario this is the case prior to the initiation of the retransition, therefore this can be performed immediately. The extension of the flaps that is executed by the operator is visible in the third graph of Figure 6.10. It must be noted that it is the pilot's responsibility to ensure that this action is performed below the structural limit speed. Nonetheless, the protection function found in Section 4.2.4 would prohibit the deployment if the airspeed is too high.

Both reduction of the throttle and the deployment of the flaps induce a deceleration of the vehicle. The latter does so by increasing the aircraft drag. In accordance to the procedure in Table 4.9, the pilot must wait until the speed of V_{LSNE} is crossed. Whenever this is the case, the activation of the LTU can be commenced.

As discussed in Chapter 4, the control allocation of the Fallback system initiates the engagement of the LTUs when the automation's State Machine M_{FB} has its second level state $s_{FB|2}$ in *HoverArmed*. According to the Equation 4.12, this is the case when the following conditions apply. First, the throttle needs to be in the right division of the gate ($\delta_T \in \mathbb{R}$). This is evident in the first graph of Figure 6.10. Secondly, in accordance with Equation 4.7, the variable $LTUengage_{rqst}$ needs to be true. As visible in Equation 4.34, this would apply whenever $OPEN_{GATE}$ is confirmed by the operator. In the current example, the processing is implemented as found in Figure 5.18b.

The time point, in which both conditions are fulfilled is indicated on the fourth graph of Figure 6.10 with arm_{rqst} . In this moment, the transition function found in Equation 4.11 is triggered, changing the state of $s_{FB|2}$ to *HoverArmed*. This is implemented by the chart found in Figure 5.21. The state change is visible in the fifth graph of Figure 6.10.

In the state tuple $\{Wingborne, HoverArmed\}$, the control allocation commands idle RPM to all units within the powered-lift system. The time history of the LTU response is visible in the lowest graph of Figure 6.10. There, the minimum RPM of all LTUs is depicted for the sake of visibility. The predefined ramp to idle following the state change is evident.

According to Table 4.9, the pilot's next responsibility is to verify the correctness of the powered-lift system's engagement. Once this is confirmed, then the Fallback system's hover mode can be activated. This is done in accordance with the transition condition, found in Equation 4.14. This is done via the variable $hover_{rqst}$ and is computed in accordance with Equation 4.35. For recollection, in order to communicate this request to the automation,

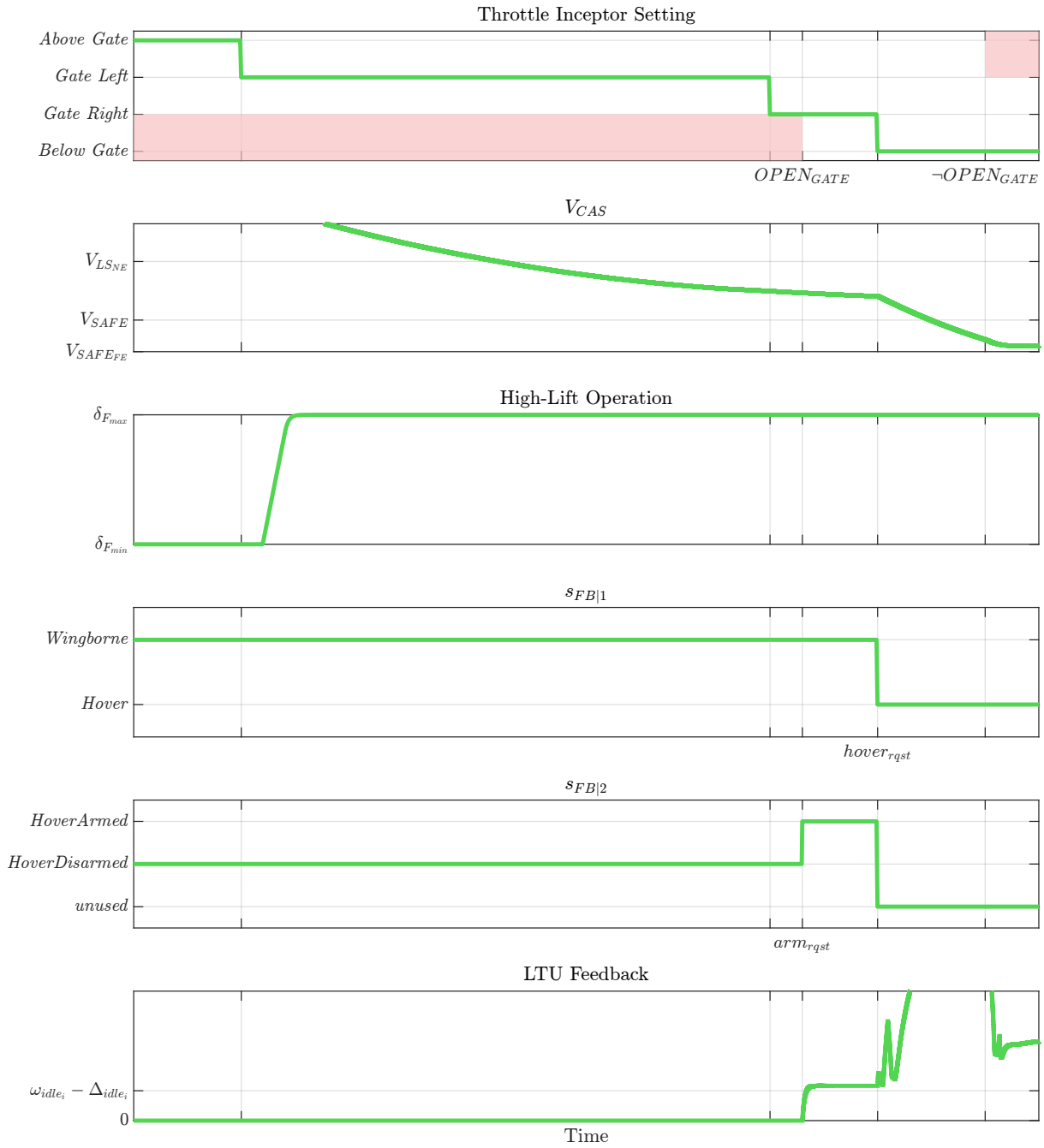


Figure 6.10: Fallback System Automation Operation During Retransition

the activities of the pilot are as follows. Firstly, the variable $OPEN_{GATE}$ needs to be *true*. This action was previously performed by the pilot who still continues engaging that item. Next, the throttle inceptor needs to be moved outside the gate in direction of the hover phase. Though it can remain in the transition region, it must nonetheless be below the gate.

The moment the latter occurs is visible in the first graph of Figure 6.10. The state transition that is executed as per Equation 4.13 is implemented as seen in Figure 5.20. In the example, the input variables are generated as depicted in Figure 5.18a. The state transition to $\{Hover, unused\}$ permits the control allocation of the Fallback system to utilize the LTUs and the reconfiguration is completed. This transition is visible in the fourth and fifth graphs of Figure 6.10.

In terms of haptic feedback, the management of the barriers is as per Table 4.7 and is implemented as visible in Figure 5.6. The barrier operation during the retransition is depicted in the upper graph of Figure 6.10, whereby the red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region. Initially the hover region where low airspeed can be commanded is therefore inaccessible. This limitation is lifted with the arm_{rqst} as visible in the figure. Because this input variable is generated using the operator input $OPEN_{GATE}$, both barriers are lifted as this point. The reason for this is that $OPEN_{GATE}$ lifts both barriers regardless of the automation output as previously explained in Section 2.4.3.1. The subsequent release of $OPEN_{GATE}$ after engaging the Fallback law's hover mode as visible in the first graph of Figure 6.9 allows for the activation of the upper barrier. This is as per Table 4.7 due to the Fallback automation's state transition to the tuple $\{Hover, unused\}$. The subsequent barrier constellation prohibits the command of low traction thrust commands.

6.3 Abnormal Scenarios

The transition and retransition processes of both Nominal and Fallback systems consider component malfunctions that could lead to mitigation strategies or abnormal procedures as discussed in Chapters 3 and 4 respectively. In addition, the procedures were compared in Section 4.4.3. This section summarizes the validation effort with relation to the off-nominal scenarios and the response of the two systems.

Namely, Section 6.3.1 present the simulation results for an LTU malfunction during retransition and the subsequent termination of the procedure in order to revert back to full wingborne flight. Similarly, Section 6.3.2 does the same for the cases where powered-lift flight has to be entered regardless of the LTU malfunction and subsequent loss of system performance.

For recollection, mitigation and off-nominal procedures were developed for both Nominal and Fallback systems during transition if an LTU is incapable of disengaging. The proposed methods for both systems account for such events as well. However, due to the decreased likelihood of the occurrence of these abnormalities, the simulation results are summarized in Appendix H.

6.3.1 Retransition Mitigation - Reversion to Wingborne Flight

The retransition procedure developed in this thesis has provisions in the event where an LTU is incapable of engaging whenever attempting to enter powered-lift flight. They are discussed at length for both Nominal and Fallback system in the corresponding chapters. To summarize, one option at the operator's disposal is to revert back to wingborne flight in order to perform a go-around and reattempt the retransition. This section examines the mechanics of the automation that facilitate this reversion.

The procedure itself is defined as in Table 4.13 of Chapter 4. The results of this scenario while performing a flight with the Nominal system are visible in Figure 6.11. They are elaborated upon below.

Firstly, as previously stated, this procedure occurs during retransition. Therefore, the initial sequence is the same as found in Section 6.1.2 of this chapter. For this reason, the initial aircraft response in Figure 6.11 is equivalent to the one in Figure 6.6.

The issue can be identified whenever the engagement of the LTUs is required. As the State Machine M_{LTU} transitions to the state *Engaging*, the control allocation begins its command ramp-up to idle RPM. The state transition is visible in the lowest graph of Figure 6.11 and the LTU response is depicted in the fourth graph of the same image. In the latter graph, the time history in red depicts the response of the faulty LTU that does not engage. This LTU is chosen arbitrarily. Similarly to before, the green time history shows the minimum RPM of all non-faulty propulsion units for the sake of visibility.

The subsequent procedure is analyzed in Table 4.13 found in Chapter 4. Due to the inability to engage all units of the distributed propulsion, the automation requires operator input to proceed with the process. According to Table 4.13, the abnormal event - the LTU not engaging - is prompted to the operator via the variable $retrans_{timeout}$. The prompt via the indication items is in accordance with the Truth Table 3.7. The time, at which it is evaluated to be *true* is visible in the first graph of Figure 6.11 and is calculated in accordance with Equation 3.17. This equation is implemented as depicted in Figure 5.10.

According to Table 4.13, the reversion to wingborne flight is by deflecting the throttle control inceptor into the wingborne region. For recollection, this movement also raises the $trans_{rqst}$ to *true* in accordance with Equation 3.3. This is captured by the operator input processing found in Figure 5.8 and is visible in the top graph of Figure 6.11.

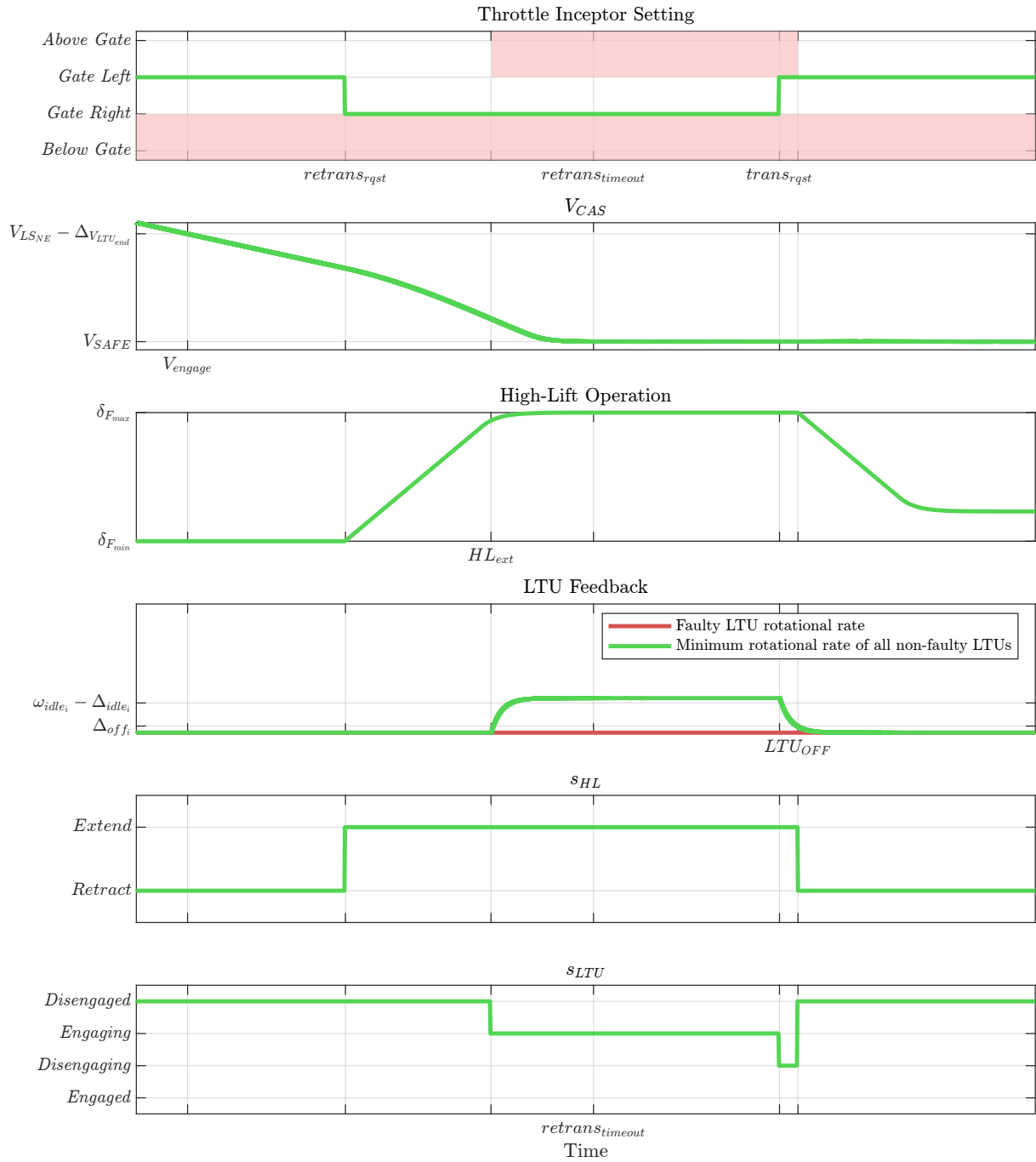


Figure 6.11: Nominal System Retransition Mitigation to Wingborne Flight

As seen in Equation 3.29, $trans_{rqst}$ also forces a state transition in M_{LTU} to the state *Disengaging*. The transition is visible in the lowest graph of Figure 6.11. Therefore, the transition mechanics are initiated. Upon LTU shutdown, the high-lift system is retracted as expected. The state transition is implemented as depicted in Figure 5.12.

In terms of haptic feedback, the management of the barriers is as per Table 4.7 and is implemented as visible in Figure 5.6. The barrier operation during the retransition is depicted in the upper graph of Figure 6.10, whereby the red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region. It is demonstrated that a safe airspeed demand is always enforced by the system in this abnormal scenario.

While the operator is not able to enter impermissible velocity commands via the haptic feedback, the Nominal system enforces a safe airspeed envelope. This is performed in accordance with Tables 3.6 and 3.5 for overspeed and underspeed protections respectively. The management of the airspeed limits is visible in Figure 6.12.

The reversion to wingborne flight following an LTU malfunction is examined next. The procedure is performed as per Table 4.13 and in terms of operator actions is the same as for the Nominal system. The simulation results are visible in Figure 6.13.

Until the point where arm_{rqst} , the retransition is identical to the one previously examined in Section 6.2. Due to the failure in the LTU, the response of the distributed propulsion is the same as for the Nominal system and is depicted in Figure 6.13.

The decision to revert to wingborne flight as per Table 4.13 is to deflect the throttle inceptor into the wingborne division. In accordance with Equation 4.8, this causes the variable $disarm_{rqst}$ to become *true* which is visible in the fourth graph of Figure 6.13. This processing is realized as depicted in Figure 5.18.

The action of deflecting the throttle inceptor thereby causes the state transition function in Equation 4.15 to be executed. The implementation of the function can be found in Figure 5.21. As visible in the fifth graph of Figure 6.13, this causes the transition of the state *HoverDisarmed*, whereby the LTUs are subsequently disengaged.

The subsequent actions of the operator are consistent with those of Table 4.9 and namely to retract the high-lift system in order to reduce the aircraft drag.

In the first graph of Figure 6.13 the haptic feedback is also depicted, whereby the convention is the same as the one previously introduced. The operation of the barriers is in accordance with Table 4.7 and realized as found in Figure 5.6. The moment where the $OPEN_{GATE}$ engaged is with arm_{rqst} as previously explained in Section 6.2. Following the reversion, it is released by the pilot and the moment where this is done is indicated in the graph and from then on the regions of the throttle inceptor where low traction thrust can be commanded are prohibited. It must be noted that in accordance with Equation 4.8, the release of the input item $OPEN_{GATE}$ is also a condition to revert to wingborne flight. Therefore, the pilot is capable of aborting the retransition without a movement of the throttle inceptor into the wingborne region but with that input item instead.

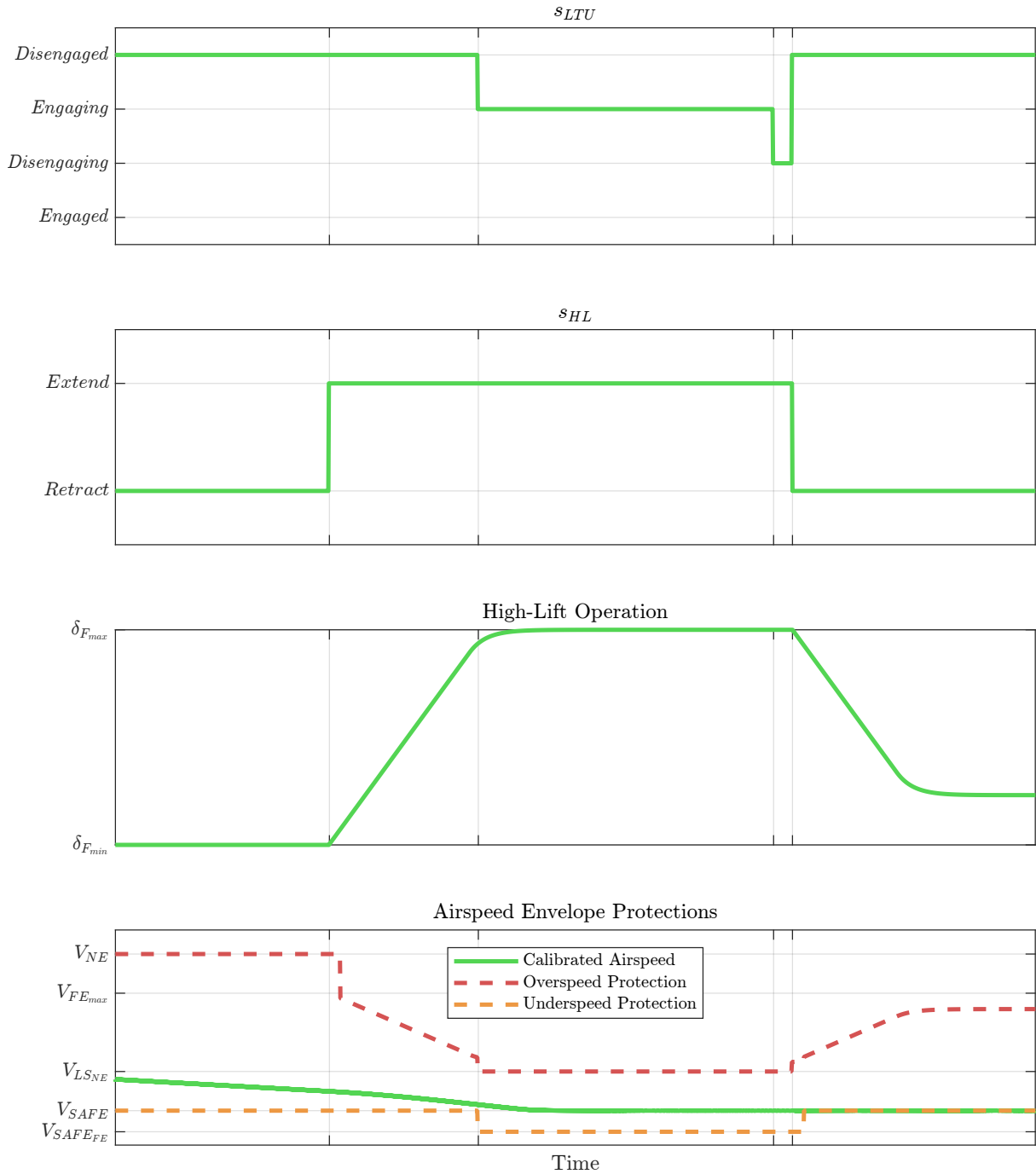


Figure 6.12: Nominal System Airspeed Protection Operation During Retransition Mitigation to Wingborne Flight

