# Design and Simulation of Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles

Christian Niebuhr



University of Bremen

Universität Bremen

DLR

# Design and Simulation of Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles

by

**Christian Niebuhr**

in partial fulfilment of the requirements for the degree of

**Master of Science**
in Space Engineering

at the University of Bremen,
in collaboration with the DLR Institute of Space Systems.

Project duration: Summer Semester 2026

First Supervisor: Dr. Stephan Theil
Second Supervisor (External): Jose Alfredo Maces Hernandez

Cover image: AI generated CALLISTO landing interpretation, generated using Gemini Nano Banana 2
with the following reference images:
`https://www.dlr.de/de/irs/bilder/images20/callisto/CALLISTO_description.jpg`
and
`https://www.friendsofnasa.org/2024/10/spacex-starship-fifth-flight-test_13.html`
Public domain. Accessed September 2026.

Universität Bremen

DLR

# Abstract

As space transportation systems across the globe move more and more towards reusability, control solutions are explored which can provide accelerated design cycles to facilitate fast turnover times under changing mission requirements. This thesis investigates whether Nonlinear Dynamic Inversion (NDI)-based control architectures are a feasible control solution for Vertical Take-off and Vertical Landing (VTVL) Reusable Launch Vehicles (RLVs) and whether they can thereby promote a change towards rapid design in this class of space transportation systems.

A specific implementation using a cascaded NDI-based control architecture for Thrust Vector Control (TVC) of pitch and yaw plane attitude and drift dynamics of the CALLISTO demonstrator as a case study is presented. For the low-energy hover flight demonstrations of the vehicle, the non-linear, rigid-body equations of motion are derived. Using small-angle approximations for the TVC deflection angles, these equations are transformed to become input-affine and a decomposition of attitude and drift dynamics into a cascaded structure is carried out. This is then used as the mathematical model at the core of the design of the inversion laws to construct the linearising feedback loops for attitude and drift dynamic inversion. An outer and inner, time-scale separated, stabilising feedback loop is devised by means of pole-placement, placing the attitude and drift closed-loop bandwidths one decade apart. In this cascaded loop structure, the drift controller provides an attitude correction increment, which is added to the reference of the inner loop to generate the required TVC deflection angle. The CALLISTO vehicle serves as an example to test feasibility and the resulting control architecture is not intended to be used during the actual flight campaigns.

The proposed control architecture is tested in a high-fidelity, non-linear CALLISTO simulator and the results underline good tracking performance under nominal conditions and a strong deterioration of performance under uncertain conditions in a Monte Carlo campaign with 53 sets of uncertain variables. The control solution is further analysed in a linear verification framework based on a linear reference plant and accordingly adjusted inversion laws where classical linear analysis is applied to showcase trends and isolate specific control behaviour. Both the system's evolution over the course of a trajectory and its robustness is analysed within this framework. Though margin requirements are upheld, the results also indicate the weak robustness of the method. On the other hand, they showcase clearly its core advantage: the unified plant transfer behaviour as a chain of integrators under changing conditions along a trajectory.

The presented design, based on classical NDI and a pole placement-based PD error controller, is found to be a feasible control solution in the investigated low-energy RLV scenarios, yet robustness issues lead to the conclusion that more advanced methods like incremental NDI, especially in combination with robust stabilising feedback loop design methods, are likely to hold more potential in the inversion-based control of VTVL RLVs for more complex mission scenarios.

# List of Figures


1.1 CALLISTO RLV demonstrator with constituent subsystems [1]. . . . . 1

2.1 Input-output feedback linearisation loop with state transformation and internal dynamics. . . . . 6
2.2 Input-state feedback linearisation loop with state transformation. . . . . 7
2.3 Generic NDI block diagram with a proportional error control gain for each of the error derivatives up to the system relative degree $\rho$. The inner linearisation loop creates the pseudo LTI, while the outer control loop stabilises the system. . . . . 9
2.4 Generic sensor-based INDI block diagram featuring pseudo control input and system control input increment construction from measurements (based on [2]). . . . . 11

3.1 Coordinate systems used to describe the RLV dynamics (based on [3]). . . . . 14
3.2 Low energy test flight mission profile. Normalised altitude over time on top, normalised dynamic pressure over time in the middle and total angle of attack over time below. . . . . 17
3.3 External forces and torques in the pitch plane acting on CALLISTO during ascent (based on [3]). . . . . 18

4.1 Full four-degrees-of-freedom cascaded control architecture. . . . . 28
4.2 Planar attitude NDI linearisation loop with equivalent double integrator representation. . . . . 31
4.3 Pitch plane stabilising feedback loop applied to the planar attitude NDI linearisation loop. . . . . 32
4.4 Cascaded plant structure using the pitch plane as an example. . . . . 34
4.5 Outer NDI linearising feedback loop with the inner stabilised attitude dynamics and nested plants. . . . . 36
4.6 Pitch plane directional drift controller applied to the planar directional drift NDI linearisation loop with time-scale separated inner loop transfer behaviour. . . . . 37
4.7 Parallel control architecture for attitude and translation using the pitch plane as an example. . . . . 39

5.1 Modularised, plane-agnostic NDI control architecture for SIMULINK implementation purposes. . . . . 42
5.2 Outer (a) and inner loop (b) reference and error signal construction. . . . . 43
5.3 Linear reference model data flow diagram. . . . . 46

6.1 Flowchart depicting the logical progression of the linear analysis suite. . . . . 50
6.2 Mission Profile of the low-energy test flight demonstration scenario, with 30 discrete time samples along the nominal trajectory. . . . . 52
6.3 Pole-zero map of the nominal linear reference plant and actuator dynamics $P(s)$ for 30 discrete time samples along the trajectory. . . . . 53
6.4 Step response of the nominal linear reference plant and actuator dynamics $P(s)$ for 30 discrete time samples along the trajectory. . . . . 53
6.5 Pole-zero map around the origin of the inner pseudo-linearised plant $P_{\mathrm{v,att}}(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 54
6.6 Bode diagram of the inner pseudo-linearised plant $P_{\mathrm{v,att}}(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 54

6.7 Bode and Nichols diagrams of the open-loop function $L_{\text{att}}(s)$ of the isolated attitude controller for 30 nominal discrete time samples along the trajectory. . . . . 55
6.8 Phase (a) and gain (b) margin evolution for the pitch plane attitude loop transfer function $L_{\text{att}}(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 56
6.9 Time and frequency response evolution for the pitch plane attitude closed-loop transfer function $T(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 56
6.10 Pole evolution for the pitch plane attitude closed-loop transfer function $T(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 57
6.11 Pole-zero map of the directional drift pseudo-linearised plant $P_{\text{v,trans}}(s)$ for 30 nominal discrete time samples along the trajectory. . . . . 57
6.12 Nichols diagrams of the open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ of the cascaded controller for 30 nominal discrete time samples along the trajectory. . . . . 58
6.13 Time response evolution for the cascaded closed-loop transfer functions $T_{\text{att}}(s)$ and $T_{\text{trans}}(s)$ with a unit step on the translational reference signal for 30 nominal discrete time samples along the trajectory. . . . . 59
6.14 Frequency response evolution for the cascaded closed-loop transfer functions $T_{\text{att}}(s)$ and $T_{\text{trans}}(s)$ with a unit step on the translational reference signal for 30 nominal discrete time samples along the trajectory. . . . . 59
6.15 Pole evolution for the pitch plane cascaded closed loop for 30 nominal discrete time samples along the trajectory. . . . . 60
6.16 Mission Profile of the first low-energy test flight demonstration scenario, $Q_{\max}$ marked as the nominal system choice. . . . . 61
6.17 Pole-zero map around the origin of the uncertain linear reference plant $P(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal poles are marked in red. 62
6.18 Pole-zero map around the origin of the uncertain pseudo-linear plants $P_{\text{v,att}}(s)$ and $P_{\text{v,trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal poles are marked in red. . . . . 62
6.19 Nichols diagrams of the uncertain open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal responses are marked in red. . . . . 63
6.20 Step responses of the uncertain closed-loop functions $T_{\text{att}}(s)$ and $T_{\text{trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal responses are marked in red. . . . . 63
6.21 System response of the planar controller of the linear reference model for a $20\frac{m}{s}$. wind disturbance applied to the $Q_{\max}$ uncertain plant for 20 randomised uncertain sets of variables. The nominal responses are marked in red. . . . . 65
6.22 Relevant parameters from the reference trajectory for the first and second flight demonstration scenarios of the CALLISTO vehicle. Normalised altitude, dynamic pressure and angle of attack over time are shown. A grey, dotted line indicates the guidance and control phase change from hover to landing in the second scenario. . 65
6.23 Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of the nominal, initial demonstration of CALLISTO. . . . . 66
6.24 Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified corner cases (green) for the initial demonstration of CALLISTO. . . . . 67
6.25 Normalised TVC deflection angles in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified corner cases (green) for the initial demonstration of CALLISTO. . . . . 68
6.26 Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of the nominal, second demonstration of CALLISTO. Shown are the ascent and hover phases during which the Non-linear Dynamic Inversion (NDI)-based control architecture is applied. . . . . 69

B.1 Gain and phase margin evolution for the planar cascaded control architecture on the loop functions of the outer $L_{\text{trans}}(s)$ and inner loops $L_{\text{att}}(s)$. . . . . VI

B.2 Nichols diagrams of the open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ of the cascaded controller based on the non-linear plant model for thirty discrete time samples along the trajectory. . . . . VII
B.3 Gain margin evolution for the isolated pitch plane attitude loop transfer function $L_{\text{att}}(s)$ for thirty discrete time samples along the trajectory. The inversion law is derived from the non-linear system equations. . . . . VII
B.4 Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified cases (green) for the second demonstration scenario of CALLISTO. Only ascent and hover flight phases are displayed. . . . . VIII
B.5 Normalised TVC deflection angles in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified cases (green) for the second demonstration of CALLISTO. Only ascent and hover flight phases are displayed. . . . . IX

C.1 Project Work Breakdown Structure. . . . . X

# List of Tables


3.1 Coordinate systems used in the description of launch vehicle dynamics (in part from [4]). . . . . 13

5.1 Dynamic effects represented within the six-degrees-of-freedom, high-fidelity CALLISTO simulator (based on [5, 6]). . . . . 42
5.2 Reference model and analysis user settings for the linear verification framework. . 47