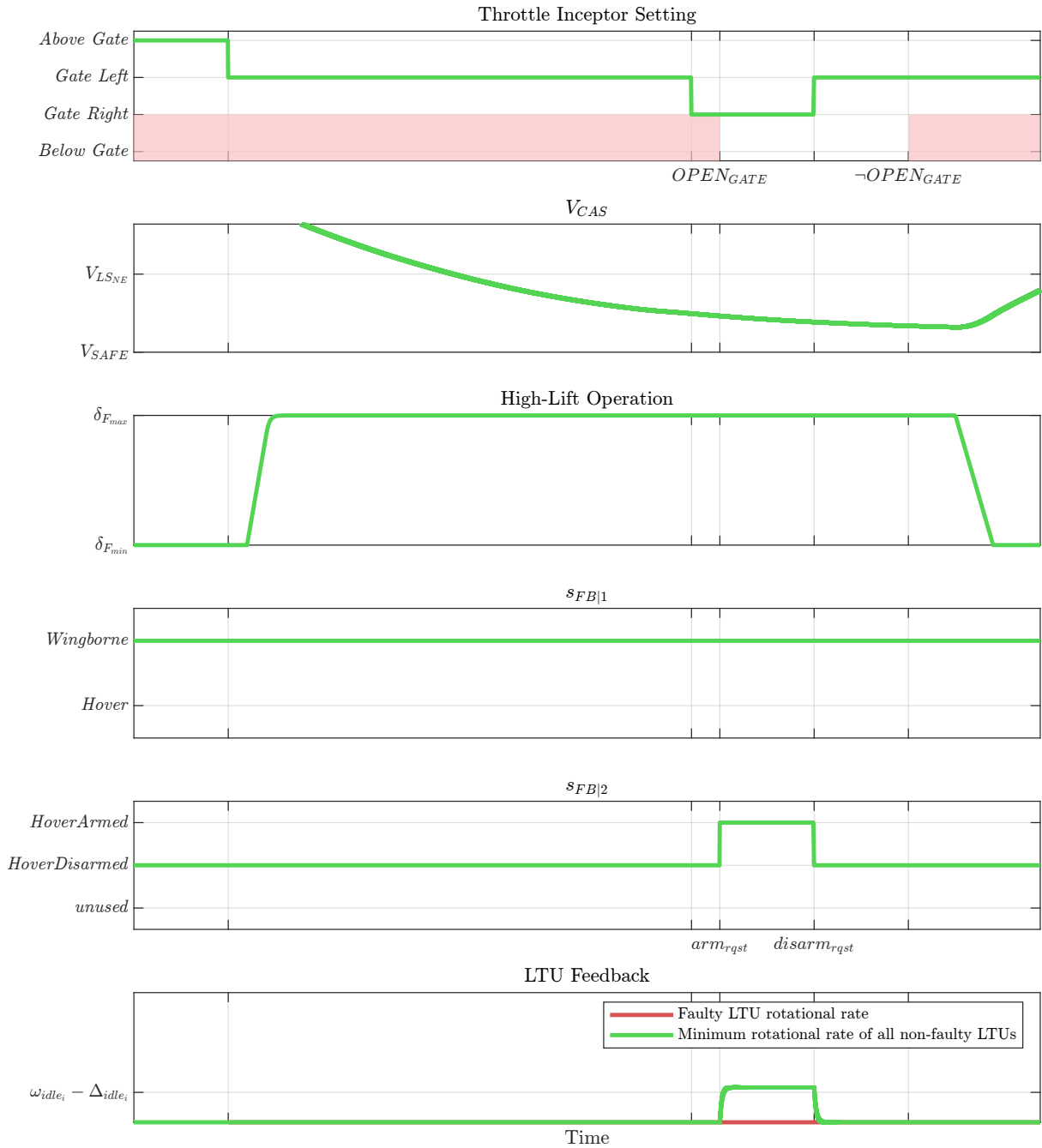


Figure 6.13: Fallback System Retransition Mitigation to Wingborne Flight

6.3.2 Retransition - Confirming the Entry to Powered-Lift Flight

The previous section examined the simulation results of the automation mechanisms that facilitate a reversion to fixed-wing flight following an LTU malfunction during retransition. The other available operator action is to accomplish the retransition and enter powered-lift flight with a degraded vertical propulsion system. The results of the validation effort for the Nominal system for this scenario are examined in this section. As discussed at length in Section 4.4.3.4, the procedure and automation mechanisms for the Fallback system in such events are exactly the same. Therefore, results for this system are not provided.

The operation of the Nominal system automation is depicted in Figure 6.14. Until the warning is forwarded to the operator via *retrans_timeout*, the time history is exactly the same as from the previous section. The remainder of the available data is different due to the difference in operator decision.

The confirmation of powered-lift flight entry is done in accordance with Table 4.14 found in Chapter 4. As visible in the table, in order to perform this procedure, the operator needs to first engage the input item *OPEN_GATE*. The moment this occurs is depicted in the upmost graph of Figure 6.14.

According to Table 4.7, the barriers are lifted due to this input. This is visible on the top graph of Figure 6.14 as well. The responsible mechanism that implements this function is as shown in Figure 5.6.

The lifting of the barrier allows the movement of the throttle control inceptor into the regions below the gate. As visible in Equation 4.35, if this is performed, then the variable *LTU_override_reqst* is evaluated to be *true* by the automation. In this particular example, this is performed as depicted in Figures 5.7 and 5.8.

As per Equation 3.31, this action and variable trigger the transition of the State Machine *M_{LTU}* to the state *Engaged*, thereby enforcing the powered-lift mode of operation. This transition is performed by the example as implemented in Figure 5.12. The state transition is visible in the lowest graph of Figure 6.14. This state enables the control allocation to use the non-faulty LTUs for force and moment production. The imminent loss of performance is confirmed by the operator via the dedicated procedure and actions. Therefore, awareness of the reduced handling qualities and the degraded system in this flight phase is gained prior to the entry into the envelope.

Figure 6.15 examines the operation of the airspeed limits during this abnormal scenario. The scheduling is performed in accordance to Tables 3.6 and 3.5 for overspeed and underspeed protections respectively. The management of the airspeed limits is visible in Figure 6.12. It can be confirmed that safe flight is guaranteed throughout the whole procedure.

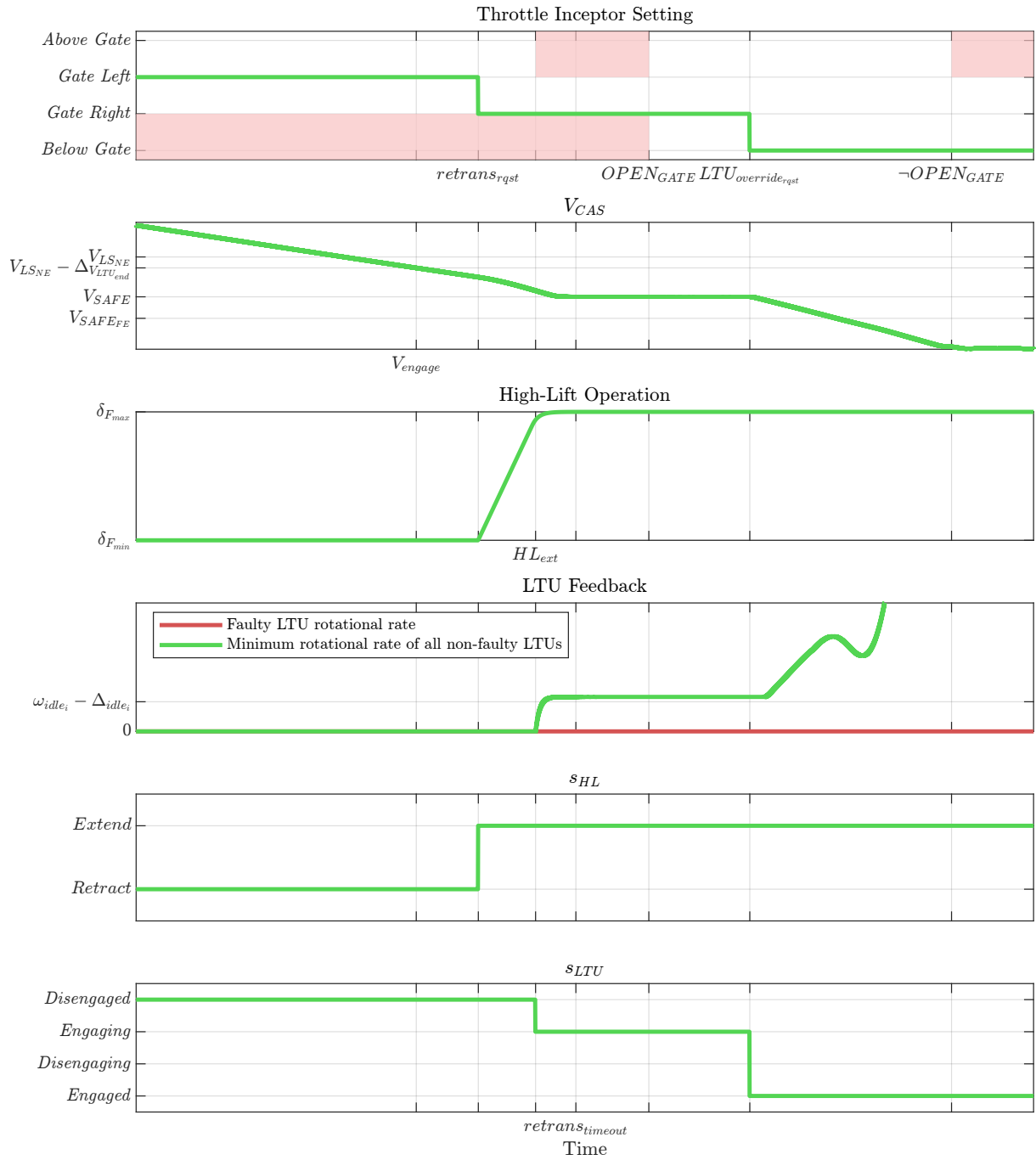


Figure 6.14: Nominal System Abnormal Entry into Powered-Lift Flight

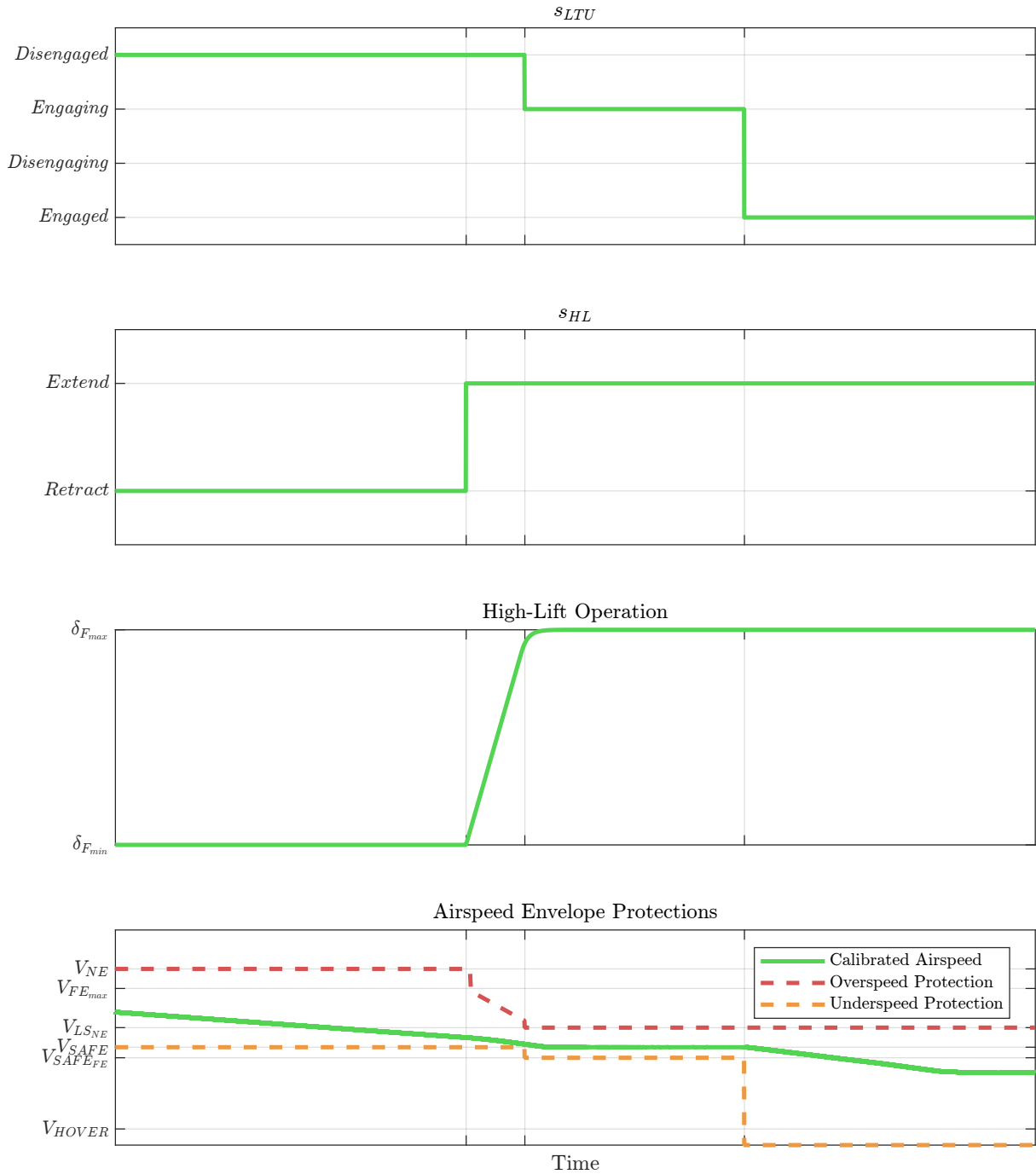


Figure 6.15: Nominal System Airspeed Protection Management During Abnormal Entry into Powered-Lift Flight

6.4 Conclusion

The previous chapters of this thesis introduced the contributions of the research. The methods and techniques were presented and the mechanisms were explained. This chapter demonstrated the plausibility of the solutions using simulation as the means of validation. It thereby also aids in the understanding of the runtime operation of the derived automation methods.

The basis for the proposed simulation is the product of Chapter 5. Thus, it demonstrated the capability of the methods of that chapter of creating high-fidelity behavioral specifications. It thereby illustrated the efficacy of **Contribution 3** as a validation method of automation functions.

The behavioral specification model implements the methods of Chapters 3 and 4 and thus provides the specification of **Contribution 1** and **Contribution 2**. The validity of the high-degree of automation strategy that was introduced and explained in Chapter 3 was analyzed in Sections 6.1 and 6.3 of this chapter for the fault-free and abnormal cases respectively. In an analogous manner, 6.2 and 6.3 did so for the low-degree of automation concept used as a fallback.

Chapter 7

Conclusion

In an industry funded project, the TUM Institute of Flight System Dynamics is responsible for the whole functional development of the flight control system for a manned lift-to-cruise aircraft. The specifics of the vehicle are subject to non-disclosure but a representative example of the airframe was introduced in Chapter 1. This thesis was inspired by the author's efforts within this project.

The aircraft's flight control system is realized with the fallback strategy. It therefore relies on two functionally dissimilar control concepts and their corresponding automation, which in this thesis are referred to as Nominal and Fallback systems. The former includes a higher level of automation, whereby high number of tasks are performed by the software. The latter provides a safety net should a malfunction of the Nominal system occur. It enables the pilot to seize control authority when required.

This author is responsible for the functions, directly associated with the vehicle automation that are allocated to the flight control system. The thesis presented selected automation concepts for both Nominal and Fallback systems that enable the transition and retransition capability of the aircraft. Though inspired by the presented airframe, the proposed solutions are built generic and could be applied to other vehicles of this type.

The solutions, developed for the Nominal system expand the state of technology as summarized with **Contribution 1**. The procedures and automation concepts for this system were presented in Chapter 3. The solutions conform to and fit in the Simplified Vehicle Operations Concept of the TUM Institute of Flight System Dynamics (FSD-SVO) seamlessly. In the nominal case, the procedures of reconfiguration from powered-lift to wingborne flight and back require no manual intervention. In addition, the automation fulfills the requirements set out from the FSD-SVO in terms of law scheduling. During reconfiguration, it provides the algorithms with the necessary information for correct execution. This is achieved using the presented human-centered high-degree of automation module that takes the operator requests and the aircraft states into account and manages the sequences and switches between the required behavioral modes.

The high-degree of automation proposal extends the FSD-SVO with regards to robustness in both nominal and off-nominal scenarios. It includes the management of the system's airspeed protection scheduling which is integrated into the automation design. The derived procedure and the automation module that implement the procedures ensure that during reconfiguration with and without faults no potentially hazardous situation can be entered. A safe state is also enforced by airspeed scheduling that is managed by the automation.

The Nominal system's automation for transition and retransition further extend the FSD-SVO to account for the utilization of a high-lift system by utilizing an additional State Machine. If the aircraft is equipped with flaps, the core State Machine responsible for reconfiguration from powered-lift to wingborne flight and back is designed modular and needs only to be supplemented by the additional automation methods found in Chapter 3.

The methods described in this thesis that are allocated to the Fallback system together with the flight control laws enable a manual transition and retransition capability, executed by the operator. This thesis demonstrates how a reconfiguration from powered-lift to wingborne flight and back can be performed and in addition derivation of the transition and retransition procedures was provided. The design ensures a high degree of operator authority and expands the state of technology as described in **Contribution 2**. The solutions for the Fallback system were presented in Chapter 4.

The design demonstrated resilience against component malfunctions and procedure deviations. In addition, the correctness of operation following a fault of the Nominal system was shown. The capability of the low-degree of automation system (Fallback) to operate in the whole flight envelope, its reduced sensor dependency and proper operation following a takeover from the Nominal system allow for correct execution of the fallback principle. This guarantees a fail-safe FCS operation in the event of an erroneous Nominal system and a multitude of sensor errors that render the Nominal system unavailable.

The above-mentioned properties are enabled by the Fallback automation design which implements transition and retransition mechanisms which are coherent to the ones of the Nominal system. Nevertheless, the Fallback automation design is highly dissimilar to the Nominal system automation and requires a significantly reduced sensor information which ensures an increased availability. In addition, the Fallback system does not impose additional requirements or restrictions on the Nominal system design and operation. Therefore, in an FCS where both Nominal and Fallback systems can be executed, all favorable Nominal system characteristics in terms of systems safety and simplified operation are retained.

The automation concepts derived in this thesis follow the notions and design considerations of human-centered automation. Thus, the operator is considered in every aspect of the procedure design. The difference between high- and low-degree of automation operational modes is the shift of the tasks from automation to pilot. In both Nominal and Fallback system operation the behavior with relation to the operator input is kept

consistent. In the events where mitigation strategies are required, the harmonization of the transition and retransition procedures ensures a fast and equivalent operator response regardless of the chosen contingency.

The tasks that are allocated to the operator during Nominal system operation are simplistic and clear. During Fallback system operation the pilot's role shifts from supervisor to aircraft operator and full command authority is ensured. In addition, the automation design accounts for deviations in the procedures and thus ensures no entry of a potentially hazardous envelope for singular deviations or faults.

The state of automation can be tracked by the crew. This statement is supported by the design which provides the indication items with all necessary information. The automation is tailored to supply the crew with the aircraft flight state and that flight state can clearly be linked to the automation's sequential logics as explained throughout the thesis. In abnormal scenarios the automation produces a very limited set of possible actions from the crew. During the pilot's decision-making process, a safe system state is enforced by the design, hence operator actions are not time critical. In addition, procedure and flight-state awareness is facilitated by the utilization of the haptic feedback. This is found in Section 4.4.1 of Chapter 4.

Lastly, the procedures and therefore the underlying automation applicability within the envisioned mission profile from the currently available regulatory effort was demonstrated. It showed that the transition and retransition with both Nominal and Fallback systems can be executed within the take-off and approach maneuvers of the MOC SC-VTOL and fully comply with the imposed requirements.

During the functional development, a method was introduced, with which the automation functions could be modeled and executed in an abstracted closed-loop environment. It was instrumental for the validation of the proposed solutions and the formulation of requirements.

Eventually, the method was expanded into the behavioral specification model, in which the control laws were embedded and the complete closed-loop flight control system response could be simulated. The behavioral specification model was presented in Chapter 5. It allows for efficient validation of the aircraft ConOps and for fast adaptations in the events where changes to the specification are necessary. It thereby advances the state of technology in accordance with **Contribution 3**.

The behavioral specification model was utilized for multiple design and validation efforts at the TUM Institute of Flight System Dynamics. Among others, the method proved useful in identifying missing functionality and pinpointing improvement potential in both control and operational concepts. The proposed solution was used for the analysis of the transient response during takeover from Nominal to Fallback systems, design of cockpit indications, pilot-in-the-loop testing, functional monitor design and many more.

The behavioral specification model is largely system-architecture agnostic and was therefore applied in the early stages of the product development cycle. The modeling method recreated the automation functions in a highly simplified but representative environment and enabled the simulation of the aircraft operation prior to the full functional development.

The behavioral specification model implements the methods of Chapters 3 and 4 and thus provides the specification of **Contribution 1** and **Contribution 2**. Analytical proof was used as a means of justification of the solution correctness in the corresponding chapters. However, Chapter 6 demonstrated the plausibility of the methods using simulation as the means of validation. The basis for the proposed simulation is the product of Chapter 5. Thus, it in addition demonstrated the capability of the methods of that chapter of creating high-fidelity behavioral specifications. It thereby illustrated the efficacy of **Contribution 3** as a validation method of automation functions.

This chapter summarizes the limitations of the solutions, presented in this thesis, and outlines the potential future work that can be accomplished using the developed methods. It is organized in three parts. First, Section 7.1 summarizes all already performed activities that were discussed in this thesis but were nonetheless performed in the context of the industry funded project. The section contains finished work that is subject to documentation in the form of future publications or technical notes. Secondly, Section 7.2 lists all effort that builds upon the solutions of this thesis and is currently ongoing. The chapter is concluded with Section 7.3. The perspectives for future improvements of the methods and ideas on how the automation solutions can be expanded are listed. In addition to describe the problems and possible solutions to each topic, emphasis is placed whenever modifications or supplementation to the thesis solutions are necessary.

7.1 Completed Topics

This section lists the tasks that were already performed within the vehicle automation development efforts. Although not in the scope of this work, they are completely consistent with the methods presented in the thesis.

Nominal System Ground Mode

Recall from Section 2.4.1.1 of Chapter 2 that the Nominal system's control law requires knowledge of the mode of operation, i.e. *HV*, *TR* and *WB*. The currently envisioned control law algorithm for this system is Incremental Nonlinear Dynamic Inversion (INDI). Available publications on the method include [111, 138].