6.1 Worst gain and phase margins along the trajectory for attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ based on the corresponding frequency responses under excitation of the $z_{\text{r}}$ signal. . . . . 58
6.2 Worst gain and phase margins among 20 sets of uncertain variables for attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ at $Q_{\max}$ based on the corresponding frequency responses under excitation of the $z_{\text{r}}$ signal. . . . . 63
6.3 Global worst gain and phase margin search results among 20 sets of uncertain variables at 30 discrete points along the trajectories. Applied to the attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ based on the corresponding frequency responses under excitation of the $\Delta\Theta$ and $z_{\text{r}}$ signals. . . . . 64

D.1 Documentation of AI tool usage. . . . . XV

# Glossaries

## Abbreviations

| | |
|---|---|
| ANDI | Actuator INDI |
| CALLISTO | Cooperative Action Leading to Launcher Innovation in Stage Toss back Operations |
| CNES | French Space Agency |
| CoM | Centre of Mass |
| CoP | Centre of Pressure |
| DCM | Direction Cosine Matrix |
| DLR | German Aerospace Center |
| DOF | Degree-of-Freedom |
| E-INDI | Extended INDI |
| ECEF | Earth Centred Earth Fixed |
| ECI | Earth Centred Inertial |
| EFUEN | Earth Fixed Up-East-North |
| EOM | Equation of Motion |
| FBL | Feedback Linearisation |
| FEM | Finite Element Method |
| GNC | Guidance, Navigation and Control |
| INDI | Incremental NDI |
| JAXA | Japan Aerospace Exploration Agency |
| LH2 | Liquid Hydrogen |
| LOX | Liquid Oxygen |
| LPF | Low-Pass Filter |
| LQR | Linear Quadratic Regulator |
| LTI | Linear Time-Invariant System |
| MC | Monte Carlo |
| MIMO | Multiple Input Multiple Output |
| MoI | Moment of Inertia |
| NDI | Non-linear Dynamic Inversion |
| OBC | On-Board Computer |

| | |
|---|---|
| PCH | Pseudo Control Hedging |
| RCS | Reaction Control System |
| RLV | Reusable Launch Vehicle |
| SISO | Single Input Single Output |
| TVC | Thrust Vector Control |
| VTOL | Vertical Take-Off and Landing |
| VTVL | Vertical Take-off Vertical Landing |
| WBS | Work Breakdown Structure |

# Symbols

| | | |
|---|---|---|
| $a$ | Translational Acceleration | $\frac{\mathrm{m}}{\mathrm{s}^2}$ |
| $C$ | Aerodynamic Coefficient | |
| $a_\mathrm{v}$ | Linearising Feedback | |
| $B_\mathrm{v}$ | Control Effectiveness | |
| $f$ | Force | N |
| $g$ | Gravitational Acceleration | $\frac{\mathrm{m}}{\mathrm{s}^2}$ |
| $h$ | Angular Momentum | $\frac{\mathrm{kg\ m}^2}{\mathrm{s}}$ |
| $h_\mathrm{alt}$ | Altitude | m |
| $J$ | Moment/Product of Inertia | $\mathrm{kg\ m}^2$ |
| $K_\mathrm{g}$ | Gain Margin | dB |
| $l$ | Length | m |
| $m$ | Mass | kg |
| $p$ | Linear Momentum | $\frac{\mathrm{kg\ m}}{\mathrm{s}}$ |
| $Q_\mathrm{dyn}$ | Dynamic Pressure | Pa |
| $S$ | Area | $\mathrm{m}^2$ |
| $t$ | Time | s |
| $T$ | Thrust | N |
| $u$ | Control System Input | |
| $v$ | Pseudo Control Input | |
| $x$ | Control System State | |
| $y$ | Control System Output | |
| $\alpha$ | Angle of Attack | rad |
| $\beta$ | Side-Slip Angle | rad |
| $\delta$ | TVC Deflection Angle | rad |
| $\epsilon$ | Euler Angles | rad |
| $\eta$ | Zero Dynamics | |
| $\gamma$ | Phase Margin | rad |
| $\omega$ | Angular Velocity | $\frac{\mathrm{rad}}{\mathrm{s}}$ |
| $\omega_\mathrm{n}$ | Natural frequency | $\frac{\mathrm{rad}}{\mathrm{s}}$ |
| $\rho$ | Relative Degree | |
| $\rho_\mathrm{d}$ | Density | |
| $\sigma$ | Time-Scale Separation Factor | |
| $\tau$ | Torque | Nm |
| $\xi$ | External Dynamics | |
| $\zeta$ | Damping Ratio | |
| A | Aerodynamic Frame | |

| | |
|---|---|
| B | Body Frame |
| I | Inertial Frame |
| S | Structural Frame |
| TF | Tip of Faring Frame |
| T | Trajectory/Kinematic Frame |

| | |
|---|---|
| 0 | Measured Signal Index |
| A | Aerodynamic Index |
| att | Attitude Index |
| c | Canonical Index |
| des | Desired Signal Index |
| e | Error Signal Index |
| G | Gravitation Index |
| r | Reference Signal Index |
| ref | Reference Geometry Index |
| T | Thrust Index |
| trans | Translational Index |
| w | Wind Index |

# Contents


**1 Introduction** **1**

**2 Inversion Control Fundamentals** **3**
2.1 Feedback Linearisation . . . 3
2.1.1 Input-Output Feedback Linearisation . . . 4
2.1.2 Input-State Feedback Linearisation . . . 7
2.2 Non-linear Dynamic Inversion Control . . . 8
2.2.1 Extensions of non-linear Dynamic Inversion . . . 10
2.3 Feedback Linearisation for MIMO Systems . . . 11

**3 System Modelling** **13**
3.1 Coordinate Systems . . . 13
3.1.1 Indexing . . . 14
3.2 General Formulation of Launch System Dynamics . . . 15
3.2.1 Rigid-Body Kinematics . . . 15
3.2.2 Simplifications for Launch Systems . . . 16
3.3 CALLISTO Dynamics . . . 17
3.3.1 The CALLISTO Mission and Vehicle . . . 17
3.3.2 Non-linear, Planar Attitude Dynamics . . . 19
3.3.3 Complete Non-linear Planar Dynamics . . . 22
3.3.4 Combined Pitch, Yaw, Lateral and Directional Dynamics . . . 25
3.3.5 Model Validity . . . 27

**4 NDI Based Controller Design** **28**
4.1 Control Strategy . . . 28
4.2 Planar Attitude Control . . . 29
4.2.1 Inner Linearisation Loop . . . 29
4.2.2 Inner Control Loop . . . 31
4.3 Inclusion of Directional Drift Control . . . 33
4.3.1 Outer Linearisation Loop . . . 34
4.3.2 Outer Control Loop . . . 36
4.4 Combined Pitch and Yaw Plane Control . . . 36
4.4.1 Yaw Plane Linearisation Loops . . . 37
4.4.2 Yaw Plane Control Loops . . . 38
4.5 Explored Design Choices . . . 38
4.5.1 Parallel Control Architecture . . . 38
4.5.2 Filters . . . 39

**5 Implementation and Simulation** **41**
5.1 Nonlinear CALLISTO Simulator . . . 41
5.1.1 Implementation Strategy . . . 41
5.1.2 Module Interfaces . . . 43
5.1.3 Discretisation . . . 44
5.2 Linear Verification Framework . . . 44
5.2.1 Linear Plant Model . . . 44
5.2.2 Adjusted Inversion Law . . . 45
5.2.3 Linear Reference Model Architecture . . . 45

**6 Results** **48**
6.1 Performance Metrics . . . 48
6.2 Linear Verification Framework Results . . . 49
6.2.1 System Evolution Analysis . . . 51
6.2.2 Robustness Analysis . . . 60
6.2.3 Disturbance Rejection . . . 64
6.3 Nonlinear Simulation Results . . . 65
6.3.1 Initial Flight Demonstration Scenario . . . 66
6.3.2 Second Flight Demonstration Scenario . . . 68

**7 Summary and Outlook** **70**

**Bibliography** **I**

**A Mathematical Concepts** **V**
A.1 Lie Derivatives and Lie Brackets . . . V

**B Additional Results** **VI**
B.1 Margin Evolution of the Cascaded Architecture . . . VI
B.2 Linearisation Loop Based on the Non-linear Model . . . VII
B.3 Second Test Flight Scenario Monte-Carlo Analysis . . . VIII

**C Project Management** **X**

**D Declarations** **XI**

# 1 Introduction

In recent years, efforts to decrease costs and increase sustainability have driven launch providers and governmental stakeholders across the globe to invest heavily in the development of Reusable Launch Vehicles (RLVs) [7]. In Europe, this trend has sprouted a number of projects to build local heritage in the operation of this class of space transportation systems [8, 9, 10, 11]. Among them is the Cooperative Action Leading to Launcher Innovation in Stage Toss back Operations (CALLISTO) mission, a joint effort of the German Aerospace Center (DLR), French Space Agency (CNES) and Japan Aerospace Exploration Agency (JAXA) that aims to demonstrate the core capabilities of Vertical Take-off Vertical Landing (VTVL) RLVs with a collaboratively developed, scaled vessel in a number of test flights of increasing complexity [12].

The vehicle and some of its constituent subsystems are depicted in figure 1.1. It measures approximately 13.5m in length with a diameter of around 1.1m and has a maximum take-off mass of under 4 tonnes. Its flight control system is equipped with the Japanese RSR-2 40kN LH2/LOX engine, which is gimballed with a Thrust Vector Control (TVC) system and can be throttled. Eight mono-propellant Reaction Control System (RCS) thrusters below the fairing control roll motion and attitude during unpowered flight and four aerodynamic control surfaces contribute towards controlling the vehicle's attitude during descent [12].

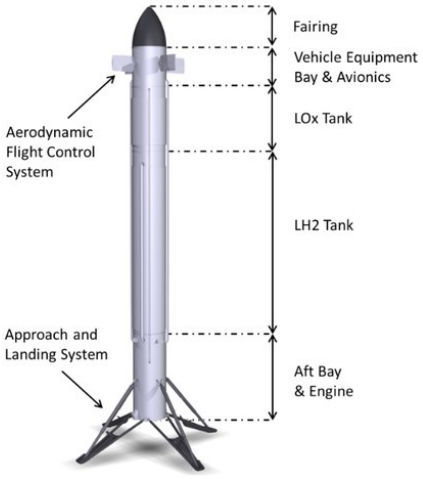



Figure 1.1: CALLISTO RLV demonstrator with constituent subsystems [1].