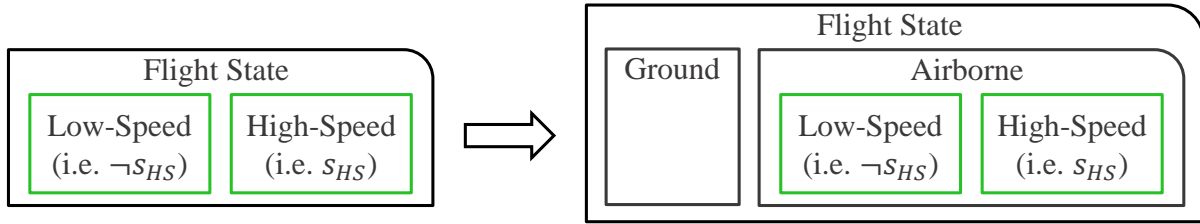


Figure 7.1: *Nominal System Ground/Airborne Mode Provision*

The INDI approach relies on command increments that are added on top of the currently assumed effector states. Those increments originate from the control error and a feedforward path. Due to the former, the algorithm resembles a global integrator. INDI offers high robustness against model uncertainties and increased disturbance rejection in the presence of unknown external factors.

Although the high response against external disturbances is desirable while airborne, this characteristic is a shortcoming when the aircraft is on the ground. The reason for this is that the forces and moments that act on the aircraft on the ground are interpreted as an external disturbance by the algorithms. As a consequence, the law may exhibit a windup that is associated with all integrator elements. Integrator windup is explained in [139]. Due to this effect, a transient may occur once a take-off is performed after a prolonged aircraft operation with an engaged law on the ground.

A method to counteract this issue is to introduce a law mode of operation in which the INDI command increments are adapted such that a windup is mitigated. For the purposes of this discussion, this mode is referred to as the “ground mode” of the Nominal system. The method of the law that mitigates this issue is not in the scope of this discussion. The emphasis here is on the definition of this mode and its logics as they are allocated to the automation functions.

The currently available solution utilizes the s_{HS} state of the Nominal system. Sections 3.4.1 and 3.4.4.1 of Chapter 3 demonstrated how the scheduling of the law is performed but this state is responsible in separating between hover and transition mode of the law. More precisely, a second level is introduced to that automation module. The first State Machine level dictates whether the aircraft is airborne or on the ground. The previously introduced decision-making represented with s_{HS} becomes a substate of the airborne state. The change is illustrated in Figure 7.1. The above-mentioned mitigation is applied or deactivated based on these states. The decision-making process of the automation relies on both sensors and operator inputs in order to determine the sub-modes.

The Deceleration Button

When wingborne with the Nominal system, the high-lift scheduling is such that the drag is minimized. Sections 3.4.2 and 3.4.4.3 of Chapter 3 explained the automation algorithms that facilitate this.

However, in certain situations it may be favorable to deploy the flaps such that the drag is maximized. This is the case when a fast rate of deceleration is pursued. Using the proposed methods of this thesis, the only available option is to deflect the throttle inceptor into the retransition region and thereby inducing the transition of the State Machine M_{HL} to the state *Extend*. This, however, may be an undesirable solution to the posed problem. The reason for this is that this action would also trigger the fully automated aircraft retransition process.

The currently implemented automation addresses the shortcoming above via a dedicated input item that is here referred to as “deceleration button”. The decision-execution module of the Nominal system processes this input and if evaluated to be engaged, the scheduling switches accordingly such that the flaps are extended. This is only enabled once the aircraft has decelerated below the structural limit speed of V_{FE} and the amount of deployment is such that no damage to the airframe may ensue. This is in accordance with Equation 3.70 which was introduced in Chapter 3.

The proposed solution does not add complexity to the decision-making process. In fact, releasing the input item enforces the default scheduling process. However, the deceleration capability of the aircraft is increased substantially.

Takeover by the Nominal System

In the thesis a takeover performed by the Fallback system was discussed. The reasons for a takeover include a design error in the Nominal system algorithms, hardware errors of that system or an operator command to change the system.

In Sections 4.2.3 and 4.4.4, the algorithms for correct Fallback system initialization following a takeover were introduced and analyzed. The selection of applicable Fallback system mode after the switch depends on the states of the Nominal system prior to the takeover.

In this thesis a takeover from Fallback to Nominal system was not discussed. A use-case for this is not a fault in the Fallback system but is rather driven from a flight testing perspective. For example, it may be desired to take-off with the Fallback system and engage the Nominal system only in-air initially. Another scenario may be to transition with the Fallback system and test solely the Nominal system wingborne mode.

For this purpose, a takeover functionality of the Nominal system was implemented within the development effort of the industry funded project. In contrast to the Fallback system takeover, the switch to Nominal system is performed only on operator request and an automatic change is not permitted.

The implemented initialization of the Nominal system resembles the one of the Fallback system as explained in Section 4.2.3. In fact, although not fully, the initialization method was introduced with Figures 5.12 and 5.15 found in Chapter 5.

In essence, the starting state selection of the Nominal system State Machine M_{LTU} is done by inverting the Truth Table 4.3. Thereby, the last valid command of the Fallback system to the LTUs is used to distinguish between the states *Disengaging* and *Disengaged* so as to not produce a step in the commands. Using the outcome of this evaluation, a subsequent function calculates the starting states of the high-lift State Machine M_{HL} . The current kinematic speed is used for the initialization of s_{HS} .

Fallback Graceful Degradation Capability

The Fallback system utilizes considerably less sensors in order to function properly. This was discussed in Section 2.4.1.2 of Chapter 2. According to Table 2.7 of that section, the Fallback law controls the aircraft attitude among other variables. Therefore, it requires sensor information of these vehicle states.

However, in the cases where these sensor data are unavailable, additional modes of the control algorithms are foreseen. This includes the possibility for the operator to directly control the body rates. Such a functionality increases the availability of the Fallback system considerably, since solely gyroscopic sensor data would be required.

The currently implemented automation module includes an additional State Machine that was not presented in this thesis. Its task is to track the control algorithm's mode in terms of tracked command variables. It therefore accounts for the pilot's requests to degrade to even lower degrees of automation control laws or to increase the level of control concept automation. In addition, automatic provisions revert to less automated laws in the cases of sensor loss. The latter function implements a graceful degradation [140] capability to the Fallback system's operation.

Fallback Rate Mode Transition and Retransition

The above-mentioned degraded modes of the Fallback system's law introduce additional complexity to the transition and retransition automation of that system. The reason for this is as follows.

Whenever in powered-lift flight with the default mode that is presented in Section 2.4.1.2 of Chapter 2, the aircraft's pitch is scheduled along the airspeed. Therefore, the operator neither has an influence on that attitude state, nor is manual control required.

However, whenever degraded in the "rate" mode, the body-pitch rate is directly controlled by the pilot as by implication the attitude may not be available to the system. The need for manual body-pitch rate input therefore requires an additional control inceptor axis and thus implicates modifications in the transition and retransition automation. The last completed work effort is the specification of an additional transition and retransition automation mechanism using the supplementary operator input.

The suggested solution utilizes the same State Machine specification in terms of decision-making. However, the decision-atomics include provisions to account for the additional input. At the time of writing this thesis, different input strategies are under consideration. The proposed method can be applied in all current input possibilities.

7.2 Ongoing Efforts

This section lists the development effort that is ongoing at the time of writing this thesis. The topics here contain either partially ready solutions or ones that require further verification and validation. They are listed in order of completeness.

Fallback Mode Variable Thrust

As discussed in Section 2.4.1.2 of Chapter 2, whenever Fallback system is operational, the operator commands the traction thrust via the throttle control inceptor's longitudinal axis.

The transition and retransition procedures and automation functions of the Fallback system pose requirements on the throttle settings at the different control inceptor deflections. This was performed in Section 4.3.2 of Chapter 4. According to Section 4.3.2, the thrust at the above-mentioned settings is such that a predefined airspeed is achieved during steady-state straight and level flight.

The throttle command is therefore estimated using model parameters, such as the aircraft drag, the traction propeller's thrust coefficients and more. Therefore, uncertainties may lead to deviations in the achieved airspeed.

In the worst case, this discrepancy between expected and achieved airspeed may lead to issues when performing the transition and retransition. Namely, under undesirable circumstances it could be that the stall speed is not exceeded when performing the transition. Alternatively, it could be that the aircraft is incapable of decelerating below the structural limit speeds during retransition.

A current solution is being developed, whereby this potential issue is mitigated. In summary, the proposal utilizes an additional input, with which the pilot is capable of commanding an offset to the thrust that is allocated by the throttle control inceptor. It operates similar to a trim function. Thereby, the pilot is capable of performing the prescribed procedures in the initial flight testing where the aircraft parameters are not well identified and are prone to uncertainties.

Selection Logics Decentralized Algorithms

Due to the distribution and replication of the functions onto different physical flight control computers, in the actual FCS certain activities are the joint decision of all involved components. Such a function is for example the algorithm that chooses the system in command - Nominal or Fallback.

In the behavioral specification model that was presented in Chapter 5, the selection algorithms are performed in a centralized manner in order to reduce the development effort while at the same time retaining a feasible system response. The specification of the command selection is available and development is ongoing to design and validate the decentralized algorithms that are executed on each physical instance within the FCS.

The command selection is responsible for automatic takeovers between flight control computers due to unavailability of the commanding instance. This can be due to complete loss of function in both Nominal and Fallback systems. In addition, certain errors that lead to graceful degradation may lead to automatic switches in the commanding instance. The validation of the algorithm includes testing for robustness against timing effects using the framework, developed at TUM-FSD and found in [141].

Code Compliance

All automation functions that are in the scope of this thesis are intended for flight and as such will be embedded onto the aircraft's avionic equipment. As a consequence, the algorithms need to be implemented in accordance with modeling guidelines in order to guarantee certain properties following code generation. Available guidelines can be found in [81, 82].

The current developing effort follows the TUM-FSD development process in order to guarantee a satisfactory level of software quality and maintainability. It utilizes the tools found in [142]. In essence, all proposed methods are being modeled in a code-compliant manner. In addition, robustness measures with regards to sensor noise, inaccuracies and model uncertainties are considered.

Formal Verification

This thesis provided analysis and simulation methods as means of ensuring the correctness of the proposed solutions. Ongoing effort aims to rigorously test the automation algorithms intended for embedding on the aircraft equipment.

In addition to testing, the aerospace community recognizes the benefit of formal methods as a means of validation in the RTCA DO-333 [143]. Formal proof uses mathematical methods to prove the correctness of the algorithms with regards to their specification. More information on the topic is available in [144].

The Simulink Design Verifier [145] is utilized in ongoing verification activities to demonstrate critical automation function properties. Thereby certain requirements are formally defined in order to check against. Examples include the impossibility to engage the flight control laws on ground in the event where a critical number of LTUs are evaluated to fail, that the LTUs will not be enabled above the structural limit speed and more.

Variable Behavioral Specification Model Capabilities

The behavioral specification model presented in Chapter 5 offers multiple functions. Introduced were the activation procedures and the automation methods from Chapters 3 and 4. However, in ongoing development efforts, multiple other functionalities are implemented and tested.

Among others, those include altitude protections, terrain following, the capability of wingborne landing as a possible contingency, the above-mentioned flap extension using the “deceleration button” and many more.

In addition to this, multiple simulator concepts are designed and validated using the behavioral specification model found Chapter 5. As a consequence of this and certain system-architecture dependencies, it could be the case that certain developed functions are not available or may vary in execution among different configuration.

To address this issue, so-called “variants” of the behavioral specification model are created. A single model is utilized for all different constellations of enabled and altered functions. However, the simulation functionality is configurable by controlling the active functions with configuration parameters. They allow to enable, disable or alter the individual functions.

As a consequence, solely one model is maintained for all developed simulator and architecture concepts. In addition, the behavioral specification model can be utilized for future applications with little added effort.

The approach has several disadvantages. Namely, the possibility to reproduce behavior of multiple concepts is associated with additional complexity within the model due to the above-mentioned parameterization. In addition, the proposed solution can only work efficiently if the structure of the core functionality does not change fundamentally.

7.3 Outlook and Perspectives

This section lists topics that are associated with the methods found in this thesis that are at this time not planned but may expand the state of technology further if explored.

Provisions for Tilt-Rotor Aircraft

The transition and retransition automation that was designed in this thesis manages the control concepts for lift-to-cruise aircraft. However, there are other vehicle topologies that are capable of both VTOL and wingborne flight. These are tilt-rotor aircraft.

Modifications of the automation could be envisioned, with which the transition and retransition could be performed with tilt-rotor aircraft. In fact, [114] is an early publication of this author and it proposes concepts for the automatic management of LTUs of an unmanned vehicle that has both hover propulsion units and tilting propellers used for wingborne flight.

The methods found here and in [114] could be used to develop industry-compliant automation for manned tilt-rotor aircraft configurations. Thereby, the positions of the tilts could be used in the evaluation whether wingborne flight has been entered¹. Subsequently, deflection of the tilt nacelles can be prohibited until requested by the operator².

Unarguably, challenges and considerations can arise due to the severely different aircraft constellation. However, many of the notions and methods found in this thesis can be utilized to address them.

Fully Automated Flight

In the current proposal, the flight with the Nominal system is solely manual. However, higher order autopilot functions will sooner or later undoubtedly be utilized in order to operate the aircraft. Highly-likely this will be done with the Nominal system.

The proposed automation concept guarantees a large amount of safety properties when operating with the Nominal system. Therefore, it is not advisable to discard the methods here, but rather utilize them as the basis for higher order automation functions instead. This can be done as follows.

The Nominal system automation's decision-atomics module evaluates the operator intentions via control inceptors and other input items. This was explained at length in Chapter 3. The operator input is processed to input symbols for the Finite-State Automata such as $trans_{rqst}$ and $retrans_{rqst}$ for the LTU automation or $extend_{rqst}$ and $retract_{rqst}$ for the high-lift system management. Instead of processing the inputs of the operator, these variables can be substituted with automatic flight function commands whenever it is engaged.

As a consequence, the autopilot functions can seamlessly be integrated into the Nominal system's automation design found in this thesis and the automation concept can retain all favorable characteristics that are summarized in **Contribution 1**. However, certain challenges need to be addressed.

More precisely, the Nominal system automation relies on the operator input as an independent source for the transition conditions. For example, the LTU disengagement is only initiated if three conditions are fulfilled - if a safe airspeed is reached, if the LTUs are not utilized for force and moment production *and* the operator requests wingborne flight. If the pilot commands to the automation are substituted by commands that originate by other automation functions, then a degree of independence is lost.

This may not pose a problem in the failure-free case, but it definitely requires extensive analysis to substantiate such a claim. In the failure cases where additional actions from the operator (the transition and retransition mitigation strategies) are required, initially the automation can still rely on the pilot input. However, once going in direction of fully automated flight and as per SVO3, it can be assumed that the operator has no piloting skills

¹Equivalent or in addition to the condition LTU_{UNUSED} found in Equation 3.15 of Chapter 3.

²E.g. only if $retrans_{rqst}$ is *true* as defined in Equation 3.4.

and thus a competent decision-making process. For recollection, the SVO levels are found in Section 1.2.2 of Chapter 1. In such situations, designing an automatic decision-making process will unarguably be a non-trivial task if varying actions are possible.

State Machine Transition Condition Modifications

Currently, the design of the automation explicitly aims to minimize the complexity. Certain robustness modifications may need to be added depending on the applicability of abnormal scenarios.

For example, currently the Nominal system automation's transition condition that changes the state s_{LTU} from *Disengaging* to *Disengaged* is solely the evaluation whether the LTUs are fully shut down and is performed as per Equation 3.21. In the state *Disengaging* the control allocation is given the task to drive the powered-lift system from the current command down to zero. This process is deterministic and the duration of the ramp down to zero RPM is known.

At the same time, provided the LTUs are incapable of shutting down, a warning is thrown and actions are expected from the operator. This was explained at length in Chapter 3. The pilot's decision-making process is not time critical and may take arbitrarily long.

A use-case could be envisioned, where the problematic LTU has loss of torque and therefore could windmill due to the propeller inflow. Depending on the aircraft state and the environmental conditions, the windmilling may stop, at which point the transition condition as per Equation 3.21 will become *true* which will cause the state s_{LTU} to change to *Disengaged*.

This event is not necessarily hazardous from a design point-of-view. However, from the operator perspective this may cause confusion. The reason for the commissioned warning for required crew actions is the incapability to disengage the LTUs by the automated system. However, in the meantime the disengagement occurs without any operator input.

If deemed necessary, modifications of the transition conditions may be performed to alleviate this scenario. In the constructed example, Equation 3.21 can be modified to

$$\bar{t}_2 = LTU_{OFF} \wedge \neg LTU_{disengtimeout} \quad (7.1)$$

in order to forbid the automatic transition once the warning is thrown via the variable $LTU_{disengtimeout}$.

Other Forms of Haptic Feedback

The throttle inceptor presented in Section 2.4.3.1 of Chapter 2 had various tactile queues and haptic feedback possibility, with which awareness with relation to the state of automation can be facilitated.

Depending on different design considerations or hardware limitations, the inceptor in future airframes may not be manufactured exactly as prescribed in the above-mentioned section. For example, the barrier functions may be omitted. In the cases where the previously enforced regions are exceeded by the operator, aural queues may be included instead. Therefore, this would need to be considered by the automation design.

Regardless of how the inceptor is modified, the design must have the specific divisions that are mentioned in Equations 2.35 and 2.36. Otherwise, proper awareness, automation decision-making ambiguity or Nominal and Fallback system procedure harmonization cannot be ensured with the provided solutions.

Flight Testing Provisions

Certain transition conditions in the Nominal system's automation are evaluated using parameters that are determined using simulation results or analytical methods such as linearization and trim tools.

An example includes the evaluation with regards to the usage of the LTUs. For recollection, the powered-lift system is deemed unused if all units are in the vicinity of the idle RPM as per Equation 3.16. This is evaluated using thresholds that may differ for each LTU depending on the LTU size, on net moments due to aerodynamic effects or on the presence of failures. An example for the latter is a traction unit failure that causes a set of LTUs to have much larger rotation rates to counteract the induced yaw moment.

The prescribed thresholds must be chosen as low as possible in order to reduce the possibility of a dive due to a reduction of the lift following the LTU disengagement. However, opting for too "pessimistic" values may render the transition conditions of the State Machines to never trigger.

This problem is exacerbated by the fact that the parameters are subject to uncertainties due to mismatch between model and actual airframe, unaccounted aerodynamic effects and more. This is especially relevant in early flight testing, where the system identification is still lacking.

Therefore, provisions may be added to the existing design, in which this issue is mitigated. A possible solution is an additional operator input item. In the event where the LTUs are indeed unused but a certain set is above the intended thresholds, the pilot can make use of this input in order to trigger the disengagement procedure. This mitigation augments solely the variable LTU_{UNUSED} which is used in combination with the remainder of the transition conditions as per Equation 3.19.

Additional protections must be added so as to mitigate an inadvertent start of deactivation while the LTU RPM is still too high. For example, two sets of thresholds can be foreseen. The first are the intended ones, which will be the ones in the end-design once the algorithms are flight-proven. They are the ones that are not satisfied in the constructed example. The latter are thresholds, under which the LTUs need to be in order to trigger

the disengagement in combination with the mentioned pilot input item. Thus, the latter parameters can be chosen to be much larger and thus availability of the wingborne flight phase can be guaranteed.

Appendix A

Relationship Between Transition Sets, Conditions and Functions

Finite-State Machines in Stateflow are modeled using charts. The mechanics of the automaton are described with so-called transition conditions. However, the theory on Finite-State Automata defines the changes of states with using transition functions. This appendix entry explains the dependencies between transition functions from the theoretical standpoint and the transition conditions from the implementation standpoint. For this, the so-called transition sets are defined.

Suppose a State Machine with the states $s0$, $s1$ and $s2$, whereby $s0$ is the starting state. It has the inputs a , b , and c from the boolean set. Suppose the State Machine must transition from $s0$ to $s1$ if c is *true*. In addition, it transitions from $s0$ to $s2$ if $\neg c \wedge (a \vee b)$ is *true*. Otherwise it remains in the state $s0$. All other details are irrelevant for the scope of this discussion. This automaton is depicted in Figure A.1, where on the left hand side the State Machine is depicted and on the right hand side - the Stateflow chart.

Given the conditions of switching to states $s1$ and $s2$, the transition conditions

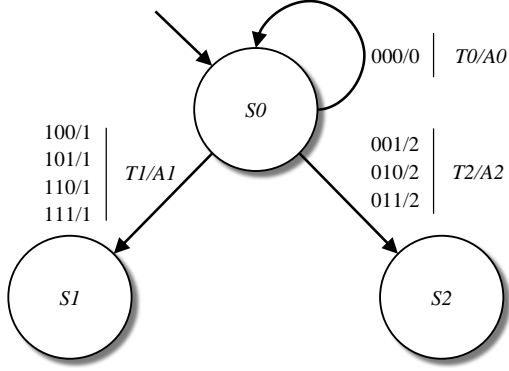
$$\begin{aligned} t_1 &= c \\ t_2 &= \neg c \wedge (a \vee b), \end{aligned} \tag{A.1}$$

which trigger the state change in the chart operation as found in Subfigure A.1b. Let $T1 \subset U$ and $T2 \subset U$ be the sets of all input combinations that cause the respective transitions in the chart of Subfigure A.1b. Therefore, $T1$ and $T2$ include all input combinations for which t_1 and t_2 respectively are satisfied, i.e.

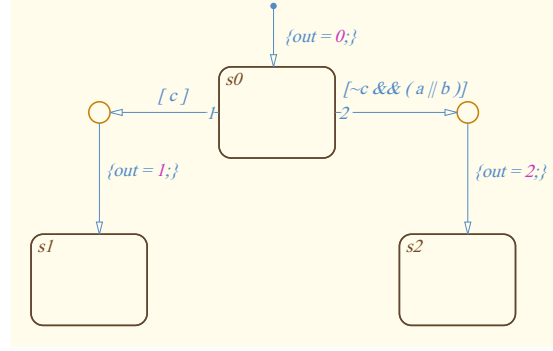
$$\begin{aligned} T1 &= \{\{100\}, \{101\}, \{110\}, \{111\}\} \\ T2 &= \{\{001\}, \{010\}, \{011\}\}. \end{aligned} \tag{A.2}$$

Therefore, if u_1 and u_2 are inputs belonging to the transition sets $T1$ and $T2$ respectively, then it holds that

$$\begin{aligned} \delta(s0, u_1) &= s1 \\ \delta(s0, u_2) &= s2. \end{aligned} \tag{A.3}$$



(a) Depiction of a State Machine



(b) Depiction of a Chart

Figure A.1: State Machine and Chart Relationship

Lastly, all remaining input combinations can be expressed as an additional set, in this example namely as

$$T0 = U \setminus (T1 \cup T2) = \{\{000\}\}. \quad (\text{A.4})$$

Therefore, for every $u_0 \in T0$ it follows that

$$\delta(s0, u_0) = s0. \quad (\text{A.5})$$

To summarize - the transition condition describes the conditions, under which a transition from one state to the other occurs. The transition set is the set of all input combinations that satisfy the given transition condition. Finally, a transition function exists for each member of the transition set that can be used to formally describe the automaton.

Appendix B

High-Automation Transition, Retransition and Mitigation Flow

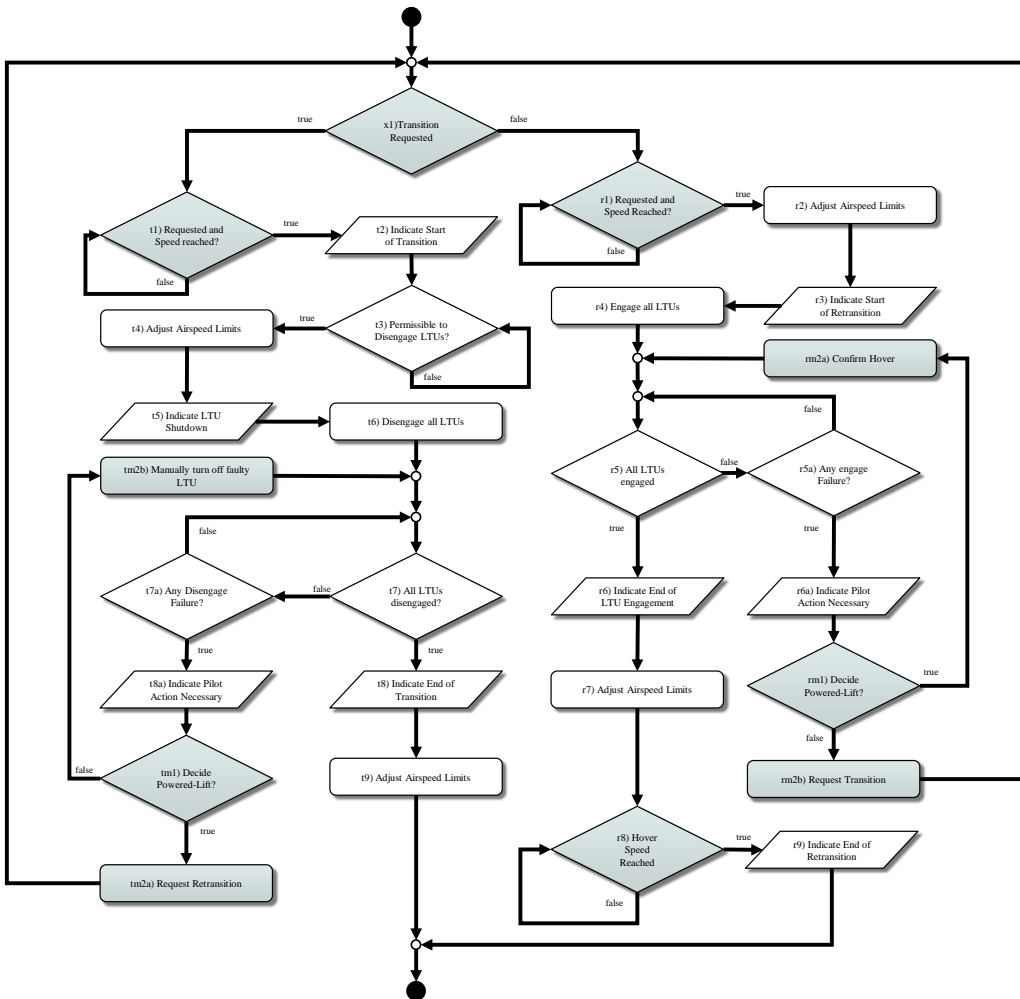


Figure B.1: Complete Transition and Retransition Process Flow. The Chart Assumes no Crew Deviations.

Appendix C

High-Automation Flight Failure Mode and Effects Analysis

Table C.1: *High-Degree of Automation Control Concept Exhaustive Failure-Modes and Effects Analysis*

Function	Failure Mode	Flight Phase	Resulting Behavior	Comment
Schedule Airspeed Limits	LTU Failure to 0 RPM	Hover	No difference in the resulting behavior.	Aircraft Controllable.
	Erroneous non-zero RPM	Hover	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	High-Lift Unit stuck extracted	Hover	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	High-Lift Unit stuck retracted	Hover	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	High-Lift Unit Jammed in non-extremum	Hover	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	High-Lift Unit hardover	Hover	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	LTU Failure to 0 RPM	Transition	No difference in the resulting behavior.	Aircraft Controllable.
	Erroneous non-zero RPM	Transition	Upper airspeed limit is retained to V_{LSNE} .	Structural damage needs to be mitigated.
	High Lift Unit stuck extracted	Transition	No difference in the resulting behavior.	Nominally set to V_{LSNE} .
	High Lift Unit stuck retracted	Transition	Lower airspeed limit is set to V_{SAFE} after the LTU shutdown has started.	Disengagement at higher airspeeds handled by "Perform Automatic Transition"

Schedule Airspeed Limits (continued)	High Lift Unit jammed in non-extremum	Transition	Lower airspeed limit is set to V_{SAFE} after the LTU shutdown has started.	Disengagement at higher airspeeds handled by "Perform Automatic Transition"
	High Lift Unit hardover	Transition	Unless hardover to extracted, lower airspeed limit is set to V_{SAFE} after the LTU shutdown has started.	Disengagement at higher airspeed handled by "Perform Automatic Transition"
	LTU Failure to 0 RPM	Wingborne	No difference in the resulting behavior.	Should be off.
	Erroneous non-zero RPM	Wingborne	Upper airspeed limit is set to V_{LSNE} .	Structural damage needs to be mitigated.
	High-Lift Unit stuck extracted	Wingborne	Upper airspeed limit is set to V_{FE} . Lower airspeed limit is set to V_{SAFEFE} .	Structural damage needs to be mitigated.
	High-Lift Unit stuck retracted	Wingborne	No difference in the resulting behavior.	Lower airspeed limit is nominally set to V_{SAFE} in wingborne.
	High-Lift Unit jammed in non-extremum	Wingborne	Upper airspeed limit is set such that no structural damage can be caused. The setting is a function of the maximum flap deflection of all units.	Structural damage needs to be mitigated.
	High-Lift Unit hardover	Wingborne	Upper airspeed limit is set such that no structural damage can be caused. The setting is a function of the maximum flap deflection.	Structural damage needs to be mitigated.
	LTU Failure to 0 RPM	Retransition	Until retransition confirmed by crew, lower airspeed limit is set to V_{SAFE} or V_{SAFEFE} depending on the high lift units position.	Ensure crew awareness of latent error prior to entering the hover region.

Function	Failure Mode	Flight Phase	Resulting Behavior	Comment
Schedule Airspeed Limits (continued)	Erroneous non-zero RPM	Retransition	Until retransition confirmed by crew, lower airspeed limit is set to V_{SAFE} or V_{SAFEFE} depending on the High-Lift Unit positions.	Ensure crew awareness of latent error prior to entering the hover region.
	High-Lift Unit stuck extracted	Retransition	No difference in the resulting behavior.	Upper airspeed limit is nominally set to V_{LSNE} in hover and (re)transition.
	High-Lift Unit stuck retracted	Retransition	Lower airspeed limit is set to V_{SAFE} .	Prevent stall.
	High-Lift Unit jammed in non-extremum	Retransition	Lower airspeed limit is set to V_{SAFE} .	Prevent stall.
	High-Lift Unit hardover	Retransition	Unless hardover to extracted, lower airspeed limit is set to V_{SAFE} .	Prevent stall.
	LTU Failure to 0 RPM	Hover	The crew is alerted of the loss of LTU.	
	Erroneous non-zero RPM	Hover	The crew is alerted of the loss of LTU.	
	High-Lift Unit stuck extracted	Hover	No difference in the resulting behavior.	
	High-Lift Unit stuck retracted	Hover	No difference in the resulting behavior.	
	High-Lift Unit jammed in non-extremum	Hover	No difference in the resulting behavior.	
Provide Transition	High-Lift Unit hardover	Hover	No difference in the resulting behavior.	
	LTU Failure to 0 RPM	Transition	No difference in the resulting behavior. The crew is alerted of the loss of LTU.	

Provide Transition (Continued)	Erroneous non-zero RPM	Transition	The crew is alerted of the loss of LTU. The crew is alerted that action is necessary. Transition can be terminated and aircraft can revert to hover. Transition can be completed by removing the power of the LTU, thereby turning it off.	The error mode requires the off-nominal procedure.
	High-Lift Unit stuck extracted	Transition	No difference in the resulting behavior.	
	High-Lift Unit stuck retracted	Transition	LTU shutdown performed after V_{SAFE} instead of V_{SAFEFE} . The crew is alerted that the shutdown is to be performed at higher speeds.	
	High-Lift Unit jammed in non-extremum	Transition	LTU shutdown performed after V_{SAFE} instead of V_{SAFEFE} . The crew is alerted that the shutdown is to be performed at higher speeds.	
	High-Lift Unit hardover	Transition	Unless hardover to extracted, LTU shutdown performed after V_{SAFE} instead of V_{SAFEFE} . The crew is alerted that the shutdown is to be performed at higher speeds.	
	LTU Failure to 0 RPM	Wingborne	The crew is alerted of the loss of LTU.	
	Erroneous non-zero RPM	Wingborne	The crew is alerted of the loss of LTU.	
Provide Retransition	High-Lift Unit stuck extracted	Wingborne	No difference in the resulting behavior.	

Function	Failure Mode	Flight Phase	Resulting Behavior	Comment
Provide Retransition (Continued)	High-Lift Unit stuck retracted	Wingborne	Motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	
	High-Lift Unit jammed in non-extremum	Wingborne	Motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	
	High-Lift Unit hardover	Wingborne	Unless extracted, motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	
	LTU Failure to 0 RPM	Retransition	The crew is alerted of the loss of LTU. The crew is alerted that action is necessary. Retransition can be terminated and aircraft can revert to wingborne. Retransition completion by crew confirmation.	Ensure crew awareness of latent error prior to entering the hover region.
	Erroneous non-zero RPM	Retransition	The crew is alerted of the loss of LTU. The crew is alerted that action is necessary. Retransition can be terminated and aircraft can revert to wingborne. Retransition completion by crew confirmation.	Ensure crew awareness of latent error prior to entering the hover region.
	High-Lift Unit stuck extracted	Retransition	No difference in the resulting behavior.	
	High-Lift Unit stuck retracted	Retransition	Motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	High-Lift System should be extracted in retransition.

Provide High-Lift Commands	High-Lift Unit jammed in non-extremum	Retransition	Motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	High-Lift System should be extracted in retransition.
	High-Lift Unit hardover	Retransition	Unless extracted, motor engagement performed with the off-nominal High-Lift position. The crew is alerted about the change in procedure.	
	LTU Failure to 0 RPM	Hover	No difference in the resulting behavior.	
	Erroneous non-zero RPM	Hover	No difference in the resulting behavior.	
	High-Lift Unit stuck extracted	Hover	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit stuck retracted	Hover	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit jammed in non-extremum	Hover	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit hardover	Hover	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	LTU Failure to 0 RPM	Transition	No difference in the resulting behavior.	
	Erroneous non-zero RPM	Transition	Follow the “Flap optimal position” after operator confirmation.	Provide longer range.

Function	Failure Mode	Flight Phase	Resulting Behavior	Comment
Provide High-Lift Commands (Continued)	High-Lift Unit stuck extracted	Transition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit stuck retracted	Transition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit jammed in non-extremum	Transition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit hardover	Transition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	LTU Failure to 0 RPM	Wingborne	No difference in the resulting behavior.	
	Erroneous non-zero RPM	Wingborne	No difference in the resulting behavior.	
	High-Lift Unit stuck extracted	Wingborne	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit stuck retracted	Wingborne	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit jammed in non-extremum	Wingborne	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	

Provide High-Lift Commands (Continued)	High-Lift Unit hardover	Wingborne	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	LTU Failure to 0 RPM	Retransition	No difference in the resulting behavior.	
	Erroneous non-zero RPM	Retransition	No difference in the resulting behavior.	
	High-Lift Unit stuck extracted	Retransition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit stuck retracted	Retransition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit jammed in non-extremum	Retransition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	
	High-Lift Unit hardover	Retransition	Other unit follows erroneous to prevent asymmetric deflection. The crew is alerted of the loss of High-Lift Unit.	



Appendix D

How Variable Starting States can be Achieved

Formally, a Mealy Machine description requires the specification of its starting state. According to the theory of Finite-State Automata, there can only be one starting state (or starting state tuple). These properties are mentioned in Section 2.2.3. However, in order for the takeover to correctly operate in Chapter 4, the *Initialize* function is introduced in Equation 4.18. The initial states of the automaton M_{FB} are determined via the function. This appendix provides an argumentation why this is permissible.

Part of the FCS operation is the decision which system is in command. This function is usually a distributed algorithm in the case of redundant flight control computers. The exact function specification is not in the scope of this thesis, but it must decide in some manner when a takeover must occur. This can for example be due to a critical sensor loss, after which the Nominal system cannot operate, a hardware fault on the Nominal system component and more.

Even if no Nominal system is being executed during a given mission, there still needs to be a phase, in which the automation (and control concept) are not engaged. This is necessary in order to for example to verify the correctness of the input signals and initialize correctly.

As a consequence, there is a phase in the operation of the system, in which it is disarmed. This initial phase is the true starting state of the State Machine. Afterwards, when the main automation is engaged (due to a takeover or simply after proper initialization), a transition from this starting state to the required state constellation can be executed and in the case of the Fallback system, it can be performed with the *Initialize* function. Figure D.1 exemplifies this effect. In Figure D.1, the contents of M_{FB} could be placed in the “Engaged” state. The state transitions from “Armed” to any other state within “Engaged” can be in accordance with *Initialize*. This way of observation is omitted in Chapter 4 for the sake of simplicity and readability.

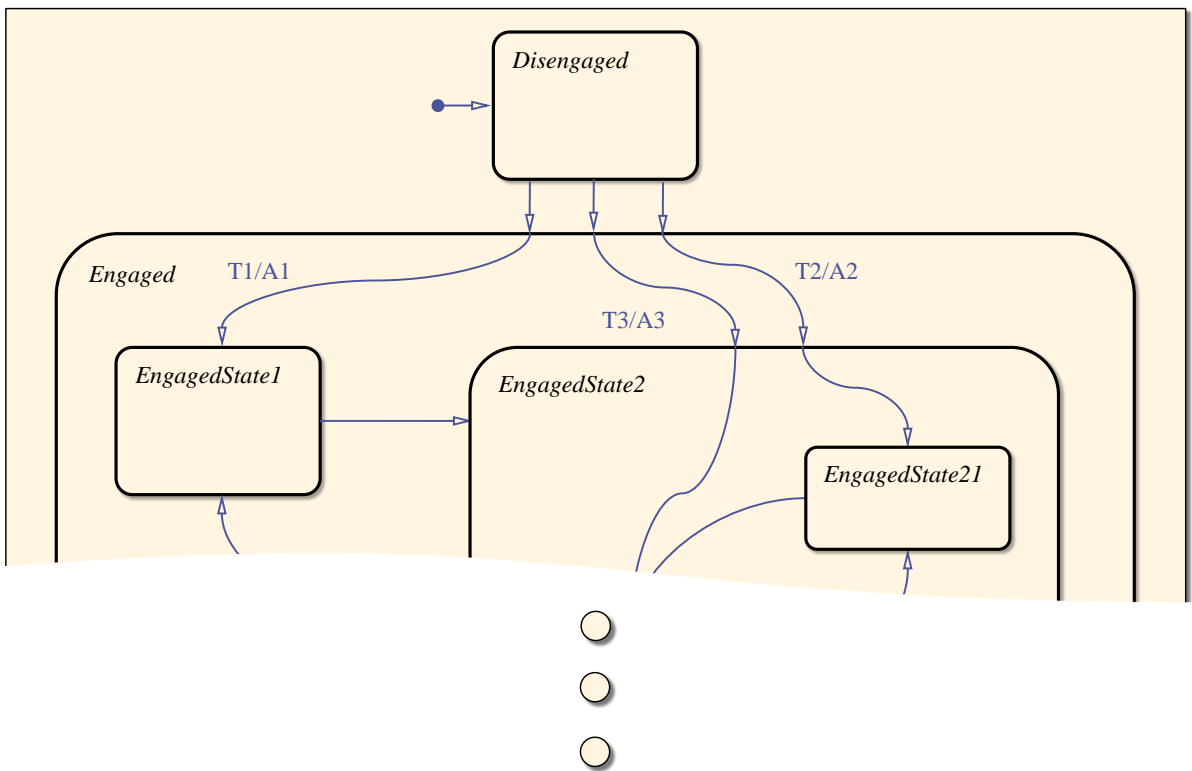


Figure D.1: *Example of How Different Starting States can be Achieved*

Appendix E

Fallback Transition, Retransition and Mitigation Flow

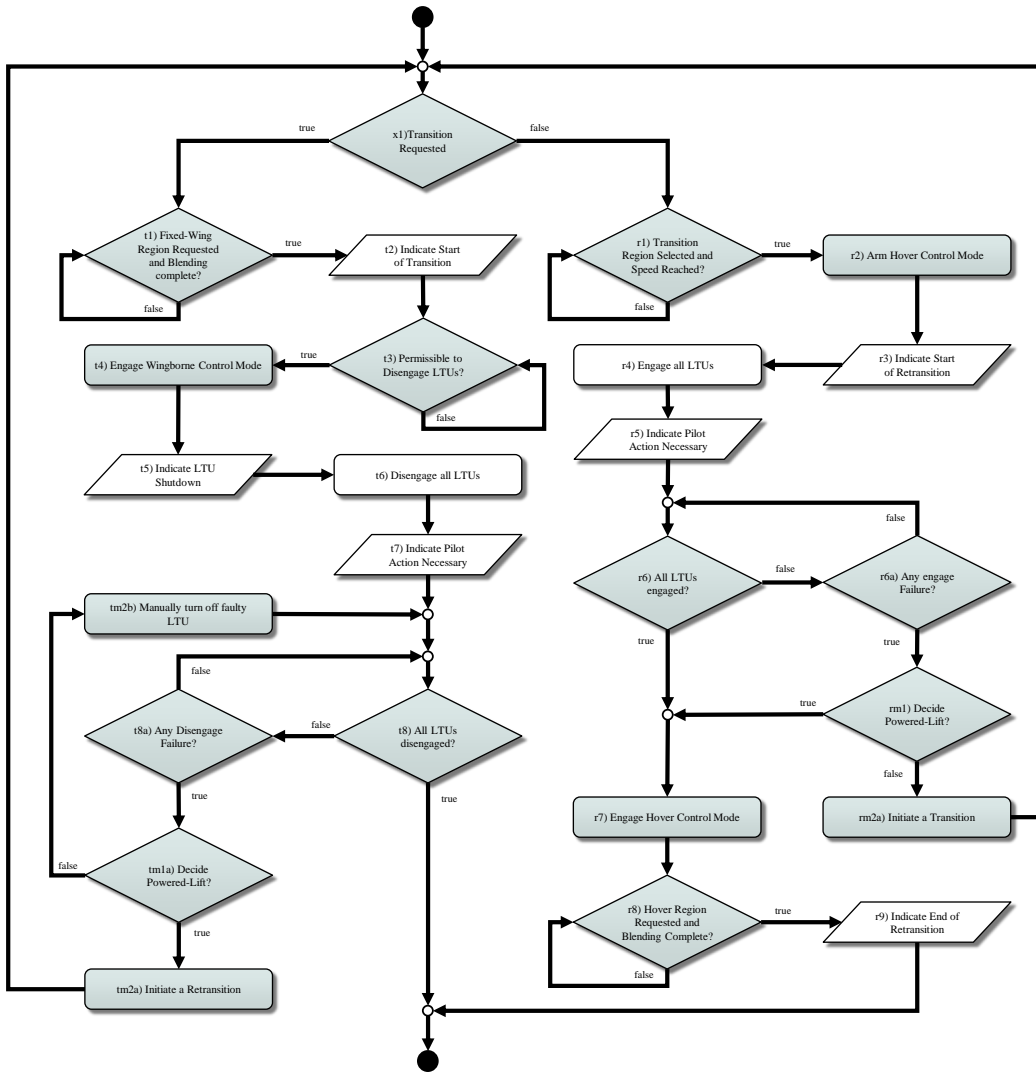


Figure E.1: Complete Transition and Retransition Process Flow. The Chart Assumes no Crew Deviations.

Appendix F

Behavioral Specification Model Implementation Examples

Figure F.1 exemplifies the behavioral specification model error detection. As discussed in Chapter 5, this is done in a highly simplified manner by utilizing Confirmation Counters and the error injection signals. One exception here is a functional failure of the Nominal system components, which is registered by the functional monitor. An exemplary integration model is found in Figure F.2, where the two DRM modules and the disengaged closed-loop behavior is deployed. Based on the vehicle automation state, the command selection forwards the needed data to the simplified plant model. The Automation integration model is shown in Figure F.3. It contains the vehicle automation and the individual control mode automation modules, found in the “Law Automation” integration model. Figure F.4 demonstrates the utilization of the instrument and error injector blocks for the Nominal DRM that reproduces the Nominal system control mode. Lastly, Figure F.5 demonstrates how the Decision-Making modules of the control concept are enabled only when engaged. This is mentioned in Section 5.3.4.3 and is necessary in order to execute the law-specific automation only when selected.