At the example of the CALLISTO vehicle, this study investigates the applicability of an NDI-based control architecture to the dynamics of VTVL RLVs during the ascent phase and beyond. The desired control architecture is not intended as flight software but merely as a feasibility study. NDI facilitates trajectory-independent control of non-linear systems, making it a promising prospect for fast turnaround, reusable systems while potentially reducing control complexity and thus reducing development costs and times [13]. Beyond that, NDI is uniquely suited for highly non-linear flight regimes as it does not merely approximate the non-linear dynamics of a system along a trajectory with local linear representations, but instead exactly linearises a systems behaviour in a large domain [14, 15]. The technique uses state feedback and inversion to accurately transform a system into a linear auxiliary representation in the form of a chain of integrators that can subsequently be

globally controlled by any linear stabilising control loop, eliminating the need for gain scheduling along a chosen state trajectory.

NDI is a term predominantly used in the domain of aeronautics for the more general method known as Exact Linearisation or Feedback Linearisation (FBL) which first appeared at the end of the last century [14, 16]. The term NDI and its emergence as an FBL technique became more precisely defined from the late 1980s onwards [17]. It was continuously refined over the last decades and work in the field sprouted a number of extensions trying to alleviate one of its major drawbacks. Feedback linearisation methods collectively suffer from an increased system model dependence as the linearisation loop at its core is derived from a mathematical model of the plant. The accuracy of the linearisation and thus the subsequent control performance directly depend on how precise the system model is. As a consequence, accurate system identification is necessary to make its application in critical systems feasible. Because this is costly and time-consuming, Incremental NDI (INDI) and its numerous variants add measurements or additional on-board modelling and estimation to reduce the model dependency and alleviate this issue [18, 19, 20].

NDI and its extension INDI have gathered growing attention in aeronautical control over the course of the last two decades, as indicated by a growing number of related publications [2]. A similar trend can be observed in the number of demonstrations ranging from Vertical Take-Off and Landing (VTOL) drones [21, 22] to large aeroplanes and fighter jets [23, 24, 25]. Steinert et al. give a comprehensive overview of the state-of-the-art demonstrations and developments, capturing in a detailed manner the trends in inversion control of recent years [2, 26]. In the domain of RLVs, inversion-based control strategies are explored for the application to winged- and lifting body configurations [27, 28], yet the application of NDI or INDI to missile type launch vehicles [29, 13] and specifically VTVL RLVs remains sparse.

This study aims to contribute to fill this void by demonstrating the feasibility of inversion-based control architectures for RLVs using the CALLISTO vehicle as a case study. Chapter 2 establishes the theoretical background around FBL and related methods and describes approaches to derive feedback laws for non-linear systems in a generic manner. Chapter 3 introduces the non-linear system dynamics of VTVL RLVs and specifically applicable to CALLISTO that form the basis of the mathematical model used in the derivations of the inversion laws. These are presented in chapter 4 with an incremental approach, starting with only the attitude dynamics and subsequently increasing complexity. The following chapters 5 and 6 introduce the non-linear simulation environment and showcase the performance of the design both in the high-fidelity simulator and applied to a linear reference model to enable the use of classical linear analysis methods. Finally, chapter 7 provides a summary of the project and an outlook for future work on the topic.

# 2 Inversion Control Fundamentals

Non-linear Dynamic Inversion (NDI) stems from a more general technique called Feedback Linearisation (FBL), whose fundamental concept revolves around cancelling non-linearities within a system's dynamics by means of feedback [30]. On a basic scale, this works similarly to how certain non-linearities can be inverted to create a pseudo-linear transfer behaviour from a new auxiliary input. As an example, imagine the system $\dot{x} = \mathbf{A}x + \mathbf{B}u^2$. It is non-linear in its input, yet a substitution of $u$ by the expression $u = \sqrt{v}$, which corresponds to introducing a new pseudo input $v$ before the original one, creates the aforementioned global pseudo-linear system $\dot{x} = \mathbf{A}x + \mathbf{B}v$. This, however, is only valid in the domain where the non-linear term is truly invertible. In this example, this is the case for $0 \leq v < \infty$. Besides the inversion of a system's non-linearity, feedback laws additionally incorporate the subtraction of non-linear terms in a feedback loop to achieve linearisation. This chapter presents the theoretical fundamentals of the methods that use these tools in order to linearise a wide range of dynamic systems and their evolution towards more sophisticated state-of-the-art methods.

## 2.1 Feedback Linearisation

As NDI is a subset of the FBL technique, which is also known as Exact Linearisation, an introduction to this more general approach shall be given in the following.

The basis of its derivation is a non-linear system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \tag{2.1}$$

$$y = h(\boldsymbol{x}) \tag{2.2}$$

This is a non-linear Single Input Single Output (SISO) system with smooth vector fields $\boldsymbol{f}$ and $\boldsymbol{g}$ and a smooth scalar function $h$, all within a domain of $\boldsymbol{x} \in D \subset \mathbb{R}^n$. The vector fields adhere to the mappings $\boldsymbol{f}(\boldsymbol{x}) : D \to \mathbb{R}^n$ and $\boldsymbol{g}(\boldsymbol{x}) : D \to \mathbb{R}^n$ whereas the scalar function adheres to $h(\boldsymbol{x}) : D \to \mathbb{R}$. An interpretation of the behaviour governed by each of the constituents of this non-linear system is given by their corresponding terminology. The vector field $\boldsymbol{f}(\boldsymbol{x})$ is referred to as the *drift vector field*. It contains information about the system's evolution without inputs. The second vector field $\boldsymbol{g}(\boldsymbol{x})$ describes the behaviour of the system induced by its inputs. It is therefore referred to as the *input vector field*. Finally, the scalar function $h(\boldsymbol{x})$ describes the output-state relation. It serves the same purpose as the C matrix in a linear system and is therefore analogously referred to as the *output function*.

As in the example in the introduction of this chapter, FBL tries to find an inversion law and further a change of variables, shown in equations 2.3 and 2.4, respectively, to transform the original non-linear system into an equivalent linear system [31].

$$u = B_{\mathrm{v}}(\boldsymbol{x})^{-1}(v - a_{\mathrm{v}}(\boldsymbol{x})) \tag{2.3}$$

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) \tag{2.4}$$

The inversion law in equation 2.3 is constructed in a way that, through the inversion of the so called *control effectiveness* $B_v(\boldsymbol{x})$ and the subtraction of the state-dependent *linearising feedback* $a_v(\boldsymbol{x})$, the original system's input-output map is linearised in the domain $D \subset \mathbb{R}^n$. To also linearise the system itself, the transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ is required. Both constituents are described in more detail below.

Linearisation by this approach is not possible for all forms of non-linear equations, however, as the system presented in equation 2.1 already belongs to a subclass of non-linear systems that are affine in input. Such systems have no non-linearities that depend on the input $u$, rather the state-dependent terms $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ contain all non-linear expressions. This assumption seems quite restrictive, yet techniques to transform a general non-linear system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}, u) \tag{2.5}$$

into an equivalent input-affine form are available [32, 33, 34, 35]. Further, systems are feedback linearisable only if the control effectiveness $B_{\mathrm{v}}(\boldsymbol{x})$ is non-zero and thus invertible. This puts a restriction on the actuation characteristics of a system. In SISO systems the control input's influence on the system must never become zero, as this would automatically make the control effectiveness zero and thus render it uninvertible.

A key characteristic of FBL methods is that $B_v(\boldsymbol{x})$ and $a_v(\boldsymbol{x})$ are derived from a model of the non-linear plant. The better this model resembles the real behaviour of the physical system, the better the linearity of the final pseudo-linear system. This introduces the most prominent drawback of FBL techniques. It is virtually impossible to perfectly depict a physical system's behaviour by means of modelling due to simplifications, computational errors and external disturbances [36]. The quality of the non-linear model used in the derivation of the inversion law directly governs the control performance, which makes system identification and modelling processes that much more important for FBL-based applications [30, 37]. If the model does not consider certain effects like actuator dynamics or if the states cannot be measured accurately to be used in the linearising state feedback, this characteristic drawback is most prominent [31]. The effect of model mismatches has been thoroughly studied and measures to circumvent this drawback led to the evolution of pure FBL into more sophisticated methods [36, 38].

Here, the fundamental approach to find the desired inversion law and state transformation for the SISO case in equations 2.3 and 2.4 that create the equivalent pseudo Linear Time-Invariant System (LTI) is described and subsequently subdivided into categories which range in generality.

### 2.1.1 Input-Output Feedback Linearisation

In Input-Output FBL, the goal is to **exactly linearise**, not the entire input-state map, but instead **a specific input-output behaviour** by finding a direct relation between a chosen system output $y(t)$ and the control input $u(t)$ [32]. To this end the system output $y(t)$ is differentiated until this relation is found. The number $\rho$ of derivatives required to find this relation is called the system's *relative degree*.

To simplify the repeated application of derivatives along the vector fields $\boldsymbol{f}$ and $\boldsymbol{g}$ to find the relative degree, the notations of differential geometry using *Lie Derivatives* $\mathcal{L}_{\boldsymbol{f}/\boldsymbol{g}}h(\boldsymbol{x})$ are applied. See Appendix A for a definition of this operation. In order to visualise the concept of relative degree, the output function of the input-affine system in 2.1 is differentiated using Lie Derivatives as shown below until a relation between the system input $u$ and the system output $y$ is found [31, 32].

$$\begin{aligned}
\dot{y} &= \nabla h(\boldsymbol{f} + \boldsymbol{g}u) = \mathcal{L}_{\boldsymbol{f}}h(\boldsymbol{x}) + \cancelto{=0}{\mathcal{L}_{\boldsymbol{g}}h(\boldsymbol{x})u} \\
\ddot{y} &= \mathcal{L}^2_{\boldsymbol{f}}h(\boldsymbol{x}) + \cancelto{=0}{\mathcal{L}_{\boldsymbol{g}}\mathcal{L}_{\boldsymbol{f}}h(\boldsymbol{x})u} \\
&\vdots
\end{aligned}$$

As long as the derivative of the scalar output function $h$ and all its derivatives along $\boldsymbol{f}$ remain zero along the vector field $\boldsymbol{g}$, one continues to differentiate. The system's relative degree $\rho$ in the region $D \subset \mathbb{R}^n$ is, $\forall \boldsymbol{x} \in D$, the derivative for which the term containing the system input is first non-zero as shown in the following.

$$\vdots$$

$$y^{(\rho-1)} = \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) + \cancelto{0}{\mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-2} h(\boldsymbol{x}) u}$$

$$y^{(\rho)} = \mathcal{L}_{\boldsymbol{f}}^{\rho} h(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) u \tag{2.6}$$

Once this relation is found, an inversion law of the form

$$u = \underbrace{\frac{1}{\mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x})}}_{B_v(\boldsymbol{x})^{-1}} (v - \underbrace{\mathcal{L}_{\boldsymbol{f}}^{\rho} h(\boldsymbol{x})}_{a_v(\boldsymbol{x})}) \tag{2.7}$$

can be defined. It contains the new *pseudo control input* $v$ which transforms the system's input-output mapping in equation 2.6 into

$$y^{(\rho)} = v \tag{2.8}$$

when inserted. This relation corresponds to a chain of $\rho$ integrators between pseudo control $v$ and the original system output $y$.