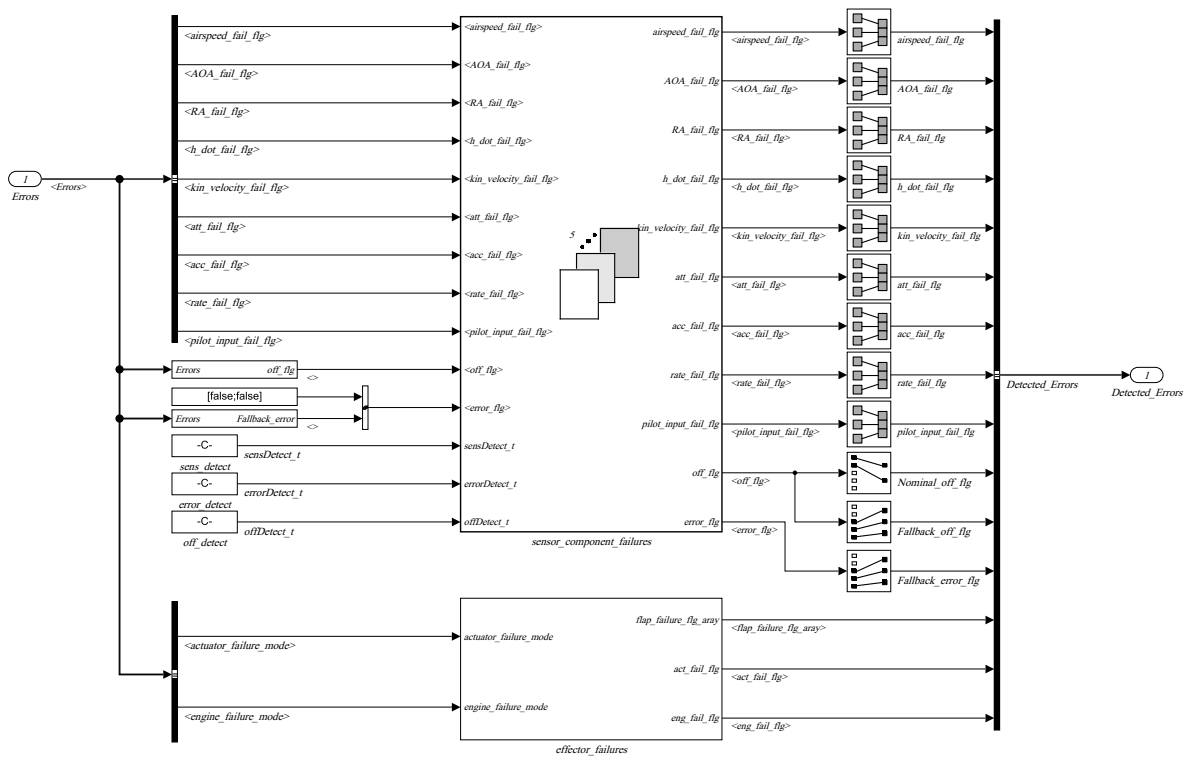


Figure F.1: Error Detection

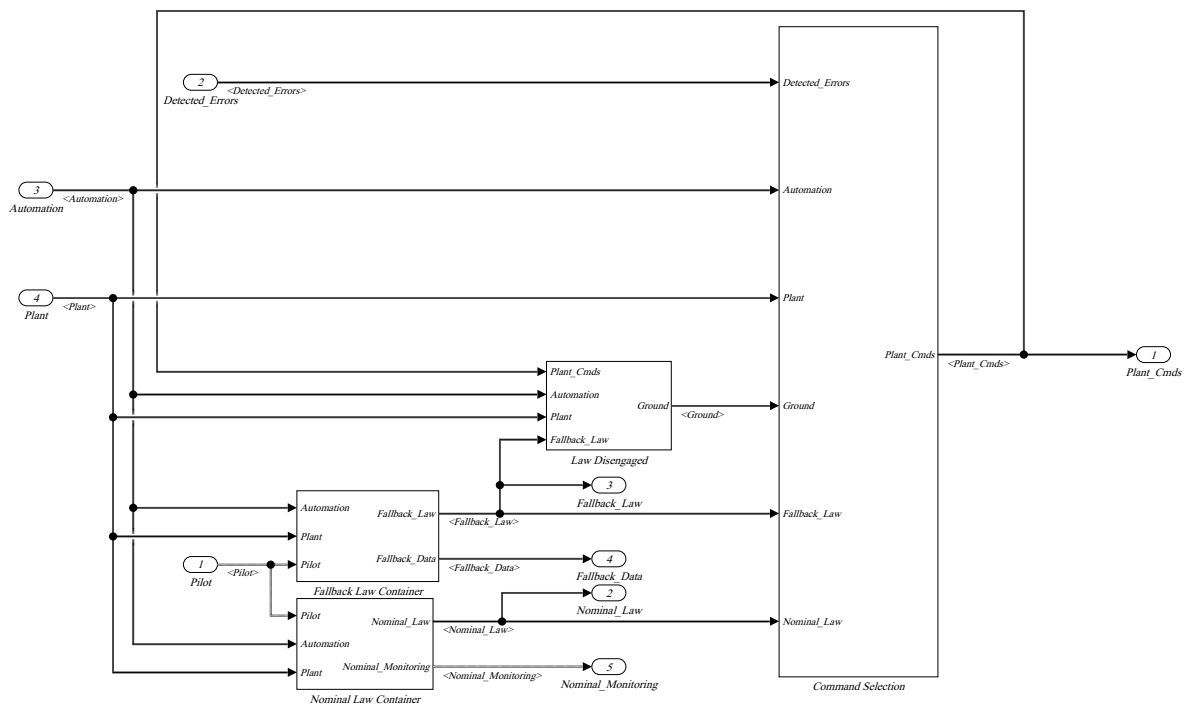


Figure F.2: Law Integration Model

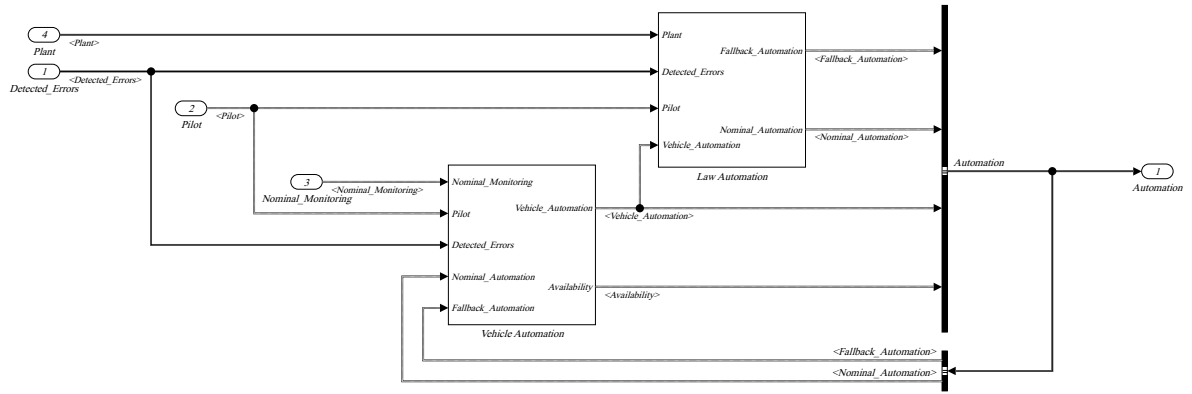


Figure F.3: Automation Integration Model

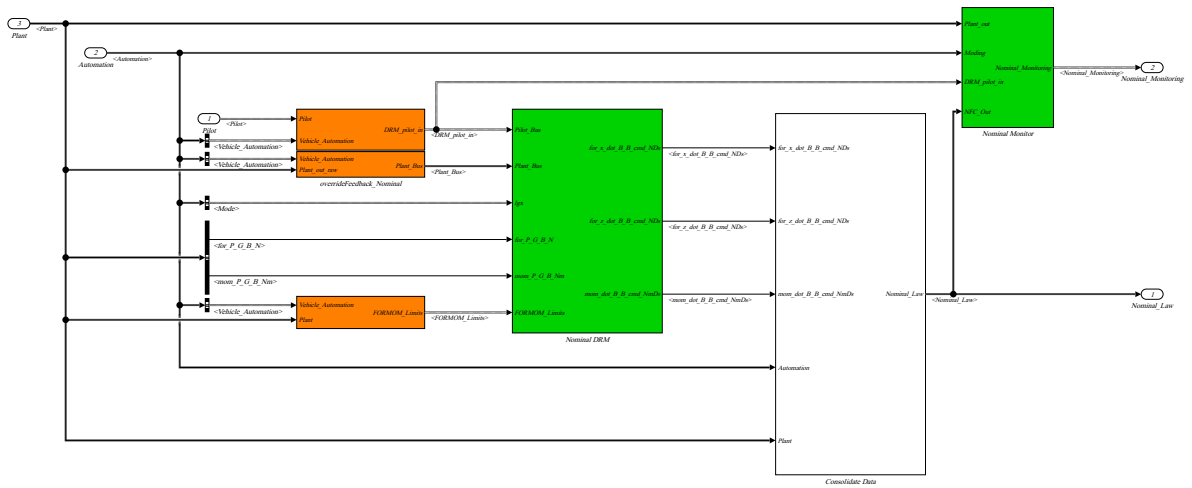


Figure F.4: *Nominal DRM Integration Model: The Error Injections and Interface Modules are Colored in Orange*

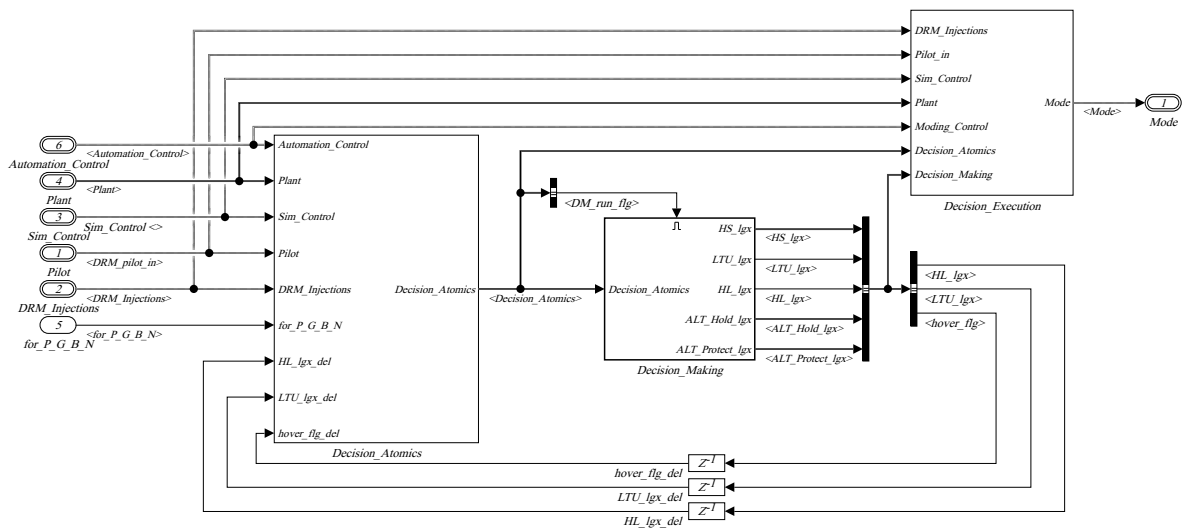


Figure F.5: *Nominal System Automation: The Decision-Making Module is in an Enabled Subsystem*

Appendix G

TUM-FSD Simulator Pilot Checklist

Certain pages of the checklists are removed as they are subject to non-disclosure. The checklists follow the patterns, utilized by Diamond Aircraft. They can be found in [146].

BEFORE ENGINE START CHECK

1	Preflight Check.....	COMPLETED	1
2	HT HOVER DISABLE	OFF	2
3	HTA.....	safety guard closed	3
4	VTA.....	safety guard closed	4
5	DIRECT.....	safety guard closed	5
6	FCS.....	safety guard closed	6
7	INTLK1.....	OFF, safety guard open	7
8	INTLK2.....	OFF, safety guard open	8
9	AUX BAT	ON	9
10	DC1.....	ON	10
11	DC2.....	ON	11
12	LV	ON	12
13	HVL.....	ON	13
14	Velocity Mode.....	CHECK armed	14
15	Flaps	CHECK retracted	15
16	Climb Stick Centered.....	CHECK	16
17	Flight Controls.....	CHECK	17

End of Checklist

ENGINE START PROCEDURE

If Take-Off Area Not Clear:

HT HOVER DISABLE.....ON

FCS open safety guard, PUSH

BIT status passed CHECK INDICATION

AFTER ENGINE START CHECK

1	LTU RPM 100 +/- 30	CHECK	1
2	Traction RPM.....	CHECK as required	2
3	Primary Control Surfaces Neutral	CHECK	3
4	Flaps	CHECK extended	4

End of Checklist

Figure G.1: TUM-FSD Simulator Pilot Checklist Page 1.

TUM-FSD eVTOL Simulator	NORMAL PROCEDURES
--------------------------------	--------------------------

BEFORE TAKE-OFF CHECK

- | | | | |
|---|------------------------------|--|---|
| 1 | Throttle Lever..... IDLE | | 1 |
| 2 | FCS..... safety guard closed | | 2 |

End of Checklist

FCS ACTIVATION PROCEDURE

- INTLK1..... ON, close safety guard*
INTLK2..... ON, close safety guard
FCS Disengage Unavailable..... CHECK INDICATION
Climb Stick Command FULL FORWARD
Velocity ModeCHECK engaged

Take-Off Procedure

VERTICAL CLIMB CHECK

- | | | | |
|---|---|--|---|
| 1 | HT HOVER DISABLE OFF | | 1 |
| 2 | Velocity ModeCHECK engaged | | 2 |
| 3 | Attitude/Acceleration Mode..... CHECK armed | | 3 |

End of Checklist

TRANSITION PROCEDURE

- Throttle Lever..... AT RIGHT GATE*

Airspeed above V_S :

Throttle Lever..... AT LEFT GATE

AFTER TRANSITION CHECK

- | | | | |
|---|------------------------------------|--|---|
| 1 | LTUs..... CHECK disengaged | | 1 |
| 2 | FlapsCHECK retracting | | 2 |
| 3 | Throttle Lever..... SET as desired | | 3 |

End of Checklist

Figure G.2: TUM-FSD Simulator Pilot Checklist Page 2.

DESCENT/APPROACH CHECK

1	VTA.....	ON, safety guard closed	1
2	HTA.....	ON, safety guard closed	2
3	HT HOVER DISABLE	OFF	3
4	Velocity Mode Engaged	CHECK	4
5	Reference Velocity Thrust.....	SET	5

End of Checklist

RETRANSITION PROCEDURE

Throttle Lever..... AT LEFT GATE

Airspeed below $V_{LS,NE}$

Throttle Lever..... AT RIGHT GATE

AFTER RETRANSITION CHECK

1	LTUs.....	CHECK engaged	1
2	Flaps	CHECK extended	2
3	Throttle Lever	SET as required	3
4	Climb Stick Centered.....	CHECK	4

End of Checklist

GO AROUND PROCEDURE

Throttle Lever..... AT RIGHT GATE

At safe altitude and horizontal flight:

Start Transition Procedure

Throttle Lever..... AT LEFT GATE

Perform After Transition Check

Continue with take-off profile

Figure G.3: TUM-FSD Simulator Pilot Checklist Page 3.

TUM-FSD eVTOL Simulator **NORMAL PROCEDURES**

VERTICAL DESCENT CHECK

1 Landing Area ClearCHECK 1

If Landing Area Not Clear:

HT HOVER DISABLE ON

End of Checklist

SHUTDOWN PROCEDURE

Climb Stick Command FULL FORWARD
 INTLK1 open safety guard, OFF
 INTLK2 open safety guard, OFF
 FCS Disengage Available CHECK INDICATION
 FCS open safety guard, PUSH
 FCS CHECK disengaged

AFTER LANDING CHECK

1	FlapsCHECK retracted	1
2	HTA safety guard closed	2
3	VTA safety guard closed	3
4	DIRECT safety guard closed	4
5	FCS safety guard closed	5
6	HVL OFF	6
7	Discharge PUSH, check discharged	7
8	LV OFF	8
9	DC1 OFF	9
10	DC2 OFF	10
11	AUX BAT OFF	11
12	HT HOVER DISABLE OFF	12

End of Checklist

Figure G.4: TUM-FSD Simulator Pilot Checklist Page 4.

EMERGENCY + ABNORMAL CHECKLIST

WARNINGS.....	2
HPU PROP FAIL	3
VPU PROP FAIL.....	3
BAT FAIL.....	4
MON FAIL.....	4
RADAR ALT FAIL	4
FIXED-WING LANDING CONFIGURATION CHECK.....	5
FIXED-WING FINAL CHECK.....	5
POWERED-LIFT LANDING CONFIGURATION CHECK.....	6

Abnormal Checklist starts at page 7

E

Figure G.5: TUM-FSD Simulator Pilot Checklist Page 7.


 TUM-FSD eVTOL Simulator		EMERGENCY PROCEDURES	
WARNINGS			
HPU PROP FAIL	Pg. 3	Traction Engine Failure (one or all)	
VPU PROP FAIL	Pg. 3	Vertical Engine Failure (one or many)	
BAT FAIL	Pg. 4	Battery Unit Failure	
MON FAIL	Pg. 4	Velocity Mode Failure Detected	
RADAR ALT FAIL	Pg. 4	Radar Altimeter Failure	
06.09.2022 <i>Edition # 1.0</i>		Property of TUM-FSD	
		Page 2	

Figure G.6: TUM-FSD Simulator Pilot Checklist Page 8.



HPU PROP FAIL

TRACTION ENGINE FAILURE (ONE OR ALL)


- Check number of failed traction engines
 - ❖ → For one failure
 - ⇒ In all modes, except Velocity, only gradually change the throttle settings
 - ❖ → For complete traction system failure
 - ⇒ In all powered-lift modes, be prepared that forward and backward acceleration is achieved via pitch

VPU PROP FAIL

VERTICAL ENGINE FAILURE (ONE OR MANY)

- Be prepared for **RETRANS ACT** caution during transition with the Velocity Mode
- For LTUs in motion, ($V_{LS,NE}$) must not be exceeded
- Check number of failed engines
 - ❖ → For one failure
 - ⇒ If the failed motor has non-zero RPM, be prepared for **TRANS ACT** caution during retransition with the Velocity Mode
 - ⇒ Be prepared for performance losses in powered-lift flight
 - ❖ → For multiple failures
 - ⇒ Perform **FIXED-WING LANDING CONFIGURATION** immediately
 - ⇒ *Procedure found on page 5*
 - ⇒ Increase airspeed
 - ⇒ Be prepared to execute a fixed-wing landing

Figure G.7: TUM-FSD Simulator Pilot Checklist Page 9.

 **TUM-FSD eVTOL Simulator**
EMERGENCY PROCEDURES

BAT FAIL

BATTERY UNIT FAILURE

- Be prepared for performance losses in powered-lift flight
- Be prepared for **RETRANS ACT** caution during transition with the Velocity Mode

MON FAIL

VELOCITY MODE FAILURE DETECTED

- Be prepared to execute transition and retransition procedures in ATT/ACC Mode. *Procedures found on pages 13 and 14*

RADAR ALT FAIL

RADAR ALTIMETER FAILURE

- Velocity Mode:
 - ⇒ Altitude Hold is unavailable
 - ⇒ Altitude Protection is unavailable
 - ⇒ Hover sink rate restriction near vertiport is unavailable
 - ⇒ Attitude Restrictions as if near vertiport always in effect

06.09.2022
Edition # 1.0

Property of TUM-FSD

Page 4

Figure G.8: TUM-FSD Simulator Pilot Checklist Page 10.

FIXED-WING LANDING CONFIGURATION CHECK

Setting must only be performed in fixed-wing flight.

- 1 VTA..... open safety guard, OFF 1
- 2 VTA..... close safety guard 2

If Velocity Mode engaged:

Height Protection CHECK disengaged
 Underspeed Protection..... CHECK disengaged

FIXED-WING FINAL CHECK

- 1 Reference Velocity ThrustSET 1
- If Velocity Mode not engaged:
- FlapsSET full travel LDG
- 2 Flaps.....CHECK extended 2
- 3 Climb Stick Centered CHECK 3

FIXED-WING GO AROUND PROCEDURE

Throttle Lever..... AT RIGHT GATE

Positive Rate of climb and at safe altitude:

Throttle Lever..... AT LEFT GATE

If Velocity Mode not engaged and Airspeed below V_{FE} :

FlapsSET full travel UP

Flaps.....CHECK retracted

Continue with take-off profile

SHUTDOWN PROCEDURE

Throttle Lever..... IDLE

INTLK1..... open safety guard, OFF

INTLK2..... open safety guard, OFF

FCS Disengage Available CHECK INDICATION

FCS open safety guard, PUSH

FCS CHECK disengaged

Figure G.9: TUM-FSD Simulator Pilot Checklist Page 11.

TUM-FSD eVTOL Simulator **EMERGENCY PROCEDURES**

POWERED-LIFT LANDING CONFIGURATION CHECK

Setting must only be performed if a fixed-wing take-off was executed.

1	VTA..... open safety guard, ON	1
2	VTA..... close safety guard	2

06.09.2022
Edition # 1.0

Property of TUM-FSD

Page 6

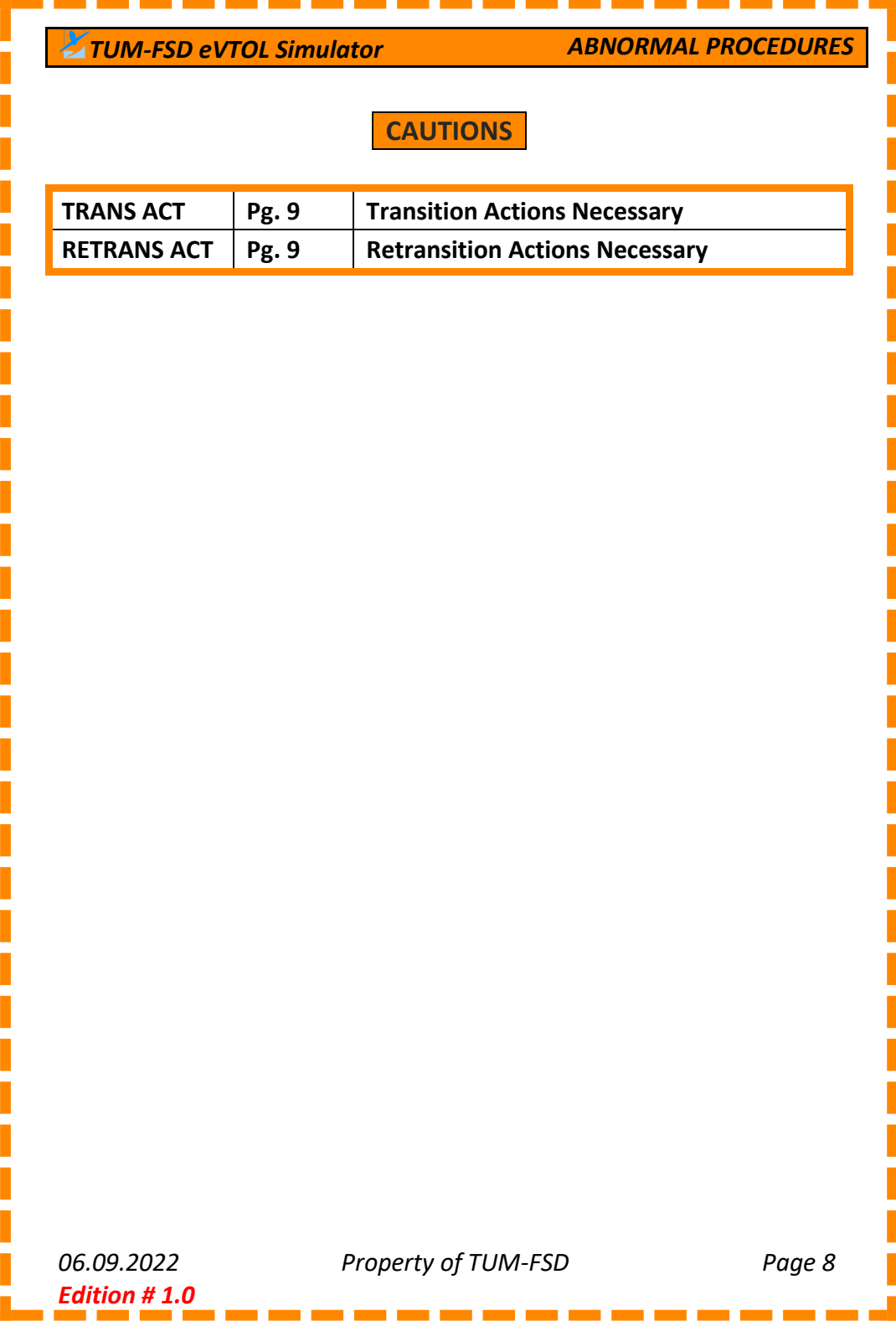
Figure G.10: TUM-FSD Simulator Pilot Checklist Page 12.

ABNORMAL CHECKLIST

CAUTIONS	8
TRANS ACT	9
RETRANS ACT	9
<u>Velocity Mode:</u>	
ABNORMAL EVENT DURING VELOCITY MODE TRANSITION	10
ABNORMAL EVENT DURING VELOCITY MODE RETRANSITION	11
<u>ATT/VS Mode:</u>	
TRANSITION ATT/VS MODE	12
RETRANSITION ATT/VS MODE	12
<u>ATT/ACC Mode:</u>	
TRANSITION ATT/ACC MODE	13
RETRANSITION ATT/ACC MODE	14
<u>RATE and DIRECT Modes:</u>	
TRANSITION RATE AND DIRECT MODE	15
RETRANSITION RATE AND DIRECT MODE	16
<u>Other Abnormal Scenarios:</u>	
FIXED-WING BEFORE ENGINE START CHECK	17
FIXED-WING AFTER ENGINE START CHECK	17
FIXED-WING BEFORE TAKE-OFF CHECK	17
FIXED-WING AFTER TAKE-OFF CHECK	17

A

Figure G.11: TUM-FSD Simulator Pilot Checklist Page 13.



TUM-FSD eVTOL Simulator **ABNORMAL PROCEDURES**

CAUTIONS

TRANS ACT	Pg. 9	Transition Actions Necessary
RETRANS ACT	Pg. 9	Retransition Actions Necessary

06.09.2022
Edition # 1.0

Property of TUM-FSD

Page 8

Figure G.12: TUM-FSD Simulator Pilot Checklist Page 14.

TRANS ACT

TRANSITION ACTIONS NECESSARY

- Actions depend on the currently engaged mode
 - ❖ → If the Velocity Mode is engaged
 - ⇒ The powered-lift system shutdown did not complete in the given timeframe
 - ⇒ Execute the **ABNORMAL EVENT DURING VELOCITY MODE TRANSITION PROCEDURE**
 - ⇒ *Procedure found on page 10*
 - ❖ → For any other engaged Mode
 - ⇒ Ensure complete powered-lift system shutdown prior to accelerating to higher airspeeds
 - ⇒ For LTUs in motion, ($V_{LS,NE}$) must not be exceeded

RETRANS ACT

RETRANSITION ACTIONS NECESSARY

- Actions depend on the currently engaged mode
 - ❖ → If the Velocity Mode is engaged
 - ⇒ The powered-lift system did not completely turn on
 - ⇒ Execute the **ABNORMAL EVENT DURING VELOCITY MODE RETRANSITION PROCEDURE**
 - ⇒ *Procedure found on page 11*
 - ❖ → For any other engaged Mode
 - ⇒ Ensure complete powered-lift system is completely turned on prior engaging the Hover Mode

Figure G.13: TUM-FSD Simulator Pilot Checklist Page 15.


 TUM-FSD eVTOL Simulator		ABNORMAL PROCEDURES	
ABNORMAL EVENT DURING VELOCITY MODE TRANSITION			
Reason for warning must be ascertained			
❖	→	If reversion to Powered-Lift Flight necessary	
		Perform Normal Retransition Procedure when able:	
	1	Throttle LeverAT RIGHT GATE	1
	2	LTUs CHECK engaged	2
❖	→	Otherwise (continue to Fixed-Wing Flight)	
		Confirm Transition:	
	1	Throttle Lever AT LEFT GATE	1
	2	Gate OverridePUSH	2
	3	Flaps.....CHECK retracting	3
		Perform Normal Retransition Procedure when able	
06.09.2022		Property of TUM-FSD	
Edition # 1.0		Page 10	

Figure G.14: TUM-FSD Simulator Pilot Checklist Page 16.

ABNORMAL EVENT DURING VELOCITY MODE RETRANSITION

Reason for warning must be ascertained


- ❖ → If **MULTIPLE** LTUs failed
 - Perform Normal Transition Procedure when able:

1	Throttle Lever	AT LEFT GATE	1
2	LTUs	CHECK disengaged	2
 - Perform go-around procedure
 - Perform **FIXED-WING LANDING CONFIGURATION**
 - Procedure found on page 5*
 - Perform fixed-wing landing
- ❖ → If **ONE** LTU failed and go-around is desired
 - Perform Normal Transition Procedure when able:

1	Throttle Lever	AT LEFT GATE	1
2	LTUs	CHECK disengaged	2
- ❖ → Otherwise (continue to Powered-Lift Flight)
 - Confirm Retransition:

1	Throttle Lever	AT RIGHT GATE	1
2	Gate Override	PUSH, CONTINUE PUSHING	2
3	Throttle Level.....	CROSS BELOW GATE	3
4	Gate Override	RELEASE	4
5	LTUs	CHECK engaged	5

Figure G.15: TUM-FSD Simulator Pilot Checklist Page 17.

 TUM-FSD eVTOL Simulator
ABNORMAL PROCEDURES

TRANSITION ATT/VVS MODE

Transition in ATT/VVS Mode not advised

Revert to ATT/ACC Mode

1	ATT/ACC Mode	ENGAGE	1
2	ATT/ACC Mode	CHECK Engaged	2

Perform abnormal scenario **TRANSITION ATT/ACC MODE**
Procedure found on page 13

RETRANSITION ATT/VVS MODE

Retransition in ATT/VVS Mode not advised

Revert to ATT/ACC Mode

1	ATT/ACC Mode	ENGAGE	1
2	ATT/ACC Mode	CHECK Engaged	2

Perform abnormal scenario **RETRANSITION ATT/ACC MODE**
Procedure found on page 14

06.09.2022
Edition # 1.0
Property of TUM-FSD
Page 12

Figure G.16: TUM-FSD Simulator Pilot Checklist Page 18.

TRANSITION ATT/ACC MODE

For LTUs in motion, ($V_{LS,NE}$) must not be exceeded

Perform acceleration to V_{FTO}

- 1 Throttle LeverAT RIGHT GATE 1

Airspeed above V_S :

- 2 Throttle Lever AT LEFT GATE 2
- 3 Aircraft not descending CHECK 3
- 4 LTUs RPM LESS THAN 500 CHECK INDICATION 4
- 5 Gate OverridePUSH 5
- 6 LTU RPM 0 +/- 5 CHECK INDICATION 6

❖ → If reversion to Powered-Lift Flight necessary

- 7 Transition Warning DISMISS 7

Perform **Retransition Procedure** when able

Procedure found on page 14

❖ → Otherwise (continue to Fixed-Wing Flight)

- 7 Flaps..... SET full travel UP 7
- 8 Transition Warning DISMISS 8

Figure G.17: TUM-FSD Simulator Pilot Checklist Page 19.

TUM-FSD eVTOL Simulator		ABNORMAL PROCEDURES	
RETRANSITION ATT/ACC MODE			
For LTUs in motion, ($V_{LS,NE}$) must not be exceeded			
Perform deceleration to V_{REF}			
1	Throttle Lever	AT LEFT GATE	1
Airspeed below V_{FE}:			
2	Flaps.....	SET full travel LDG	2
Airspeed below $V_{LS,NE}$:			
3	Throttle Lever	AT RIGHT GATE	3
4	Gate Override	PUSH, CONTINUE PUSHING	4
5	LTU RPM 100 +/- 30.....	CHECK INDICATION	5
❖	→ If MULTIPLE LTUs failed		
6	Gate Override	RELEASE	6
Perform go-around procedure			
Perform FIXED-WING LANDING CONFIGURATION			
<i>Procedure found on page 5</i>			
Perform fixed-wing landing			
❖	→ If ONE LTU failed and go-around is desired		
6	Gate Override	RELEASE	6
Perform go-around procedure and reattempt			
❖	→ Otherwise continue to Powered-Lift desired		
6	Throttle Level.....	CROSS BELOW GATE	6
7	Gate Override	RELEASE	7
8	LTUs	CHECK engaged	8

06.09.2022 Property of TUM-FSD Page 14
Edition # 1.0

Figure G.18: TUM-FSD Simulator Pilot Checklist Page 20.

TRANSITION RATE AND DIRECT MODE

Retransition in ATT/VIS Mode not advised.

1 ATT/ACC ModeCHECK available 1

❖ → If **ATT/ACC Mode Available**

Revert to ATT/ACC Mode

2 ATT/ACC Mode ENGAGE 2

3 ATT/ACC Mode CHECK engaged 3

Perform **TRANSITION ATT/ACC MODE**

Procedure found on page 13

❖ → Otherwise

Perform acceleration to V_{FTO}

2 Coolie Hat Throttle SettingAT RIGHT GATE 2

Airspeed above V_S :

3 Coolie Hat Throttle Setting AT LEFT GATE 3

4 Aircraft not descending CHECK 4

5 Gate OverridePUSH 5

6 LTU RPM 0 +/- 5 CHECK INDICATION 6

7 Throttle Lever AT LEFT GATE 7

8 Flaps..... SET full travel UP 8

9 Transition Warning DISMISS 9

Figure G.19: TUM-FSD Simulator Pilot Checklist Page 21.

TUM-FSD eVTOL Simulator		ABNORMAL PROCEDURES	
RETRANSITION RATE AND DIRECT MODE			
Retransition in ATT/VS Mode not advised.			
1	ATT/ACC Mode	CHECK available	1
❖	→ If ATT/ACC Mode Available		
	Revert to ATT/ACC Mode		
2	ATT/ACC Mode	ENGAGE	2
3	ATT/ACC Mode	CHECK engaged	3
	Perform RETRANSITION ATT/ACC MODE		
	<i>Procedure found on page 14</i>		
❖	→ Otherwise		
	Perform FIXED-WING LANDING CONFIGURATION		
	<i>Procedure found on page 5</i>		
	Perform fixed-wing landing		
06.09.2022		Property of TUM-FSD	Page 16
Edition # 1.0			

Figure G.20: TUM-FSD Simulator Pilot Checklist Page 22.

FIXED-WING BEFORE ENGINE START CHECK

Setting must only be performed on ground with FCS Disengaged.

Perform **BEFORE ENGINE START CHECK**

- 1 VTA..... open safety guard, OFF 1
- 2 VTA..... safety guard closed 2

FIXED-WING ENGINE START PROCEDURE

FCS open safety guard, PUSH
 BIT status passed CHECK INDICATION

FIXED-WING AFTER ENGINE START CHECK

- 1 Traction RPM 100 +/-30 CHECK 1
- 2 LTU RPM 0 CHECK 2
- 3 Primary Control Surfaces Neutral..... CHECK 3
- 4 Flaps.....CHECK extended 4

FIXED-WING BEFORE TAKE-OFF CHECK

- 1 Throttle LeverIDLE 1
- 2 FCS safety guard closed 2

FIXED-WING FCS ACTIVATION PROCEDURE

INTLK1..... ON, close safety guard
 INTLK2..... ON, close safety guard
 FCS Disengage Unavailable..... CHECK INDICATION
 Throttle Lever..... FULL BACKWARD
 Velocity Mode Engaged CHECK

Perform Wingborne Take-Off Procedure

FIXED-WING AFTER TAKE-OFF CHECK

If Velocity Mode not engaged at safe altitude reached:

FlapsSET full travel UP

Figure G.21: TUM-FSD Simulator Pilot Checklist Page 23.

Appendix H

Additional Simulation Results

This appendix demonstrates additional results that validate the off-nominal transition procedures. These are in effect due to the impossibility of an LTU to disengage via the automation. The reasons where this may occur are listed in Section 2.3.1 of Chapter 2. According to Section 4.4.3 of Chapter 4, on one hand, the procedures are to revert back to powered-lift flight. This is examined in Section H.1.

Alternatively, the operator could shut down the problematic LTU manually. For the exemplary aircraft found in Section 1.1.4 this is not possible, therefore results for such a scenario are not provided. However, both Chapters 3 and 4 demonstrate no design modification regardless of the possibility of the pilot to manually disengage an LTU.

Lastly, prolonged flight at high speed may be demanded. In this case the aerodynamic efficiency needs to be increased. Keeping the already disengaged LTUs aids in this goal. In addition, the high-lift system can be retracted accordingly to further minimize the drag. This scenario is examined in Section H.2.

H.1 Transition Mitigation - Reversion to Powered-Lift Flight

The procedure for powered-lift flight reversion is defined as in Table 4.10 of Chapter 4. The results of this scenario while performing a flight with the Nominal system are visible in Figure H.1. They are elaborated upon below.

Firstly, as previously stated, this procedure occurs during transition. Therefore, the initial sequence is the same as found in Section 6.1.1 of Chapter 6. For this reason, the initial aircraft response is in Figure 6.11 is equivalent to the one in Figure 6.2.

As discussed in Section 6.1.1, the disengagement of the powered-lift system begins once requested by the operation (via $trans_{rqst}$), the disengagement speed has been exceeded (via V_{trans}) and the LTUs are not actively used for force and moment production (via LTU_{UNUSED}). Therefore, this sequence matches the one found in Figure 6.2 and is not elaborated further here.

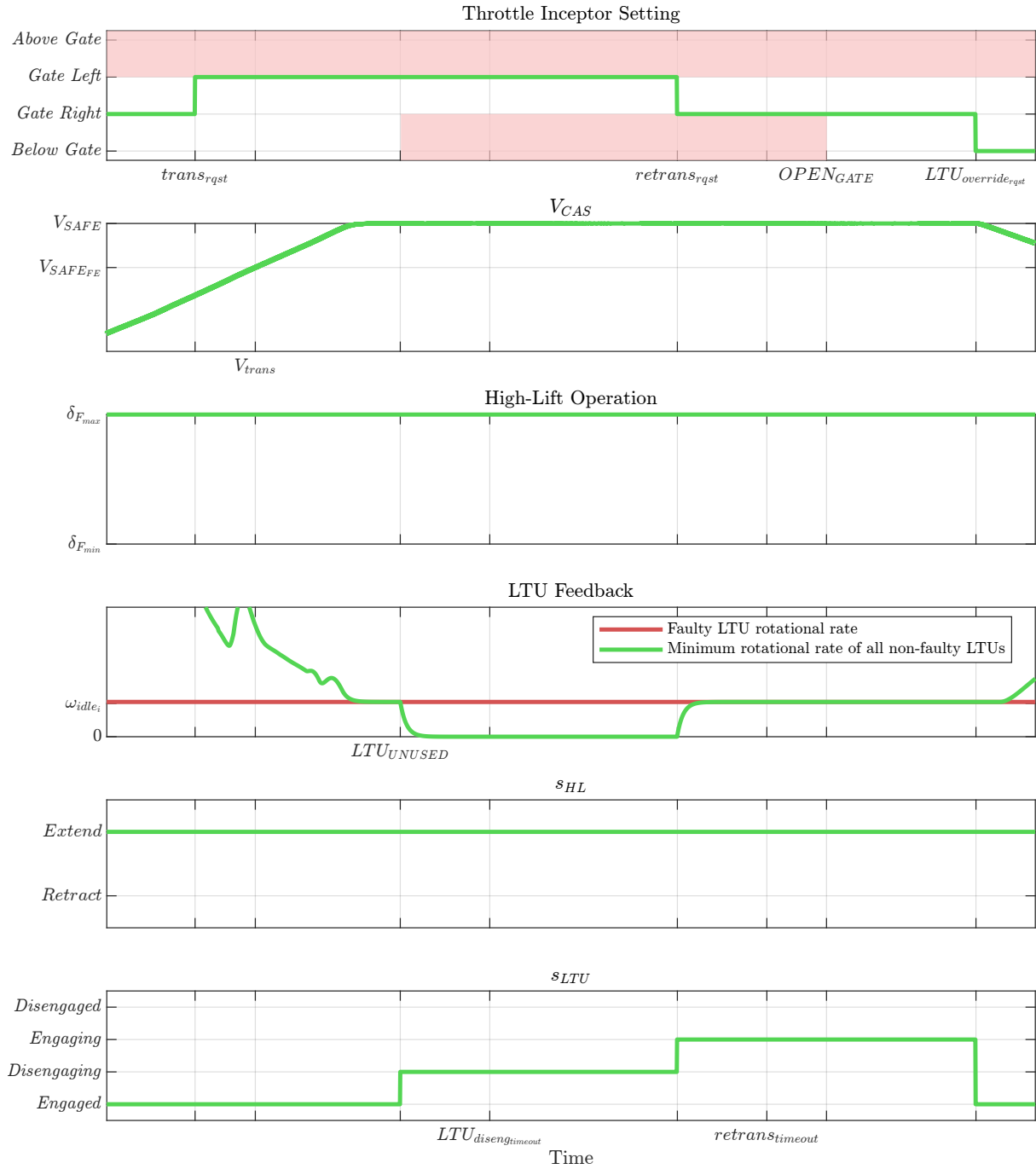


Figure H.1: Nominal System Transition Mitigation to Powered-Lift Flight

For the purposes of this observation, a “stuck at value” failure is injected in one LTU such that it is rotating at idle RPM. The driving factors for this failure mode are discussed at length in Section 2.3.1 of Chapter 2. The fourth graph of Figure 6.2 shows the response of LTUs, where the time history of the red depicts the erroneous LTU response and the one in green shows the minimum RPM of all non-faults units for the sake of transparency.

Note that if this failure is detected by the LTU integrity monitoring, a warning could be commissioned to the operator via the cautions and warning system. However, this would just raise the operator awareness of the upcoming off-nominal transition process. The behavior of the automation would not be changed.

Upon starting the disengagement process marked with the state transition of M_{LTU} to *Disengaging*, the lower barrier is closed in accordance with Table 4.7. This is depicted in the first graph of Figure 6.2. The red sections depict that the corresponding barrier is closed and therefore the operator is incapable of deflecting the inceptor into that region.

Once the disengagement has been initiated, a counter starts running as per Equation 3.44. In this example, the counter is implemented as depicted in Figure 5.14. Due to the failure, the variable $LTU_{disengtimeout}$ is evaluated to *true* after the timer has elapsed. This throws a warning to the pilot that additional actions are necessary in accordance with Table 3.7. The moment this is in effect can be seen on the lowest graph of Figure 6.2.

According to Table 4.10, the pilot needs to deflect the throttle inceptor into the powered-lift region in order to initiate the LTU engagement process. This is processed by the automation via the $trans_{rqst}$ variable, calculated as per Equation 3.4 and implemented as per Figure 5.7. It is visible in the up-most graph of Figure 6.2.

The $trans_{rqst}$ toggles the transition of M_{LTU} to *Engaging* in accordance with Equation 3.27. This transition function is implemented as depicted in Figure 5.12. Thereupon, the retransition procedure is initiated. This procedure was elaborated upon in detail in Section 4.4.3.3 of Chapter 4 and the simulation results were discussed at length in Section 6.3.2 of Chapter 6.

Figure H.2 demonstrates the management of the airspeed protections during this off-nominal scenario as proof that a safe state is always enforced by the system by design. The management of the functions is as per Tables 3.5 and 3.6 for underspeed and overspeed respectively.

Simulation results for the Fallback system in this scenario are not provided because prior to the mitigation, the transition is identical to the one presented in Section 6.2. Subsequently, the conduction of the retransition is the same as presented in the above-mentioned section. Section 4.4.3 discusses at length why this is the case.