So far it has not been considered what shape the initial system in equation 2.1 must have to create an LTI upon applying the feedback law. Only the input output mapping has been shown to be linear. A generic input-affine system as in 2.1 is only feedback-linearisable if a so called *diffeomorphism* $T : D \to D_z \in \mathbb{R}^n$ exists that transforms the system using the change of variables $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ into the form

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}[a_v(\boldsymbol{x}) + B_v(\boldsymbol{x})u] \tag{2.9}$$

such that, when an inversion law of the form given in 2.3 is applied, an LTI remains [31]. The transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ is only classifiable as a diffeomorphism if its invertibility and the continuous differentiation of itself and its inverse are given [15, 31].

The state transformation thus ensures any non-linear, input-affine system can be transformed to become exactly linearised upon application of the inversion law.

A transformation that fulfils this requirement has the shape

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) \\ \vdots \\ \phi_{n-\rho}(\boldsymbol{x}) \\ - - - \\ h(\boldsymbol{x}) \\ \vdots \\ \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}) \\ - - - \\ \boldsymbol{\psi}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta} \\ - - - \\ \boldsymbol{\xi} \end{bmatrix} \tag{2.10}$$

which splits the system dynamics into two separate sections [31]. These are the *external dynamics* $\boldsymbol{\xi}$ that remain observable through the state transformation and the so called *zero dynamics* $\boldsymbol{\eta}$ that become unobservable through the transformation. When this transformation is applied to a non-linear system of the shape given in equation 2.1, the resulting new system equations in so called *canonical form* are

$$\dot{\boldsymbol{\eta}} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}) \tag{2.11}$$

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_c \boldsymbol{\xi} + \boldsymbol{b}_c (\mathcal{L}_{\boldsymbol{f}}^{\rho} h(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) u) \tag{2.12}$$

$$y = \boldsymbol{c}_c \boldsymbol{\xi} \tag{2.13}$$

with the canonical system matrices

$$\boldsymbol{A}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \boldsymbol{b}_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{c}_c = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Due to the nature of the canonical form, a repeated differentiation of $y$ up to the relative degree $\rho$ yields the final relation

$$y^{(\rho)} = \mathcal{L}_{\boldsymbol{f}}^{\rho} h(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) u \tag{2.14}$$

that was also the exact result of the continuous differentiation of the output equation in 2.6. Thus, by inserting the inversion law given in equation 2.7, the input-output map is once more transformed into a chain of integrators. With the transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$, it is now additionally ensured that the system also represents the desired LTI characteristics, with a final state space of

$$\dot{\boldsymbol{\eta}} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}) \tag{2.15}$$

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_c \boldsymbol{\xi} + \boldsymbol{b}_c v \tag{2.16}$$

$$y = \boldsymbol{c}_c \boldsymbol{\xi} \tag{2.17}$$

Figure 2.1 shows the linearisation loop resulting from applying the inversion law to the transformed non-linear system. The internal dynamics $\dot{\boldsymbol{\eta}}$ act like a sink in the signal flow, as they become unobservable after the state transformation [15]. The transfer behaviour from the pseudo control input $v$ to the system output $y$ is now approximated by a chain of $\rho$ integrators.

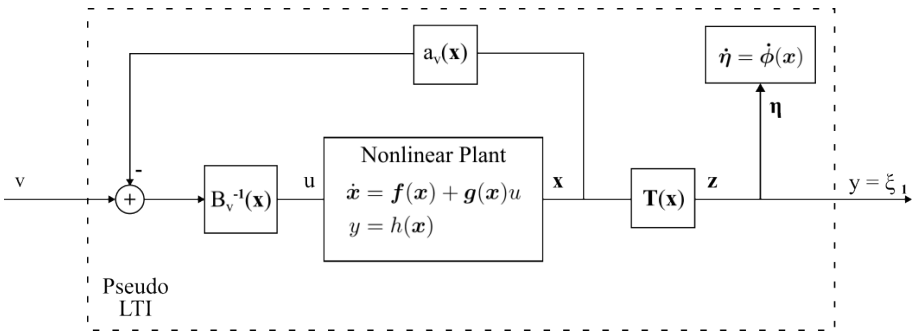



Figure 2.1: Input-output feedback linearisation loop with state transformation and internal dynamics.

The linearised system in equation 2.16, which describes the external dynamics $\dot{\boldsymbol{\xi}}$, can now be stabilised with any common linear control design method like pole placement or a Linear Quadratic Regulator (LQR) [35, 26]. Robust methods like $\mu$- and $\mathcal{H}_\infty$-synthesis are also being proposed to close the stabilising loop [22, 39] and even other non-linear frameworks like sliding mode controllers are investigated [40].

It should be noted that while the system equation 2.12 for the external dynamics $\dot{\boldsymbol{\xi}}$ can be found easily following the transformation defined in equation 2.10, the system equation 2.11 for the zero dynamics $\dot{\boldsymbol{\eta}}$ is more complex to solve. It is defined such that no dependence on the control input $u$ occurs [31], which enables finding a solution based on solving the differential equations

$$\nabla \boldsymbol{\phi}_i \boldsymbol{g}(\boldsymbol{x}) = 0 \qquad \forall\, i \in [1, 2, ..., n-\rho]\ ,\ \forall\, \boldsymbol{x} \in D \tag{2.18}$$

with the additional constraint that they must retain the diffeomorphic behaviour of the transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$. Though the zero dynamics are not observable after the transformation, to

ensure stability of the overall system, their internal stability has to be assessed carefully. Through a specific choice of the *control variables* defined in the output equation 2.2 and its scalar output function $h(\boldsymbol{x})$, the internal dynamics can often be stabilised in aeronautical applications [30].

One of the most important by-products of an input-output linearised system are these zero dynamics, which appear due to the difference between relative degree $\rho$ and system order $n$, which always adheres to $\rho \leq n$. Moving from the general case $\rho \leq n$ to the specific case of $\rho = n$ brings us into the subdomain of *input-state feedback linearisation*, described in the following.

### 2.1.2 Input-State Feedback Linearisation

In contrast to the result obtained via input-output FBL, input-state FBL aims to **exactly linearise the entire state space** of any input-affine, non-linear system like the one in equation 2.1. The key difference that leads to this method being applicable is choosing the output equation 2.2 to have a relative degree $\rho$ equal to the system order $n$. That way, no internal dynamics remain unobservable after transforming the system and the entire state space is linearised [31].

The state transformation in equation 2.10 reduces to the form

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} h(\boldsymbol{x}) \\ \vdots \\ \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) \end{bmatrix} \tag{2.19}$$

without internal dynamics $\dot{\boldsymbol{\eta}}$. This transformation leads to the system dynamics

$$\dot{\boldsymbol{z}} = \boldsymbol{A}_c \boldsymbol{z} + \boldsymbol{b}_c (\mathcal{L}_{\boldsymbol{f}}^{\rho} h(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{\rho-1} h(\boldsymbol{x}) u) \tag{2.20}$$

which are equal to the expression in equation 2.12 if $\boldsymbol{z} = \boldsymbol{\xi}$. This is given here, as no internal dynamics remain. As before, the insertion of the inversion law in equation 2.7 leads to a pseudo LTI that can again be stabilised via common linear control methods.

The resulting linearised system is depicted in figure 2.2. Now the transformed coordinate $\boldsymbol{z}$ does not contain any zero dynamics, hence the disappearance of the sink before the system output. From the perspective of the pseudo input $v$, the LTI again has the shape of a chain of $\rho = n$ integrators.

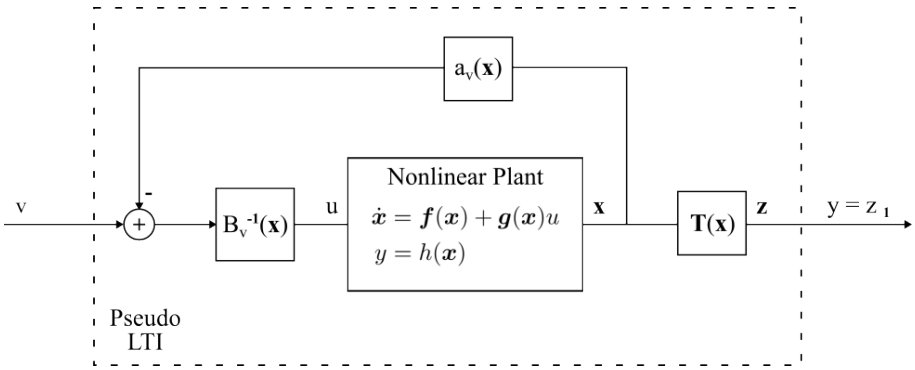



Figure 2.2: Input-state feedback linearisation loop with state transformation.

Equations 2.19 and 2.20 implicitly contain the system of partial differential equations that needs to be solved to find a suitable output function $h(\boldsymbol{x})$ with a relative degree $\rho = n$. Similar to the determination of the relative degree in equation 2.6, the relation

$$\mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}}^{i-1} h(\boldsymbol{x}) = 0 \qquad \forall\, i \in [1, 2, ..., n-1] \tag{2.21}$$

must hold for all derivatives of the output function $y^{(i)}$ under the condition that the final derivative $y^{(n)}$

$$\mathcal{L}_{\boldsymbol{g}}\mathcal{L}_{\boldsymbol{f}}^{n-1}h(\boldsymbol{x}) \neq 0 \tag{2.22}$$

Whether or not such a solution exists, however, is prone to two conditions that must be fulfilled in a domain $D_0 \subset D$ for the system at hand [31, 32]. If they are fulfilled, the system is called input-state linearisable. First, the matrix

$$\boldsymbol{\mathcal{G}}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}) & ad_{\boldsymbol{f}}\boldsymbol{g}(\boldsymbol{x}) & \cdots & ad_{\boldsymbol{f}}^{n-1}\boldsymbol{g}(\boldsymbol{x}) \end{bmatrix} \tag{2.23}$$

constructed from the smooth vector fields $\boldsymbol{f}$ and $\boldsymbol{g}$ in the system equation 2.1 has to have a full rank $\forall \boldsymbol{x} \in D_0$. It contains the Lie Brackets of $\boldsymbol{g}$ along $\boldsymbol{f}$ which, like the Lie Derivative, stem from differential geometry, explained in Appendix A.1. This first condition is sometimes called the *Controllability* condition [32]. Second, the distribution

$$\boldsymbol{\mathcal{D}} = span\{\boldsymbol{g}, ad_{\boldsymbol{f}}\boldsymbol{g}(\boldsymbol{x}), \cdots, ad_{\boldsymbol{f}}^{n-2}\boldsymbol{g}(\boldsymbol{x})\} \tag{2.24}$$

has to be involutive in the domain $D_0$. Involutivity demands that each Lie Bracket of individual spanning vector fields of a non-singular distribution exists and is itself also part of the distribution [31]. This condition is sometimes called the *Solvability* condition [32].

## 2.2 Non-linear Dynamic Inversion Control

As mentioned in the introduction of this chapter, Non-linear Dynamic Inversion (NDI) is a subcategory of FBL techniques, whose term is predominantly used in the domain of aeronautics [13, 30]. Its derivation follows the same logic as its parent method, yet most systems in aeronautics do not require a transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ to ensure that an LTI results from the application of the inversion law [30]. Further, the output equation in aeronautical applications can often be chosen such that the system's relative degree becomes $\rho = 1$ and no internal dynamics remain, technically making the result an input-state linearisation. Thus, the following derivation will assume this result. If indeed no transformation is required, the differentiation of the output equation, akin to the process described in subsection 2.1.1, then yields

$$\dot{y} = \mathcal{L}_{\boldsymbol{f}}h(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}}h(\boldsymbol{x})u \tag{2.25}$$

from which the inversion law

$$u = \frac{1}{\mathcal{L}_{\boldsymbol{g}}h(\boldsymbol{x})}(v - \mathcal{L}_{\boldsymbol{f}}h(\boldsymbol{x})) \tag{2.26}$$

can be derived. Upon inserting this relation into equation 2.25 the pseudo input-output map

$$\dot{y} = v \tag{2.27}$$

results. Just like in other FBL methods, this map corresponds to a chain of integrators with length equal to the relative degree of the system $\rho$. And since it is assumed no state transformation is required, inserting the inversion law 2.26 into the non-linear system equation yields

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\left(\frac{1}{\mathcal{L}_{\boldsymbol{g}}h(\boldsymbol{x})}(v - \mathcal{L}_{\boldsymbol{f}}h(\boldsymbol{x})\right) \tag{2.28}$$

which reduces to an LTI, completing the inner, so called linearisation loop. To subsequently ensure that the system is stable, an outer stabilising feedback loop is applied, which can, as previously presented, be based on a wide range of linear control methods. To this end, consider the tracking error

$$e = r - y \tag{2.29}$$

which is defined as the difference between the state reference $r$ and the measured system output $y$. With this relation and the pseudo input output map in equation 2.27, a definition for the pseudo control can be found. Taking the derivative of the error $\dot{e} = \dot{r} - \dot{y}$, rearranging it for $\dot{y}$ and inserting it into equation 2.27 results in

$$v = \dot{r} - \dot{e} \tag{2.30}$$

Solving the tracking problem aims to ensure that the tracking error asymptotically converges to zero, which can be fulfilled by the error dynamics

$$\dot{e} = K_p e \tag{2.31}$$

with the proportional gain $K_p$ as a free design parameter. For a system with an arbitrary relative degree $\rho$ the chain of integrators extends, and the definition of the pseudo control input and the desired error dynamics change to

$$v = r^{(\rho)} - e^{(\rho)}$$
$$e^{(\rho)} = \sum_{i=1}^{\rho} K_i e^{(i-1)}$$

with the scalar gains $K_i$ as design parameters. For this general case figure 2.3 shows both the linearisation loop which transforms the original non-linear system shown under *Non-linear Plant* into the integrator chain, and the outer control loop which stabilises the system asymptotically if the gains are chosen accordingly.

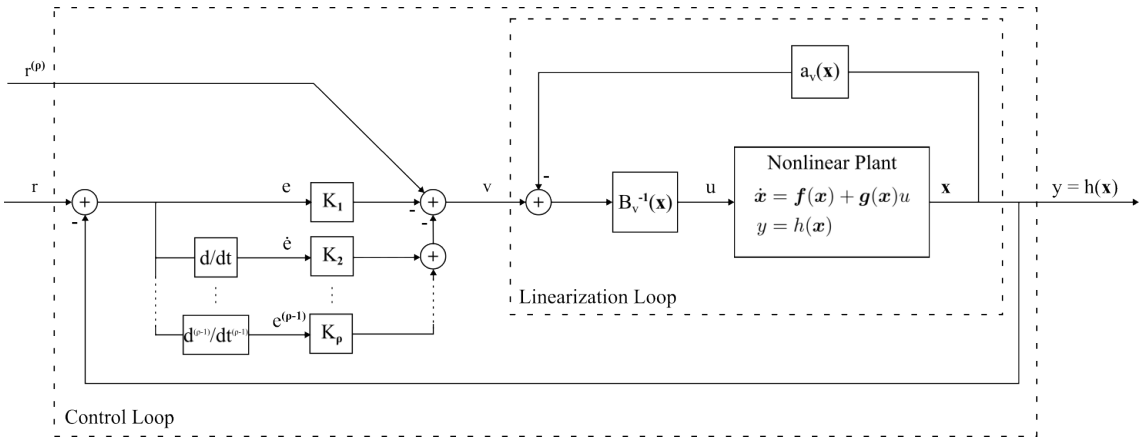



Figure 2.3: Generic NDI block diagram with a proportional error control gain for each of the error derivatives up to the system relative degree $\rho$. The inner linearisation loop creates the pseudo LTI, while the outer control loop stabilises the system.

NDI-based architectures promise to drastically improve the flexibility of the control solution by removing its dependence on linearisations along pre-flight trajectories and subsequent gain scheduling. This introduces a degree of mission agnosticism, making a found solution easily transferable to systems which are describable with a similar model. Finally though, it suffers the same drawbacks as its parent FBL methods. Model uncertainties still pose a threat to the performance of NDI-controlled systems in flight regimes where modelling errors occur, especially in the aerodynamics, like transonic or high-angle-of-attack flight [27, 41, 30]. The method is additionally only applicable without restrictions to a class of non-linear systems which are affine in input and have a square input-output relation.

These drawbacks, specifically the strong dependence on the system model accuracy, led to the development of extensions of the classic NDI inversion laws over the course of the last thirty years, which are presented in short in the following.

### 2.2.1 Extensions of non-linear Dynamic Inversion

The most well-known and diverse extension to NDI is a variant based on the generation of increments for the pseudo control input $v$ called INDI. Moreover, as presented earlier, the choice of design method for the stabilising feedback loop is free and can introduce certain measures of robustness and adaptability which are inherent to the specific method.

#### Incremental NDI

The incremental reformulation of the inversion law promises a range of advantages over NDI-based implementations, most notably a reduced dependence on the mathematical system model. This in turn improves the solution's disturbance rejection, as externally disturbing effects are usually not considered in the mathematical model and difficult to predict. Beyond that, the incremental adaptation is also suitable for input non-affine systems, extending the method's range of application considerably and further improving the mathematical model's accuracy [2].

The general concept works as follows. Reconsider the generic NDI law in equation 2.3, restated here as

$$u = B_v(\boldsymbol{x}_0)^{-1}(v_{\text{des}} - a_v(\boldsymbol{x}_0))$$

with adjusted indices to explicitly denote the states as measured and the pseudo control as the desired value. The form of the control effectiveness $B_v(\boldsymbol{x}_0)$ and linearising feedback $a_v(\boldsymbol{x}_0)$ is entirely based on the mathematical plant model. This dependence can be reduced by replacing $a_v(\boldsymbol{x}_0)$ with additional measurements of the current pseudo control input $v_0$ and the system control input $u_0$ [2]. To this end, solve the NDI law for $a_v(\boldsymbol{x}_0)$, resulting in

$$a_v(\boldsymbol{x}_0)) = v_0 - B_v(\boldsymbol{x}_0)u_0 \tag{2.32}$$

which, because of the introduced measurements, describes the current linearising feedback matrix. This relation is inserted into the NDI law to obtain the INDI law

$$\begin{aligned} u &= B_v(\boldsymbol{x}_0)^{-1} \cdot (v_{\text{des}} - (v_0 - B_v(\boldsymbol{x}_0) \cdot u_0) && (2.33)\\ &= B_v(\boldsymbol{x}_0)^{-1} \cdot (v_{\text{des}} - v_0) + u_0 && (2.34) \end{aligned}$$

which can be equally expressed with increments as

$$\Delta u = B_v(\boldsymbol{x}_0)^{-1} \cdot \Delta v \tag{2.35}$$

Because the feedback linearisation matrix no longer depends only on state measurements $\boldsymbol{x}_0$ but also on measurements of the current inputs $v_0$ and $u_0$, the law given in equation 2.34 is commonly referred to as sensor-based INDI [2]. Figure 2.4 depicts the corresponding block diagram, now featuring the actuator dynamics specifically in the plant signal path as well as navigation subsystem.

Diverse variants of this initial approach can be found, ranging from a return to strictly model-based approaches which are very closely related to the common NDI laws to mixtures of model- and sensor-based approaches, which are referred to as hybrid INDI with varying choices regarding the specific domains where measurements and where models are applied. Further, so called Actuator INDI (ANDI) and Extended INDI (E-INDI) approaches consider actuator dynamics specifically in the determination of the current system control input $u_0$. Among a range of additional smaller modifications Pseudo Control Hedging (PCH) is of particular note. It introduces a measure of protection against control allocation mismatches between the pseudo control input increment $\Delta v$ and the system control input increment $\Delta u$, which can stem for example from physical limitations of the actuator, leading to saturation [42].