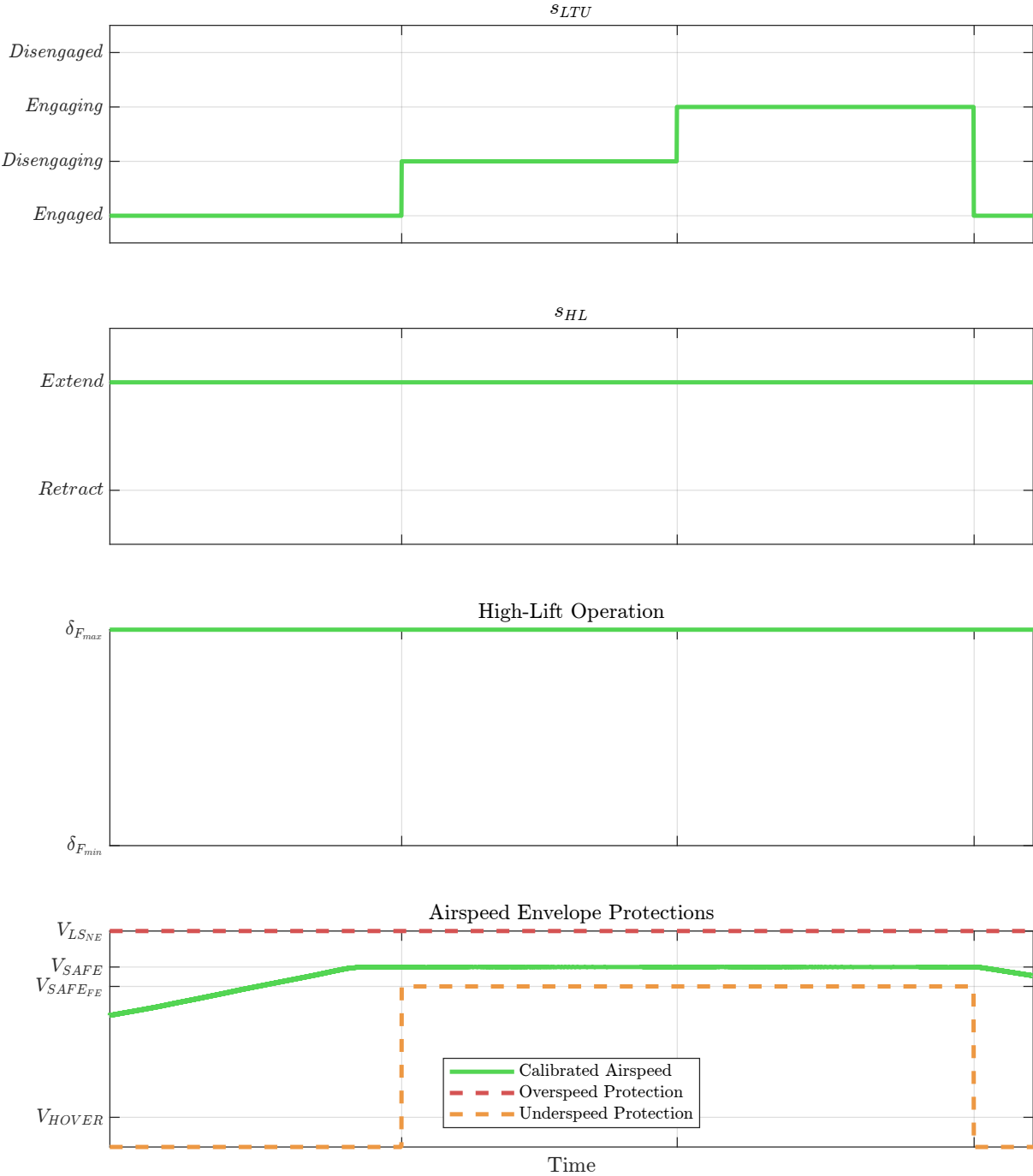


Figure H.2: Nominal System Airspeed Protection Operation During Transition Mitigation to Powered-Lift Flight

H.2 Transition - Prolonged High-Speed Flight

The other scenario to examine in the case where an LTU is incapable of disengaging would require prolonged high-speed flight. This could be the case when the mission needs to be continued despite the impossibility to fully disengage the powered-lift system. The procedure for this scenario is in accordance with Table 4.12. The simulation results for this capability are examined here and can be obtained from Figure H.3.

The results are identical to the ones found in the previous section until the moment where the warning via $LTU_{disengtimeout}$ as per Equation 3.44. The two time histories diverge due to the different operator decision.

To confirm the entry into high-speed flight as per Table 4.12, the item $OPEN_{GATE}$ needs to be engaged. This is visible in Equation 4.33, in which this variable is linked also to $HL_{override_{rqst}}$. The moment this action is performed by the operator is visible in the fifth graph of Figure H.3.

The variable $HL_{override_{rqst}}$ assignment is performed by the behavioral specification as visible in Figure 5.14a. As per Equation 3.50, this enforces the transition condition of M_{HL} to the state *Retract*, which is implemented as depicted in Figure 5.15. Therefore, the high-lift scheduling is done such that the drag can be minimized. The scheduling following the state transition is provided in the third graph of Figure H.3. For the exemplary aircraft found in Section 1.1.4 this minimizes the drag by approximately 35%.

Figure H.4 demonstrates the management of the airspeed protections during this off-nominal scenario as proof that a safe state is always enforced by the system by design. The management of the functions is as per Tables 3.5 and 3.6 for underspeed and overspeed respectively. Following the state transition, the upper barrier is opened in accordance with Table 4.7. This allows the command of higher airspeeds by the pilot. However, as visible in Figure H.4, due to the running LTU, a speed beyond V_{LSNE} cannot be reached.

Simulation results for the Fallback system in this scenario are not provided because prior to the mitigation, the transition is identical to the one presented in Section 6.2. Section 4.4.3 discusses at length why this is the case. Following the transition, the operator's responsibility is to not exceed the structural limit speed V_{LSNE} .

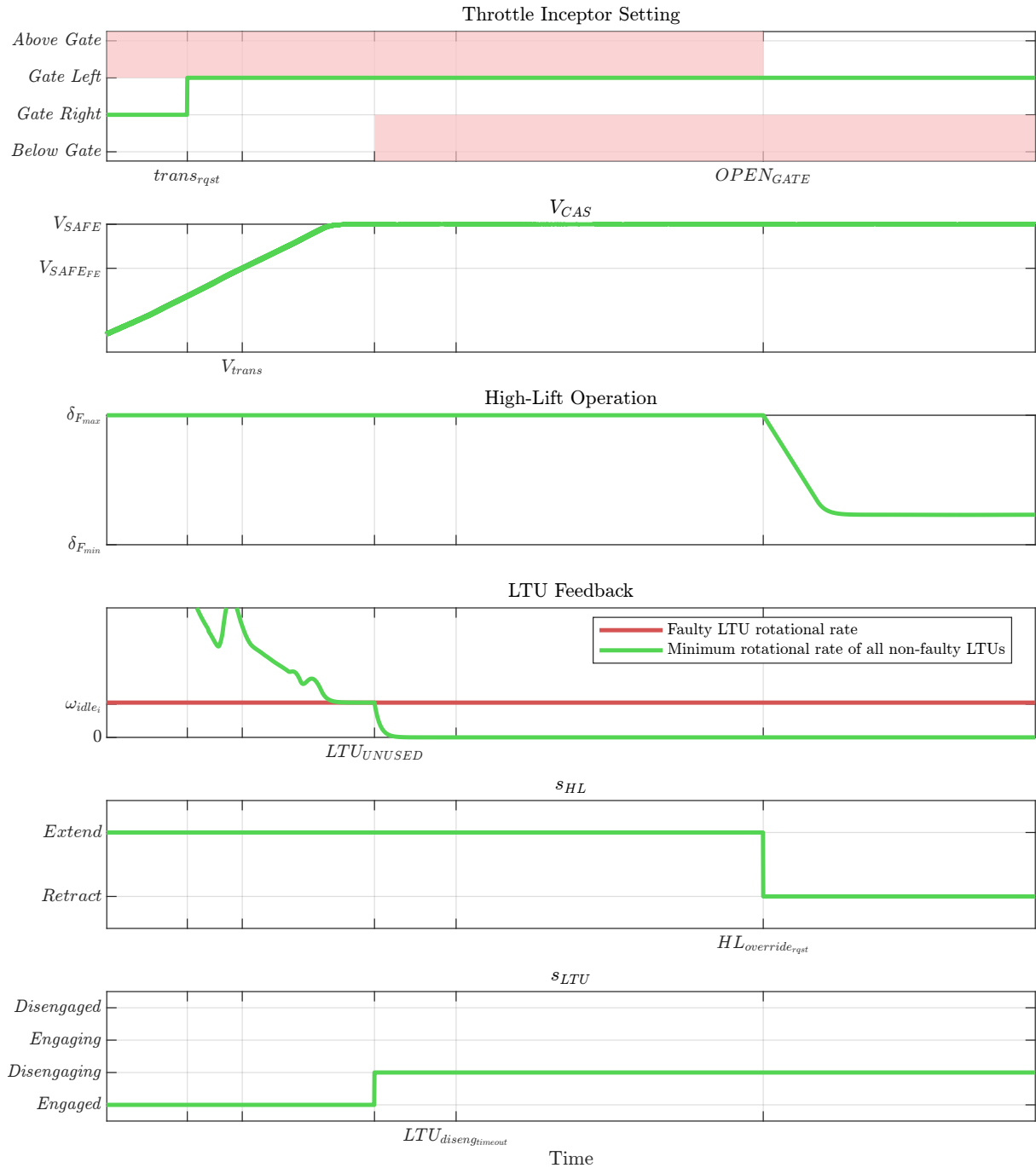


Figure H.3: Nominal System Abnormal Entry into High Dynamic Pressures with a Partially Disengaged Powered-Lift System

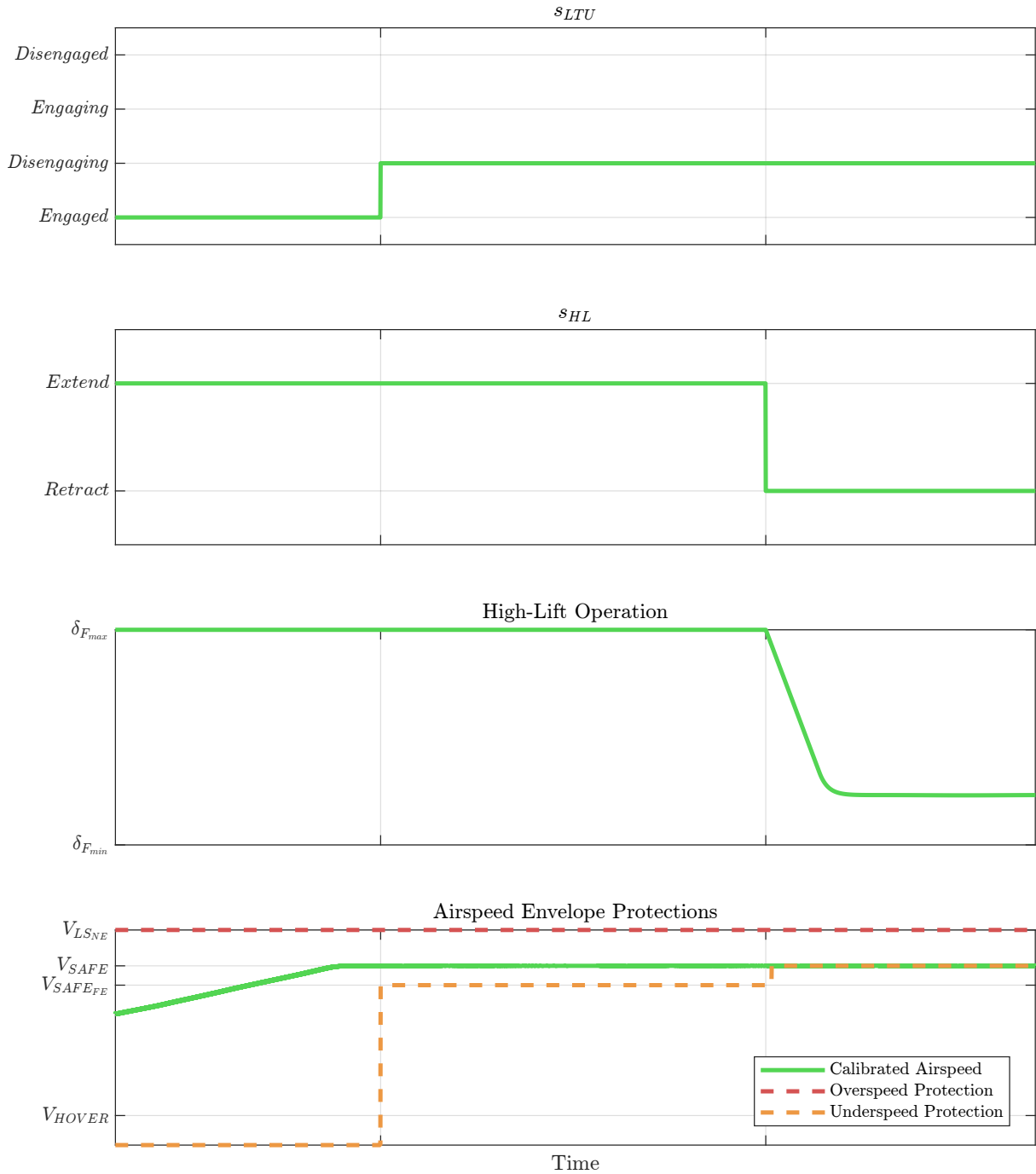


Figure H.4: Nominal System Airspeed Protection Management During Abnormal Entry into High Dynamic Pressures with a Partially Disengaged Powered-Lift System

Bibliography

- [1] Flugwerft Schleissheim. Otto Doppeldecker - Flugwerft Schleissheim. [Online]. Available: <https://www.deutsches-museum.de/en/flugwerft-schleissheim/exhibition/military-aviation/otto-doppeldecker> (Accessed 29.06.2022).
- [2] Technik Museum Speyer. Douglas DC-3 | Technik Museum Speyer | Germany. [Online]. Available: <https://speyer.technik-museum.de/en/douglas-dc-3> (Accessed 29.06.2022).
- [3] Munich Airport. Presse: Vier weitere Airbus A350 am Münchner Airport. [Online]. Available: <https://www.munich-airport.de/presse-vier-weitere-airbus-a350-am-muenchner-airport-12523087> (Accessed 29.06.2022).
- [4] Volocopter. VoloCity. [Online]. Available: <https://www.volocopter.com/solutions/velocity/> (Accessed 01.07.2022).
- [5] The Vertical Flight Society. The Electric VTOL News™. Airbus CityAirbus. [Online]. Available: <https://evtol.news/airbus-helicopters/> (Accessed 01.07.2022).
- [6] Joby Aviation. Joby Aviation | Joby. [Online]. Available: <https://www.jobyaviation.com/> (Accessed 01.07.2022).
- [7] Volocopter. VoloConnect. [Online]. Available: <https://www.volocopter.com/solutions/voloconnect/> (Accessed 01.07.2022).
- [8] J. Angelov, “Model-Based Systems Engineering of Flight Control for VTOL Transition Aircraft,” Dissertation, Technical University of Munich, Munich, 2023.
- [9] D. Dollinger, P. Reiss, J. Angelov, D. Löbl, and F. Holzapfel, “Control Inceptor Design for Onboard Piloted Transition VTOL Aircraft Considering Simplified Vehicle Operation,” in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-1896
- [10] European Aviation Safety Agency, “Proposed Means of Compliance with the Special Condition VTOL, Doc. No: MOC-2 SC-VTOL,” 23.06.2021.

- [11] M. W. Groll and D. Heber, “E/E-Product Data Management in Consideration of Model-Based Systems Engineering,” in *Transdisciplinary engineering*, ser. Advances in transdisciplinary engineering. IOS Press, 2016. doi: 10.3233/978-1-61499-703-0-289
- [12] J. A. Stoop and J. P. Kahan, “Flying is the safest way to travel,” *European Journal of Transport and Infrastructure Research*, vol. 5, no. 2, 2005. doi: 10.18757/E-JTIR.2005.5.2.4392
- [13] Electric VTOL News. Volocopter VC1 (defunct prototype). [Online]. Available: <https://evtol.news/volocopter-vc1-vc2/> (Accessed 25.07.2022).
- [14] J. Holden and N. Goel, *Fast-Forwarding to a Future of On-Demand Urban Air Transportation*. San Francisco, CA: Uber Inc., 2016.
- [15] Fortune Business Insights, “Urban Air Mobility Market Size, Share and Growth Report [2028],” Report ID.: FBI106344, February 2022.
- [16] Emergen Research, “Urban Air Mobility Market By Component (Platform, Infrastructure), By Operation (Piloted, Hybrid, Fully Autonomous), By Range (Inter-city, Intracity), By End-Use, and By Region Forecast to 2030,” Report ID.: MKMK16535188, April 2022.
- [17] A. Sikora, “European Green Deal – legal and financial challenges of the climate change,” *ERA Forum*, vol. 21, no. 4, 2021. doi: 10.1007/s12027-020-00637-3
- [18] R. H. Bezdek, “The Jobs Impact of the USA New Green Deal,” *American Journal of Industrial and Business Management*, vol. 10, no. 6, 2020. doi: 10.4236/ajibm.2020.106072
- [19] T. Lombaerts, J. Kaneshige, and M. Feary, “Control Concepts for Simplified Vehicle Operations of a Quadrotor eVTOL Vehicle,” in *AIAA AVIATION 2020 FORUM*. AIAA, 2020. doi: 10.2514/6.2020-3189
- [20] M. Feary, “A First Look at the Evolution of Flight Crew Requirements for Emerging Market Aircraft,” *NASA*, 2018.
- [21] S. J. Kapurch, *NASA Systems Engineering Handbook*. DIANE Publishing, 2010. ISBN 9781437937305
- [22] A. P. Schulz, D. P. Clausing, H. Negele, and E. Fricke, “Shifting the View in Systems Development: Technology Development at the Fuzzy Front End As a Key to Success,” in *11th International conference on design theory and methodology*. American Society of Mechanical Engineers, 1999. doi: 10.1115/DETC99/DTM-8773

- [23] Youngshin Kang, Bumjin Park, Changsun Yoo, Yushin Kim, and Samok Koo, "Flight test results of automatic tilt control for small scaled tilt rotor aircraft," in *2008 International Conference on Control, Automation and Systems*. IEEE, 2008. doi: 10.1109/iccas.2008.4694527
- [24] L. Zhong, H. Yuqing, Y. Liying, and H. Jianda, "Control techniques of tilt rotor unmanned aerial vehicle systems: A review," *Chinese Journal of Aeronautics*, vol. 30, no. 1, 2017. doi: 10.1016/j.cja.2016.11.001
- [25] H. Yang and R. Morales, "Robust Full-Envelope Flight Control Design for an eVTOL Vehicle," in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-0254
- [26] P. Casau, D. Cabecinhas, and C. Silvestre, "Autonomous Transition Flight for a Vertical Take-Off and Landing aircraft," in *IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011. doi: 10.1109/cdc.2011.6160819
- [27] Z. Zaludin and A. S. M. Harituddin, "Challenges and Trends of Changing from Hover to Forward Flight for a Converted Hybrid Fixed Wing VTOL UAS from Automatic Flight Control System Perspective," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2019. doi: 10.1109/icsengt.2019.8906483
- [28] B. Yüksel and S. Mores, "METHOD OF CONTROLLING AN AIRCRAFT, FLIGHT CONTROL DEVICE FOR AN AIRCRAFT, AND AIRCRAFT WITH SUCH FLIGHT CONTROL DEVICE," U.S. Patent US 2021/0 303 004 A1, 2021.
- [29] C. E. Billings, *Aviation automation: The search for a human-centered approach / Charles E. Billings*, ser. Human factors in transportation. Mahwah, N.J.: Lawrence Erlbaum Associates, 1997. ISBN 0805821260
- [30] J. D. Foster, E. Moralez, J. A. Franklin, and J. A. Schroeder, "Integrated Control and Display Research for Transition and Vertical Flight on the NASA V/STOL Research Aircraft (VSRA)," in *The future of vehicle electrical power systems and their impact on system design by G.A. Williams and M.J. Holt*. SAE International, 1991. doi: 10.4271/872329
- [31] B. Marthos and N. Duerkson, "Simplified Vehicle Operations (SVO), presented by the VFS/SFTE/SETP/AIAA Electric VTOL Flight Test Council," Webinar, 01.12.2020.
- [32] J. Denham, "STOVL Integrated Flight and Propulsion Control: Current Successes and Remaining Challenges," in *2002 Biennial International Powered Lift Conference and Exhibit*. AIAA, 2002. doi: 10.2514/6.2002-6021