Though INDI is widespread and outperforms NDI in most applications, the incremental method introduces a number of its own drawbacks, which should be carefully considered upon implementation. Measurements of the current inputs $v_0$ and $u_0$ might not always be easily obtainable and even if they are, noise on the signal might require the introduction of additional signal processing

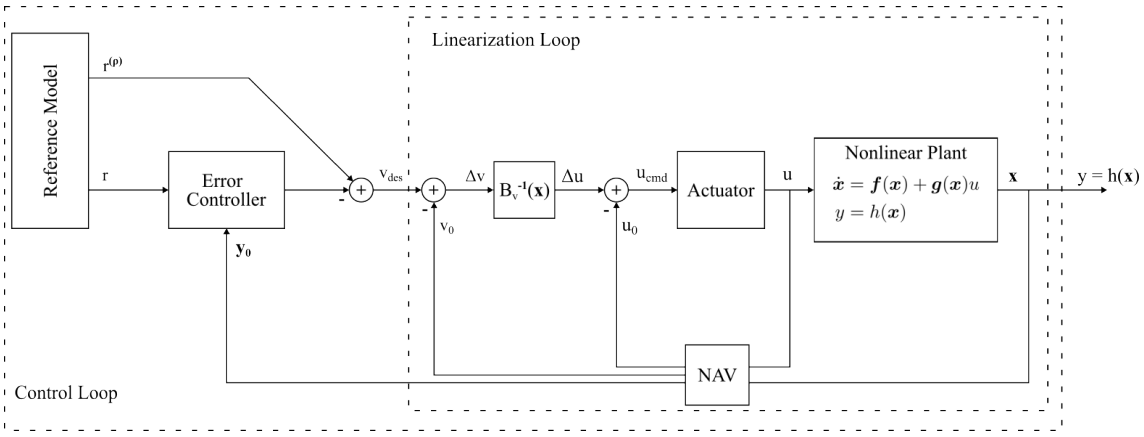



Figure 2.4: Generic sensor-based INDI block diagram featuring pseudo control input and system control input increment construction from measurements (based on [2]).

like filters, which in turn can introduce undesired dynamics. The precise synchronisation between the two measured signals has been identified as a major performance driver and understanding signal delays induced by processing and sensors is paramount to an effective INDI-based control implementation [43].

## 2.3 Feedback Linearisation for MIMO Systems

The preceding presentation of FBL fundamentals merely considered SISO systems at its core. With the understanding built therein, FBL is easily expandable to Multiple Input Multiple Output (MIMO) systems [15]. Here, the non-linear system dynamics change to

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{2.36}$$

with the expansion of the system inputs and outputs to vectors $\boldsymbol{u}, \boldsymbol{y} \in \mathbb{R}^m$. Accordingly, the surrounding elements change shape. The input vector field, formerly $\boldsymbol{g}(\boldsymbol{x})$, now $\boldsymbol{G}(\boldsymbol{x})$, becomes a matrix with functions of $\boldsymbol{x}$ describing the influence of each system input $u_\mathrm{i}\ \forall i \in [1, ..., m]$ on each state $x_\mathrm{i}\ \forall i \in [1, ..., n]$. The output function $\boldsymbol{h}(\boldsymbol{x})$ becomes a vector field containing one output function for each system output $y_\mathrm{i}\ \forall i \in [1, ..., m]$.

These changes propagate to the process of deriving the FBL inversion law in multiple ways. Instead of the relative degree of the whole system being a scalar, each transfer behaviour from individual inputs to outputs can be attributed a relative degree $r_\mathrm{i}\ \forall i \in [1, ..., m]$. The total relative degree $\rho$ of the system is then the sum of all constituent relative degrees.

Further along the path to the inversion law, the control effectiveness $\boldsymbol{B}_\mathrm{v}(\boldsymbol{x})$ becomes a square matrix and the linearising feedback $\boldsymbol{a}_\mathrm{v}(\boldsymbol{x})$ a vector. The requirement that the control effectiveness remains invertible in the whole linearised domain $D \subset \mathbb{R}^n$ still holds firm. Only now, the condition is for the matrix to be non-singular. Above, the number of inputs and outputs $m$ is assumed to be equal, rendering the system square. This is a common assumption of FBL-based MIMO applications to avoid control allocation complexities in the calculation of the control effectiveness inverse $\boldsymbol{B}_\mathrm{v}(\boldsymbol{x})^{-1}$ [13]. This concern is well studied and implementations of FBL-based control architectures are proposed that use pseudo-inverses instead, trading computational cost for feedback linearisability of under- or over-actuated systems [34, 26].

These changes conclude the adjustments introduced by the generalised formulation for MIMO systems. A final representation of the inversion law is obtained, which takes the form

$$\boldsymbol{u} = \boldsymbol{B}_\mathrm{v}(\boldsymbol{x})^{-1} \cdot (\boldsymbol{v} - \boldsymbol{a}_\mathrm{v}(\boldsymbol{x})) \tag{2.37}$$

where it is assumed that no state transformation $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ is required to obtain an LTI upon insertion. One of the by-products of the linearising feedback loop in MIMO systems is a decoupling

behaviour of the individual pseudo control input-output maps $y_{\mathrm{i}}^{(\rho)} = v_{\mathrm{i}}\ \forall i \in [1, ..., m]$ [2, 13]. When complexities arising from the treatment of a MIMO plant directly are to be avoided, transforming it into a cascaded multi-loop structure can in certain situations provide separated SISO systems. These can, when operated under time-scale separation, be designed independently of one another. Then, the derivations presented at the beginning of this chapter become applicable again.

# 3 System Modelling

As laid out in chapter 2, NDI and INDI rely heavily on the precision of the system model. Assumptions and simplifications made during the modelling process, and their effect on the accuracy of the model, should therefore be investigated with great care. In this chapter the foundations of rigid-body dynamics and their application to RLVss are described. These rigid-body Equations of Motion (EOMs) will serve as the basis for the subsequent linearisation and stabilising loop design process.

## 3.1 Coordinate Systems

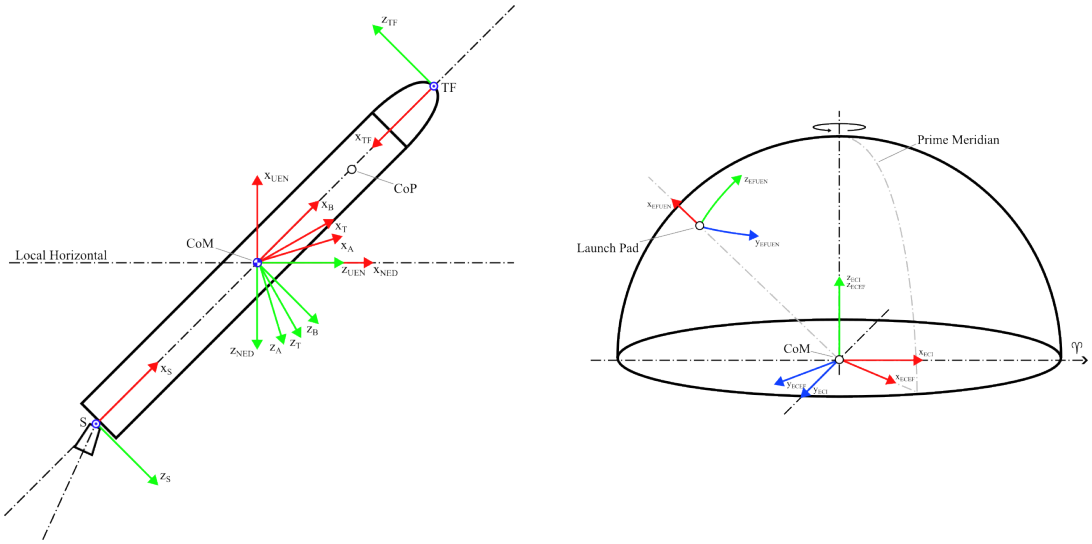
Table 3.1: Coordinate systems used in the description of launch vehicle dynamics (in part from [4]).

| **Abbr.** | **Name** | **Origin** | **Description** |
|---|---|---|---|
| Body Fixed Frames | | | |
| B* | Body | Body CoM | structural frame moved to the centre of mass |
| T | Trajectory | Body CoM | x: along the kinematic velocity vector<br>y: in earth horizontal plane<br>z: pointing downwards |
| A | Aerodynamic | Body CoM | x: along the aerodynamic velocity vector<br>y: in $x_A$-$y_T$ plane<br>z: pointing downwards |
| UEN | Up-East-North | Body CoM | x: zenith pointing<br>y: east pointing<br>z: north pointing |
| NED | North-East-Down | Body CoM | x: north pointing<br>y: east pointing<br>z: nadir pointing |
| TF* | Tip of Fairing | Tip of Fairing | body frame rotated 180° around the y axis and moved to the tip of the fairing |
| S | Structural | Gimbal Point | x: body longitudinal direction<br>y: defined by a structural element<br>z: completes the right-handed system |
| Earth Fixed Frames | | | |
| ECI | Earth Centred Inertial | Earth CoM | x: towards spring equinox<br>z: pointing towards geographical north<br>y: completing the right-handed system |
| ECEF | Earth Centred Earth Fixed | Earth CoM | x: towards prime meridian<br>z: pointing towards geographical north<br>y: completing the right-handed system |
| EFUEN* | Earth Fixed Up-East-North | Launch Pad Centre | x: upward, orthogonal to the earth's surface<br>y: east pointing<br>z: north pointing |

Describing the states of a dynamic system in terms of EOMs commonly involves a number of coordinate systems. In three-dimensional space, properties are expressed as vectors, and their

triplet components change depending on the reference coordinates. Thus, their context is crucial to establish a common ground for the description of any mathematical model. Here, an initial definition of used coordinate systems is presented. Table 3.1 gives an overview, in which the entries used to derive the EOMs are marked with an asterisk. Depending on the flight profile, the inertial reference frame of the body could be one of several of coordinate systems. For low-altitude flights the rotation of the earth can be neglected and either Earth Centred Inertial (ECI), Earth Centred Earth Fixed (ECEF) or even Earth Fixed Up-East-North (EFUEN) can be used as inertial references.

Also of note is how the body-fixed coordinate systems are aligned with respect to the geometry of the body. For aeroplanes that have no rotational symmetry around the longitudinal axis, the z-axis is usually defined to point away from the wing plane towards earth during nominal cruise flight with the y-axis completing the coordinate system. For bodies with rotational symmetry like many launch systems including CALLISTO, the definition is more difficult. As stated in table 3.1 a structural element on the rocket simply defines the y axis and the z axis completes the right-handed coordinate system.




(a) Body-fixed systems in the pitch plane

(b) Earth-fixed systems

Figure 3.1: Coordinate systems used to describe the RLV dynamics (based on [3]).

Figures 3.1a and 3.1b depict the coordinate systems in their respective fixed bodies. All body-fixed systems are shown in the pitch plane since pitch and yaw are equal and interchangeable. The relation between the earth and body-fixed coordinate systems is defined by the positioning of the launch system on the launch pad. The body axes (or interchangeably the structural axes) are placed according to the rotation $\sigma$

$$^{\mathrm{B}}R_{EFUEN} = R_1(\sigma)$$

around the coinciding x axes of the earth and body-fixed coordinate systems.

### 3.1.1 Indexing

The nomenclature used to describe both translational and rotational vectors is given below. In the case of a distance vector the notation reads *distance of* FROM *to* TO *in coordinate system* IN.

$$^{\mathrm{IN}}\boldsymbol{r}_{\mathrm{TO,FROM}}$$

In the case of all other physical properties, the notation reads *property of* OF *with respect to* WRT *in coordinate system* IN.

$$^{\mathrm{IN}}\boldsymbol{a}_{\mathrm{WRT,OF}}$$

Finally, rotations between coordinate systems are expressed with Direction Cosine Matrices (DCMs). Their description follows the nomenclature

$$^{\mathrm{TO}}\boldsymbol{T}_{\mathrm{FROM}}(\phi,\theta,\psi) = {}^{\mathrm{TO}}\boldsymbol{R}_{\mathrm{Aux2,x}}(\phi) \cdot {}^{\mathrm{Aux2}}\boldsymbol{R}_{\mathrm{Aux1,y}}(\theta) \cdot {}^{\mathrm{Aux1}}\boldsymbol{R}_{\mathrm{FROM,z}}(\psi) \tag{3.1}$$

which gives the rotation of a vector expressed *in coordinate system* FROM *to coordinate system* TO as a concatenation of rotations $\boldsymbol{R}$ around axes $z, y, x$ of the initial system and auxiliary systems $Aux1, Aux2$ by the Euler angles $\phi, \theta, \psi$.

## 3.2 General Formulation of Launch System Dynamics

To begin the derivation of RLV dynamics and later specifically the CALLISTO dynamics, a basic presentation of the generalised rigid body as a tool to approximate launch vehicle motion is given.

### 3.2.1 Rigid-Body Kinematics

It is assumed that a rigid body's motion can be fully described as the motion of its Centre of Mass (CoM). Following the general derivation of Siouris [44], the set of differential EOMs that describe a missile's translational and rotational motion as a rigid body originate from Newton's second law. It states that the sum of external forces acting on a body equals the rate of change of its momentum and equivalently for rotational motion that the sum of external torques acting on a body equals the rate of change of its angular momentum

$$\sum {}^{\mathrm{I}}\boldsymbol{f} = \frac{d}{dt}(m\,{}^{\mathrm{I}}\boldsymbol{v}) \tag{3.2}$$

$$\sum {}^{\mathrm{I}}\boldsymbol{\tau} = \frac{d}{dt}({}^{\mathrm{I}}\boldsymbol{r} \times m\,{}^{\mathrm{I}}\boldsymbol{v}) \tag{3.3}$$

with the linear momentum $\boldsymbol{p} = m\boldsymbol{v}$ and the angular momentum $\boldsymbol{h} = \boldsymbol{r} \times m\boldsymbol{v}$. These vector equations can be rewritten as scalar equivalents using the constituents

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad ; \quad \boldsymbol{v} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad ; \quad \boldsymbol{f} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad ; \quad \boldsymbol{\tau} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix}$$

The equations 3.2 and 3.3 are defined relative to the inertial reference frame, but the definition of external forces and velocities acting on the body is often much easier in the inherent body frame. A transformation into body coordinates yields an extra term that occurs when the body has an angular velocity $\boldsymbol{\omega}$ relative to the inertial frame [44]. For a generic vector this results in

$${}^{\mathrm{I}}\left(\frac{d\boldsymbol{a}}{dt}\right) = {}^{\mathrm{B}}\left(\frac{d\boldsymbol{a}}{dt}\right) + {}^{\mathrm{B}}\boldsymbol{\omega} \times {}^{\mathrm{B}}\boldsymbol{v}$$

with the indices I and B corresponding to the inertial and body reference frames. Using this relation to transform equations 3.2 and 3.3 into the body frame results in

$$\sum {}^{\mathrm{B}}\boldsymbol{f} = m\frac{d\,{}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}}}{dt} + m({}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times {}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}}) \tag{3.4}$$

$$\sum {}^{\mathrm{B}}\boldsymbol{\tau} = \frac{d\,{}^{\mathrm{B}}\boldsymbol{h}_{\mathrm{I,B}}}{dt} + {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times {}^{\mathrm{B}}\boldsymbol{h}_{\mathrm{I,B}} \tag{3.5}$$

with additional terms due to the rotating frame. This set of differential equations is also known as Euler's equations [44].

When the angular momentum $\boldsymbol{h}$ is rewritten in terms of the moment of inertia $\boldsymbol{J}$ and the angular velocity $\boldsymbol{\omega}$ using the relation $\boldsymbol{h} = \boldsymbol{J} \cdot \boldsymbol{\omega}$, the resulting formulation represents a common

form of the set of six non-linear differential equations of second-order that fully describe the 6 Degree-of-Freedom (DOF) motion of a rigid body in body coordinates [44].

$$\sum {}^{\mathrm{B}}\boldsymbol{f} = m\,{}^{\mathrm{B}}\dot{\boldsymbol{v}}_{\mathrm{I,B}} + m({}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times {}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}}) \tag{3.6}$$

$$\sum {}^{\mathrm{B}}\boldsymbol{\tau} = {}^{\mathrm{B}}\boldsymbol{J}\,{}^{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{I,B}} + {}^{\mathrm{B}}\dot{\boldsymbol{J}}\,{}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} + {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times ({}^{\mathrm{B}}\boldsymbol{J}\,{}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}}) \tag{3.7}$$

To be able to analyse the EOMs regarding full state behaviour and allow the construction of a state-space representation, they are further reformulated into two sets of first-order non-linear differential equations. They directly show the relationship between the position and attitude with their respective derivatives via the kinematic equations 3.8 and 3.10. This yields the general, rigid-body equations of motion for the translation

$${}^{\mathrm{I}}\dot{\boldsymbol{r}}_{\mathrm{I,B}} = {}^{\mathrm{I}}\boldsymbol{R}_{\mathrm{B}}\,{}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}} \tag{3.8}$$

$${}^{\mathrm{B}}\dot{\boldsymbol{v}}_{\mathrm{I,B}} = \frac{\sum {}^{\mathrm{B}}\boldsymbol{f}}{m} - {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times {}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}} \tag{3.9}$$

and for the rotation

$${}^{\mathrm{I}}\dot{\boldsymbol{\epsilon}}_{\mathrm{I,B}} = {}^{\mathrm{I}}\boldsymbol{T}_{\mathrm{B}}\,{}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \tag{3.10}$$

$${}^{\mathrm{B}}\dot{\boldsymbol{\omega}}_{\mathrm{I,B}} = {}^{\mathrm{B}}\boldsymbol{J}^{-1}\left(\sum {}^{\mathrm{B}}\boldsymbol{\tau} - {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} \times ({}^{\mathrm{B}}\boldsymbol{J}\,{}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}}) - {}^{\mathrm{B}}\dot{\boldsymbol{J}}\,{}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}}\right) \tag{3.11}$$

with states

$${}^{\mathrm{I}}\boldsymbol{r}_{\mathrm{I,B}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad ; \quad {}^{\mathrm{B}}\boldsymbol{v}_{\mathrm{I,B}} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad ; \quad {}^{\mathrm{I}}\boldsymbol{\epsilon}_{\mathrm{I,B}} = \begin{pmatrix} \Phi \\ \Theta \\ \Psi \end{pmatrix} \quad ; \quad {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{I,B}} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.12}$$

### 3.2.2 Simplifications for Launch Systems

Due to the characteristically long and slender geometry of launch vehicles and the type of mission profile, a number of simplifications can be made. It should, however, be noted that each of these simplifications comes at a cost of performance for NDI-based controllers as they deteriorate the accuracy of the model with regard to the real dynamics.

The geometry of the rigid-body affects the magnitudes of the constituents of the inertia matrix

$${}^{\mathrm{B}}\boldsymbol{J} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix}$$

For systems with a symmetry about the x-y plane like aeroplanes or certain launch systems like the Ariane 62 and the SLS, the products of inertia depending on y can be omitted. If, further, the system has a rotational symmetry about the longitudinal axis, as is the case for CALLISTO or for example the Vega C launch vehicle, the inertia moment about the z-axis is approximately equal to that about the y-axis. Products of inertia depending on z can therefore also be omitted [44]. With this, the moment of inertia matrix reduces to

$${}^{\mathrm{B}}\boldsymbol{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$$

with $J_{yy} \approx J_{zz}$.

Further simplifications can only be made if the mission profile and other vehicle specifications are known. An example of mission-specific simplifications is the assumption of a non-rotating earth for low-altitude trajectories. As an example of system-specific simplifications, consider a roll-rate-controlled vehicle. For this type of control strategy a roll rate of around zero can be a valid assumption.

## 3.3 CALLISTO Dynamics

Applying the rigid-body EOMs to the CALLISTO vehicle first requires a mission specification. This will yield a number of simplifications as well as the governing forces and moments, both from external sources and the actuators. The EOMs are then incrementally derived, adding more DOFs with each step to arrive at the full relevant description.

### 3.3.1 The CALLISTO Mission and Vehicle

The CALLISTO vehicle's capabilities are demonstrated in an incremental test flight approach with five different scenarios [12]. The flights range from low-energy hover scenarios in a fixed flight configuration to a high-energy return to launch site operation with unpowered aerodynamic reentry and system configuration changes like the deployment of landing legs or aerodynamic control surfaces. This work will focus on the low-energy scenarios and can as such make use of a range of simplifications.

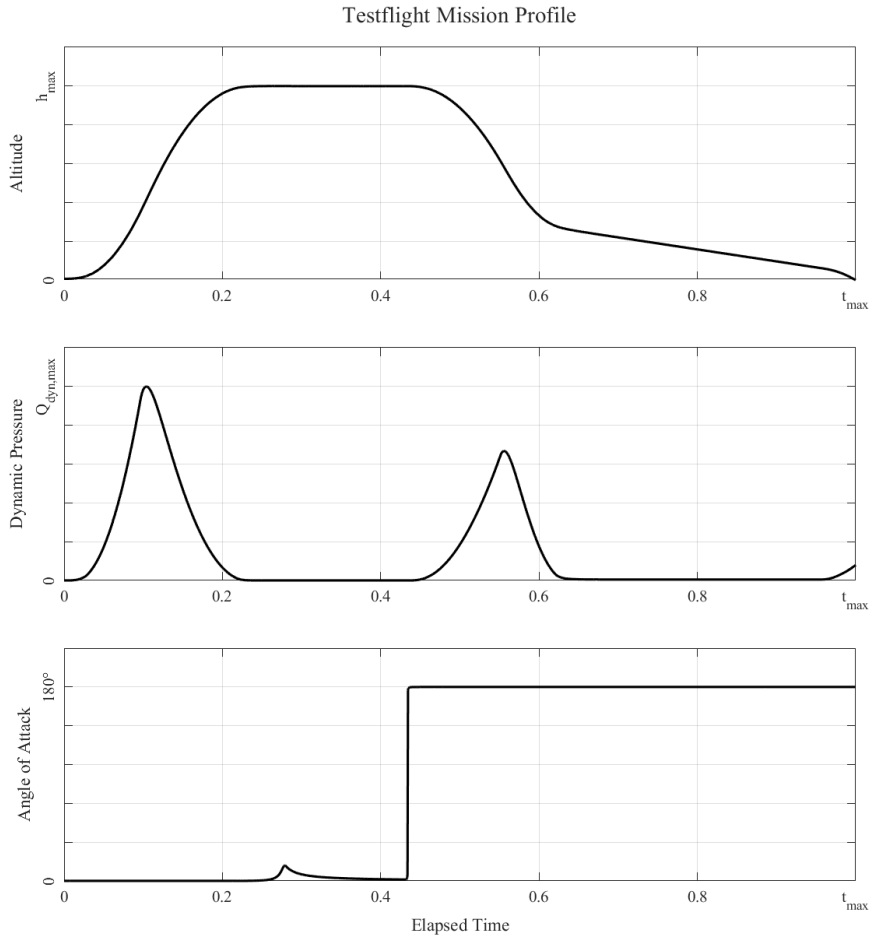



Figure 3.2: Low energy test flight mission profile. Normalised altitude over time on top, normalised dynamic pressure over time in the middle and total angle of attack over time below.