- [33] D. Dollinger, V. A. Marvakov, and F. Holzapfel, “Increasing Operator Situational Awareness During Transition Procedures for Fixed-Wing VTOL UAV Operations,” in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-1179
- [34] J. Angelov and F. Holzapfel, “A Novel Command Concept for Simplified Vehicle Operations of Onboard Piloted VTOL Transition Aircraft,” in *Deutscher Luft- und Raumfahrtkongress 2021*, DLRK, Ed., 2021. doi: 10.25967/550011
- [35] European Aviation Safety Agency, “SPECIAL CONDITION Vertical Take-Off and Landing (VTOL) Aircraft, Doc. No: SC-VTOL-01,” 02.07.2019.
- [36] European Aviation Safety Agency, “Proposed Means of Compliance with the Special Condition VTOL, Doc. No: MOC SC-VTOL,” 12.05.2021.
- [37] European Aviation Safety Agency, “Proposed Means of Compliance with the Special Condition VTOL, Doc. No: MOC-3 SC-VTOL,” 29.06.2022.
- [38] M. A. Wechner, M. M. Marb, M. Zintl, D. Seiferth, and F. Holzapfel, “Design, Conduction and Evaluation of Piloted Simulation Mission Task Element Tests for Desired Behavior Validation of an eVTOL Flight Control System,” in *AIAA AVIATION 2022 Forum*. AIAA, 2022. doi: 10.2514/6.2022-3790
- [39] M. Ortlieb, F.-M. Adolf, and F. Holzapfel, “Computation of a Database of Trajectories and Primitives for Decision-Based Contingency Management of UAVs over Congested Areas,” in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021. doi: 10.1109/DASC52595.2021.9594333
- [40] M. Ortlieb and F.-M. Adolf, “Rule-Based Path Planning for Unmanned Aerial Vehicles in Non-Segregated Air Space over Congested Areas,” in *39th DASC - Digital Avionics Systems Conference*. IEEE, 2020. doi: 10.1109/DASC50938.2020.9256624
- [41] C. Krammer, S. Scherer, C. Mishra, and F. Holzapfel, “Concept for a Vision-Augmented Automatic Landing System for VTOL Aircraft,” in *AIAA AVIATION 2021 FORUM*. AIAA, 2021. doi: 10.2514/6.2021-3217
- [42] Federal Aviation Administration, “Urban Air Mobility Concept of Operations version 1.0,” 26.06.2020.
- [43] Brian P Hill, Dwight DeCarme, Matt Metcalfe, Christine Griffin, Sterling Wiggins, Chris Metts, Bill Bastedo, Michael D Patterson, and Nancy L Mendonca, “UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4,” NASA, 2020.
- [44] P. Goupil, “AIRBUS state of the art and practices on FDI and FTC in flight control system,” *Control Engineering Practice*, vol. 19, no. 6, 2011. doi: 10.1016/j.conengprac.2010.12.009

- [45] Y. C. Yeh, “Triple-triple redundant 777 primary flight computer,” in *1996 IEEE Aerospace Applications Conference. Proceedings*. IEEE, 1996. doi: 10.1109/aero.1996.495891
- [46] L. Sha, “Using simplicity to control complexity,” *IEEE Software*, vol. 18, no. 4, 2001. doi: 10.1109/MS.2001.936213
- [47] E. Bartocci and Y. Falcone, *Lectures on Runtime Verification*. Cham: Springer International Publishing, 2018, vol. 10457. ISBN 978-3-319-75631-8
- [48] P. Nagarajan, S. K. Kannan, C. Torens, M. E. Vukas, and G. F. Wilber, “ASTM F3269 - An Industry Standard on Run Time Assurance for Aircraft Systems,” in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-0525
- [49] Eric M. Peterson, Michael DeVore, Jared Cooper, and Greg Carr, “Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes,” *NASA*, 2020.
- [50] Society of Automotive Engineers Aerospace, “Guidelines for development of civil aircraft and systems: SAE ARP 4754 rev. A,” 21.12.2010.
- [51] RTCA, “RTCA DO-178C: Software Considerations in Airborne Systems and Equipment Certification,” 13.08.2018.
- [52] A. B. M. Moniruzzaman and S. A. Hossain, “Comparative Study on Agile software development methodologies,” *Global Journal of Computer and Science Technology (C)*, 2013. doi: 10.48550/arXiv.1307.3356
- [53] K. Dmitriev, S. A. Zafar, K. Schmiechen, Y. Lai, M. Saleab, P. Nagarajan, D. Dollinger, M. Hochstrasser, F. Holzapfel, and S. Myschik, “A Lean and Highly-automated Model-Based Software Development Process Based on DO-178C/DO-331,” in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE, 2020. doi: 10.1109/dasc50938.2020.9256576
- [54] M. Meyer, “Continuous Integration and Its Tools,” *IEEE Software*, 2014. doi: 10.1109/ms.2014.58
- [55] J. Shore and S. Warden, *The Art of Agile Development*. O’Reilly Media, Inc., 2021. ISBN 9781492080640
- [56] F. Paternò, *Model-Based Design and Evaluation of Interactive Applications*. Springer Science & Business Media, 1999. ISBN 9781852331559
- [57] RTCA, “RTCA DO-331: Model-Based Development and Verification Supplement to DO-178C and DO-278A,” 13.12.2011.

- [58] J. THOMAS and N. LEVESON, “Applying existing safety design techniques to software safety,” in *21st Aerospace Sciences Meeting*. AIAA, 1983. doi: 10.2514/6.1983-327
- [59] D. B. Turner, R. D. Burns, and H. Hecht, “Designing micro-based systems for fail-safe travel: For reliable control of railroads, aircraft, and space vehicles, designers are harnessing the power of the microprocessor,” *IEEE Spectrum*, vol. 24, no. 2, 1987. doi: 10.1109/MSPEC.1987.6448028
- [60] W. Hammer, *Handbook of system and product safety*. Prentice-Hall, 1972. ISBN 0133822265
- [61] M. E. Doyle, “Retrofit Reconfigurable Flight Control System and the F/A-18C,” Master’s thesis, Tennessee, 2006.
- [62] Federal Aviation Administration, *Advanced Avionics Handbook*. Winter Haven, Fl.: Pentagon Pub, 2011. ISBN 9781601707925
- [63] International Organization for Standardization, “ISO/IEC/IEEE 29148:2018-11: Systems and software engineering - Life cycle processes - Requirements engineering,” Berlin, 2018.
- [64] K. Bordignon and J. Bessolo, “Control Allocation for the X-35B,” in *2002 Biennial International Powered Lift Conference and Exhibit*. AIAA, 2002. doi: 10.2514/6.2002-6020
- [65] E. L. WIENER and R. E. CURRY, “Flight-deck automation: promises and problems,” *ERGONOMICS*, vol. 23, no. 10, 1980. doi: 10.1080/00140138008924809
- [66] N. Sarter, D. D. Woods, and C. E. Billings, “Automation surprises,” *Handbook of Human Factors and Ergonomics*, 2nd ed., 1997.
- [67] K. S. Bibby, F. Margulies, J. E. Rijnsdorp, R. Withers, and I. M. Makarov, “Man’s Role in Control Systems,” *IFAC Proceedings Volumes*, vol. 8, no. 1, Part 3, 1975. doi: 10.1016/S1474-6670(17)67612-2
- [68] L. Bainbridge, “Ironies of automation,” *Automatica*, vol. 19, no. 6, 1983. doi: 10.1016/0005-1098(83)90046-8
- [69] D. A. Norman, *The psychology of everyday things*. New York: Basic Books, 1988. ISBN 0465067093
- [70] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 30, no. 3, 2000. doi: 10.1109/3468.844354

-
- [71] Society of Automotive Engineers Aerospace, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” 30.04.2021.
- [72] C. E. Billings, *Human-centered Aircraft Automation: A Concept and Guidelines*. NASA, 1991.
- [73] M. Fernández, *Models of computation: An introduction to computability theory / by Maribel Fernández*, ser. Undergraduate topics in computer science. New York and London: Springer, 2009. ISBN 9781848824331
- [74] B. HOLDSWORTH and R. C. WOODS, Eds., *Digital logic design*, 4th ed. Oxford: Newnes, 2002. ISBN 9780750645829
- [75] B. HOLDSWORTH and R. C. WOODS, “Boolean algebra,” in *Digital logic design*, B. HOLDSWORTH and R. C. WOODS, Eds. Newnes, 2002, pp. 28–42.
- [76] A. Gill, *Introduction to the Theory of Finite-State Machines*. McGraw-Hill Inc., 1962, vol. 18. ISBN 0070232431
- [77] C. Krause, “Safe and Robust Automation of Aircraft and System Operation,” Dissertation, Technical University of Munich, Munich, 2022.
- [78] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages and computation*, 3rd ed. Boston, MA: Pearson/Addison Wesley, 2007. ISBN 0321455363
- [79] D. Kozen, Ed., *Automata and computability*, ser. UNDERGRADUATE TEXTS IN COMPUTE. New York, NY: Springer-verlag New York Inc, 2012. ISBN 978-1-4612-7309-7
- [80] European Organisation for Civil Aviation Equipment, “EUROCAE ED 281: MINIMUM AVIATION SYSTEM PERFORMANCE STANDARDS FOR RPAS AUTOMATION AND EMERGENCY RECOVERY,” 01.10.2020.
- [81] The MathWorks Inc. Modeling Guidelines for Code Generation. [Online]. Available: https://de.mathworks.com/help/releases/R2016a/pdf_doc/simulink/cg_guidelines.pdf (Accessed 28.08.2022).
- [82] The MathWorks Inc. Modeling Guidelines for High-Integrity Systems. [Online]. Available: https://de.mathworks.com/help/releases/R2016a/pdf_doc/simulink/hi_guidelines.pdf (Accessed 28.08.2022).
- [83] C. Krause and F. Holzapfel, “Implementing a multi-level finite state machine with MATLAB Simulink and Stateflow in the environment of high-integrity aircraft controller software,” in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2018. doi: 10.1109/ICCAR.2018.8384660

- [84] C. Krause and F. Holzapfel, "Designing a system automation for a novel UAV demonstrator," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2016. doi: 10.1109/ICARCV.2016.7838813
- [85] C. Silva, W. R. Johnson, E. Solis, M. D. Patterson, and K. R. Antcliff, "VTOL Urban Air Mobility Concept Vehicles for Technology Development," in *2018 Aviation Technology, Integration, and Operations Conference*. AIAA, 2018. doi: 10.2514/6.2018-3847
- [86] P. Yedamale, *AN885: Brushless DC (BLDC) motor fundamentals*. Microchip Technology Inc., 2003.
- [87] G. Liu, S. Chen, S. Zheng, and X. Song, "Sensorless Low-Current Start-Up Strategy of 100-kW BLDC Motor With Small Inductance," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, 2017. doi: 10.1109/tii.2016.2607158
- [88] J. C. Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil, "Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends," *Sensors*, vol. 10, no. 7, 2010. doi: 10.3390/s100706901
- [89] G. H. Jang, J. H. Park, and J. H. Chang, "Position detection and start-up algorithm of a rotor in a sensorless BLDC motor utilising inductance variation," *IEE Proceedings - Electric Power Applications*, vol. 149, no. 2, 2002. doi: 10.1049/ip-epa:20020022
- [90] N. Matsui and M. Shigyo, "Brushless DC motor control without position and speed sensors," *IEEE Transactions on Industry Applications*, vol. 28, no. 1, 1992. doi: 10.1109/28.120220
- [91] W. Johnson, *Helicopter Theory*. Courier Corporation, 2012. ISBN 9780486131825
- [92] G. P. Falconi, C. D. Heise, and F. Holzapfel, "Fault-tolerant position tracking of a hexacopter using an Extended State Observer," in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2015. doi: 10.1109/ICARA.2015.7081207
- [93] G. P. Falconi, V. A. Marvakov, and F. Holzapfel, "Fault tolerant control for a hexarotor system using Incremental Backstepping," in *2016 IEEE Conference on Control Applications (CCA)*. IEEE, 2016. doi: 10.1109/CCA.2016.7587842
- [94] M. Hedayatpour, M. Mehrandezh, and F. Janabi-Sharifi, "Propeller Performance in Presence of Freestream: Applications in Modeling Multirotor UAVs," in *Advances in Motion Sensing and Control for Robotic Applications*. Springer, Cham, 2019. doi: 10.1007/978-3-030-17369-2_4

-
- [95] M. Cerny, N. Herzog, J. Faust, M. Stuhlpfarrer, and C. Breitsamter, “Systematic Investigation of a Fixed-Pitch Small-Scale Propeller Under Non-Axial Inflow Conditions,” in *Deutscher Luft- und Raumfahrtkongress (DLRK) [67., 2018, Friedrichshafen]*, 2018.
- [96] M. Cerny and C. Breitsamter, “Investigation of small-scale propellers under non-axial inflow conditions,” *Aerospace Science and Technology*, vol. 106, no. 4, 2020. doi: 10.1016/j.ast.2020.106048
- [97] M. Söpper, J. Zhang, N. Bähr, and F. Holzapfel, “Required Moment Sets: Enhanced Controllability Analysis for Nonlinear Aircraft Models,” *Applied Sciences*, vol. 11, no. 8, 2021. doi: 10.3390/app11083456
- [98] J. Zhang, M. Söpper, and F. Holzapfel, “Attainable Moment Set Optimization to Support Configuration Design: A Required Moment Set Based Approach,” *Applied Sciences*, vol. 11, no. 8, 2021. doi: 10.3390/app11083685
- [99] Deutsches Institut für Normung, “DIN 50100:2016-12, Schwingfestigkeitsversuch - Durchführung und Auswertung von zyklischen Versuchen mit konstanter Lastamplitude für metallische Werkstoffproben und Bauteile,” Berlin, 2016.
- [100] J. D. Anderson, *Aircraft performance and design*. McGraw-Hill Education, 1999. ISBN 0070019711
- [101] F. N. Stoliker, *Introduction to Flight Test Engineering (Introduction aux techniques des essais en vol)*, ser. Flight test technique series. Neuilly-sur-Seine Cedex: N.A.T.O, 2005, vol. 14. ISBN 92-837-1126-2
- [102] Code of Federal Regulations, “Title 14: Aeronautics and Space: 14 C.F.R. § 25.111.”
- [103] Code of Federal Regulations, “Title 14: Aeronautics and Space: 14 C.F.R. § 1.2.”
- [104] Code of Federal Regulations, “Title 14: Aeronautics and Space: 14 C.F.R. § 25.121.”
- [105] Code of Federal Regulations, “Title 14: Aeronautics and Space: 14 C.F.R. § 25.149.”
- [106] Y. ZHU, J. WANG, Y. CHEN, and Y. WU, “CALCULATION OF TAKEOFF AND LANDING PERFORMANCE UNDER DIFFERENT ENVIRONMENTS,” *International Journal of Modern Physics: Conference Series*, vol. 42, 2016. doi: 10.1142/S2010194516601745
- [107] P. Bhardwaj, “Nonlinear Flight Control Strategies for Urban Air Mobility [Unpublished],” Dissertation, Technical University of Munich, Munich, 2023.
- [108] P. Bhardwaj, S. A. Raab, J. Zhang, and F. Holzapfel, “Thrust command based Integrated Reference Model with Envelope Protections for Tilt-rotor VTOL Transition UAV,” in *AIAA Aviation 2019 Forum*. AIAA, 2019. doi: 10.2514/6.2019-3266

- [109] P. Bhardwaj, S. A. Raab, J. Zhang, and F. Holzapfel, “Integrated Reference Model for a Tilt-rotor Vertical Take-off and Landing Transition UAV,” in *2018 Applied Aerodynamics Conference*. AIAA, 2018. doi: 10.2514/6.2018-3479
- [110] S. A. Raab, J. Zhang, P. Bhardwaj, and F. Holzapfel, “Proposal of a Unified Control Strategy for Vertical Take-off and Landing Transition Aircraft Configurations,” in *2018 Applied Aerodynamics Conference*. AIAA, 2018. doi: 10.2514/6.2018-3478
- [111] S. A. Raab, J. Zhang, P. Bhardwaj, and F. Holzapfel, “Consideration of Control Effector Dynamics and Saturations in an Extended INDI Approach,” in *AIAA Aviation 2019 Forum*. AIAA, 2019. doi: 10.2514/6.2019-3267
- [112] J. Zhang, P. Bhardwaj, S. A. Raab, and F. Holzapfel, “Control Allocation Framework with SVD-based Protection for a Tilt-rotor VTOL Transition Air Vehicle,” in *AIAA Aviation 2019 Forum*. AIAA, 2019. doi: 10.2514/6.2019-3265
- [113] P. Bhardwaj, S. A. Raab, and F. Holzapfel, “Higher Order Reference Model for Continuous Dynamic Inversion Control,” in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-1130
- [114] V. A. Marvakov and F. Holzapfel, “Defining Robust Transition and Re-Transition Procedures for Unmanned Fixed-Wing VTOL Aircraft,” in *AIAA Scitech 2021 Forum*. AIAA, 2021. doi: 10.2514/6.2021-1634
- [115] Society of Automotive Engineers Aerospace, “Aerospace - Passive Side Stick Unit General Requirements for Fly by Wire Transport and Business ARP6001,” 31.07.2012.
- [116] Society of Automotive Engineers Aerospace, “Aerospace - Passive Side Stick Unit General Requirements for Fly by Wire Transport and Business ARP6001A,” 10.04.2018.
- [117] Society of Automotive Engineers Aerospace, “Aerospace Active Inceptor Systems for Aircraft Flight and Engine Controls ARP5764,” 24.07.2018.
- [118] Society of Automotive Engineers Aerospace, “Aerospace - Passive Side Stick Unit General Requirements for Fly by Wire Transport and Business ARP6001B,” 23.04.2020.
- [119] D. Dollinger, J. Rhein, K. Schmiechen, and F. Holzapfel, “Be Lean — How to Fit a Model-Based System Architecture Development Process Based on ARP4754 Into an Agile Environment,” in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021. doi: 10.1109/dasc52595.2021.9594340
- [120] C. M. Ananda, “General aviation aircraft avionics: Integration & system tests,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 24, no. 5, 2009. doi: 10.1109/MAES.2009.5109949

-
- [121] J. E. Veitengruber, "Design Criteria for Aircraft Warning, Caution, and Advisory Alerting Systems," *Journal of Aircraft*, vol. 15, no. 9, 1978. doi: 10.2514/3.58409
- [122] Society of Automotive Engineers Aerospace, "Vehicle Management Systems - Flight Control Function, Design, Installation and Test of Piloted Military Aircraft, General Specification For AS94900A," 13.08.2018.
- [123] S. A. Shappell and D. A. Wiegmann, "U.S. naval aviation mishaps, 1977-92: differences between single- and dual-piloted aircraft," *Aviation, space, and environmental medicine*, vol. 67, no. 1, 1996.
- [124] P. Kurzahls and R. Onken, *Integrity in Electronic Flight Control System*. N.A.T.O, 1977.
- [125] M. Eigner, D. Roubanov, and R. Zafirov, Eds., *Modellbasierte virtuelle produktentwicklung*. Berlin, Germany: Springer Vieweg, 2014. ISBN 978-3-662-43815-2
- [126] J. R. Sklaroff, "Redundancy Management Technique for Space Shuttle Computers," *IBM Journal of Research and Development*, vol. 20, no. 1, 1976. doi: 10.1147/rd.201.0020
- [127] B. Hardekopf, K. Kwiat, and S. Upadhyaya, "Secure and fault-tolerant voting in distributed systems," in *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*. IEEE, 2001. doi: 10.1109/aero.2001.931341
- [128] J. H. Lala and R. E. Harper, "Architectural principles for safety-critical real-time applications," *Proceedings of the IEEE*, vol. 82, no. 1, 1994. doi: 10.1109/5.259424
- [129] N. A. Lynch, *Distributed algorithms*, ser. The Morgan Kaufmann series in data management systems. San Francisco, CA: Morgan Kaufmann Publishers, 1996. ISBN 9781558603486
- [130] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in the Presence of Faults," *Journal of the ACM*, vol. 27, no. 2, 1980. doi: 10.1145/322186.322188
- [131] K. Driscoll, B. Hall, M. Paulitsch, P. Zumsteg, and H. Sivencrona, "The real Byzantine Generals," in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*. IEEE, 2004. doi: 10.1109/DASC.2004.1390734
- [132] The MathWorks Inc. Configure data dictionary. [Online]. Available: <https://de.mathworks.com/help/simulink/slref/simulink.data.dictionary.html> (Accessed 12.10.2022).
- [133] W. F. Clement, R. H. Hoh, Ferguson, S. W., III, D. G. Mitchell, I. L. Ashkenas, and D. T. Mcruer, "Mission-Oriented Eequirements for Updating MIL-H-8501. Volume 1: STI Proposed Structure," *NASA*, 1985.

- [134] The MathWorks Inc. Modeling Considerations with Algebraic Loops - MATLAB & Simulink - MathWorks Deutschland. [Online]. Available: <https://de.mathworks.com/help/simulink/ug/modeling-considerations-with-algebraic-loops.html> (Accessed 16.10.2022).
- [135] The MathWorks Inc. Identify Algebraic Loops in Your Model - MATLAB & Simulink - MathWorks Deutschland. [Online]. Available: <https://de.mathworks.com/help/simulink/ug/identify-algebraic-loops-in-your-model.html> (Accessed 16.10.2022).
- [136] The MathWorks Inc. Algebraic Loop Concepts - MATLAB & Simulink - MathWorks Deutschland. [Online]. Available: <https://de.mathworks.com/help/simulink/ug/algebraic-loops.html> (Accessed 16.10.2022).
- [137] The MathWorks Inc. Inline Code - MATLAB & Simulink - MathWorks Deutschland. [Online]. Available: <https://de.mathworks.com/help/simulink/ug/inlining-functions.html> (Accessed 16.10.2022).
- [138] S. Sieberling, Q. P. Chu, and J. A. Mulder, "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, 2010. doi: 10.2514/1.49978
- [139] K. J. Astrom and L. Rundqwist, "Integrator Windup and How to Avoid It," in *1989 American Control Conference*. IEEE, 1989. doi: 10.23919/ACC.1989.4790464
- [140] M. P. Herlihy and J. M. Wing, "Specifying graceful degradation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, no. 1, 1991. doi: 10.1109/71.80192
- [141] V. A. Marvakov and F. Holzapfel, "A Framework for Simulation and Formal Verification of Redundant Flight Control Systems with Components Subject to Partially Synchronous Timing Effects," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021. doi: 10.1109/DASC52595.2021.9594390
- [142] M. T. Hochstrasser, "Modular model-based development of safety-critical flight control software," Dissertation, Technical University of Munich, Munich, 2020.
- [143] RTCA, "RTCA DO-333: Formal Methods Supplement to DO-178C and DO-278A," 13.12.2011.
- [144] J. B. Almeida, *Rigorous software development: An introduction to program verification*, ser. Undergraduate topics in computer science. London: Springer, 2011. ISBN 978-0-85729-018-2
- [145] The MathWorks Inc. Simulink Design Verifier Documentation - MathWorks Deutschland. [Online]. Available: <https://de.mathworks.com/help/sldv/> (Accessed 28.12.2022).

- [146] Diamond Aircraft Industries. Flight Training - Checklists. [Online]. Available: <https://www.diamondaircraft.com/en/service-and-support/diamond-flight-training/checklists/> (Accessed 28.09.2022).