Figure 3.2 displays the trajectory of a low-energy test flight. Depicted are the normalised altitude and dynamic pressure over time as well as the total angle of attack over time. The altitude can be seen to follow the flight regimes ascent, hover, descent, approach and landing, which is a sequence similar for all low-energy demonstrations. They differ most notably in the configurations of the vehicle. The first demonstration, the one this work revolves around, remains in a fixed configuration, with stowed aerodynamic fins, deployed landing legs and a continuously operated

main engine. The depicted dynamic pressure gives an overview of the velocities during flight following the relation $Q_{\mathrm{dyn}} = \frac{1}{2}\rho_{\mathrm{d}} V_{\mathrm{A}}^2$ [44], which shows the dependence between dynamic pressure $Q_{\mathrm{dyn}}$ and free-stream velocity $V_{\mathrm{A}}$. The maximum dynamic pressure is a critical design driver as it governs the structural loads the vehicle experiences during launch. Because the considered scenario remains in the low-energy domain, the magnitude is much lower here than during later scenarios. Finally, the total angle of attack $\alpha_{\mathrm{tot}}(\alpha, \beta)$, which is a function of the angle of attack and side-slip angle, gives a measure of the vehicle's attitude. For the entire mission duration the vehicle remains upright, leading to an inversion of the angle of attack as soon as the descent flight phase begins.

Because the targeted low-energy trajectory has a fixed system configuration, aerodynamic modelling can be based on a single database instead of a complex mode switching mechanism. Further, the low altitude of the first test flights allows the rotation of the earth to be neglected and all earth-fixed frames to be treated as inertial.

Figure 3.3 shows the external forces and torques acting on the rocket over the course of the mission duration. Among them is the thrust force $f_{\mathrm{T}}$ applied to the origin of the structural frame at the gimbal point of the TVC system. It is decomposed into its contributions along the body axes, which in turn are used to derive the EOMs. The directional component $f_{\mathrm{Tz}}$, which depends on the TVC angle $\delta_{\mathrm{y}}$, creates a torque $\tau_{\mathrm{T}}$ that is used to actuate the vehicle's attitude during powered flight. Further, the aerodynamic force $f_{\mathrm{A}}$ acting on the Centre of Pressure (CoP) of the vehicle contributes to the vehicle's dynamics. It also generates a torque around the vehicle's CoM based on the directional component $f_{\mathrm{Az}}$. When the aerodynamic control surfaces are deployed, this force also contributes to the attitude control loop of the vehicle. Under the assumption of a spherical earth, the gravitational force $f_{\mathrm{G}}$ always points towards the earth's CoM. For the low-energy hover scenarios the gravitational force coincides with the longitudinal axis because the vehicle remains upright. As it acts on the CoM of the vehicle and a negligible gravity gradient along the vehicle is assumed, no moment occurs. Finally, an external roll torque $\tau_{\mathrm{RCS}}$ generated by the RCS acts on the body, allowing the control of its roll attitude motion.

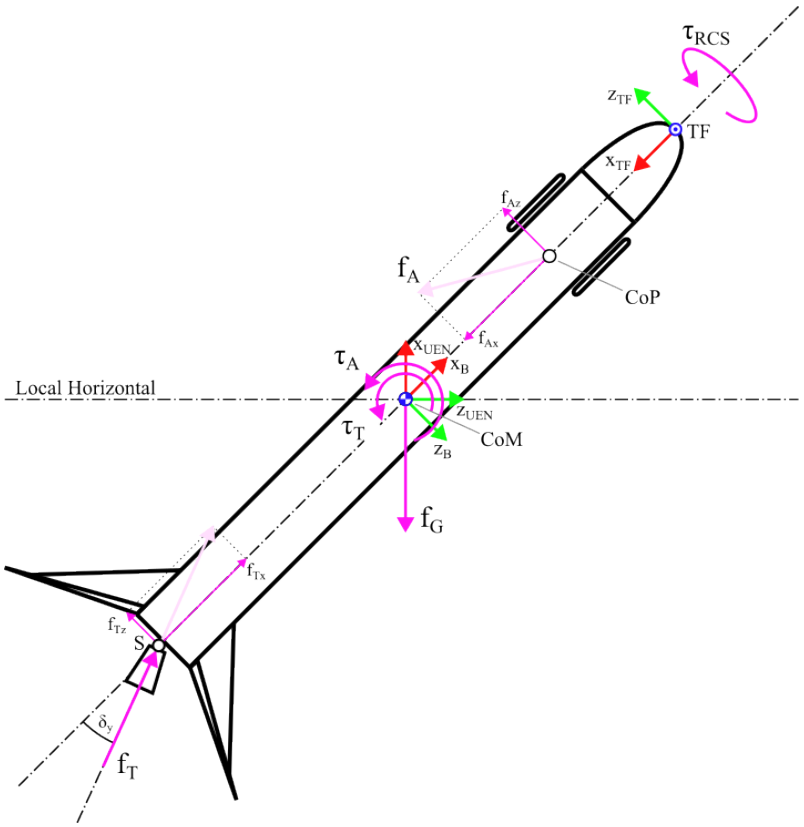



Figure 3.3: External forces and torques in the pitch plane acting on CALLISTO during ascent (based on [3]).

Three actuators contribute to the dynamics during the test flight at hand. The aerodynamic actuation surfaces remain stowed and therefore do not contribute. What remains are the engine

thrust modulation, the engine TVC and the RCS. The actual CALLISTO control architecture treats each actuator individually, which allows for separate control strategies. This study investigates whether the TVC system specifically can be controlled by an NDI-based controller. As will be shown below, this results in a four DOF controller for pitch and yaw rotational motion as well as lateral and directional drift translational motion. The translational motion along the longitudinal direction of the vehicle is only very marginally impacted by the range of motion the TVC system is capable of. The dominant contribution comes from the engine thrust and its controller. The remaining DOF in the roll motion cannot be controlled by the TVC system when a perfectly centred engine is assumed. The roll moment resulting from a non-perfectly centred engine is negligible compared to the control effort by the RCS, which is therefore the main roll actuator.

With this mission context established, the EOMs applicable for CALLISTO can be derived, starting with the pitch dynamics, which will be incrementally expanded.

### 3.3.2 Non-linear, Planar Attitude Dynamics

Taking the rigid-body equations of motion from equations 3.10 and 3.11 and decomposing them into their scalar constituents using the states from equation 3.12, the formulation

$$
{}^{\mathrm{I}}\begin{pmatrix}\dot{\Phi}\\ \dot{\Theta}\\ \dot{\Psi}\end{pmatrix} = {}^{\mathrm{I}}\boldsymbol{T}_{\mathrm{B}}\ {}^{\mathrm{B}}\begin{pmatrix}p\\ q\\ r\end{pmatrix} \tag{3.13}
$$

$$
\begin{aligned}
{}^{\mathrm{B}}\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{bmatrix}J_{\mathrm{xx}} & J_{\mathrm{xy}} & J_{\mathrm{xz}}\\ J_{\mathrm{yx}} & J_{\mathrm{yy}} & J_{\mathrm{yz}}\\ J_{\mathrm{zx}} & J_{\mathrm{zy}} & J_{\mathrm{zz}}\end{bmatrix}^{-1} \Bigg( &\begin{pmatrix}\sum {}^{\mathrm{B}}\tau_{\mathrm{x}}\\ \sum {}^{\mathrm{B}}\tau_{\mathrm{y}}\\ \sum {}^{\mathrm{B}}\tau_{\mathrm{z}}\end{pmatrix} - \begin{pmatrix}p\\ q\\ r\end{pmatrix} \times \begin{bmatrix}J_{\mathrm{xx}} & J_{\mathrm{xy}} & J_{\mathrm{xz}}\\ J_{\mathrm{yx}} & J_{\mathrm{yy}} & J_{\mathrm{yz}}\\ J_{\mathrm{zx}} & J_{\mathrm{zy}} & J_{\mathrm{zz}}\end{bmatrix} \begin{pmatrix}p\\ q\\ r\end{pmatrix} - \cdots\\
&- \begin{bmatrix}\dot{J}_{\mathrm{xx}} & \dot{J}_{\mathrm{xy}} & \dot{J}_{\mathrm{xz}}\\ \dot{J}_{\mathrm{yx}} & \dot{J}_{\mathrm{yy}} & \dot{J}_{\mathrm{yz}}\\ \dot{J}_{\mathrm{zx}} & \dot{J}_{\mathrm{zy}} & \dot{J}_{\mathrm{zz}}\end{bmatrix} \begin{pmatrix}p\\ q\\ r\end{pmatrix} \Bigg)
\end{aligned} \tag{3.14}
$$

emerges. To improve readability, coordinate system indices are removed and are only reintroduced if changes of note occur. The following rules apply to distinguish coordinate conventions. Euler angles $\boldsymbol{\epsilon}$ are always defined in the inertial frame I whereas angular velocities $\boldsymbol{\omega}$ and accordingly torques $\boldsymbol{\tau}$ are always defined in the body frame as indicated by the remaining indices in equations 3.13 and 3.14. Because of these conventions, a transformation matrix ${}^{\mathrm{I}}\boldsymbol{T}_{\mathrm{B}}$ results, which in the case of CALLISTO describes the rotation from the body coordinates into the inertial reference frame, chosen as EFUEN. The resulting matrix is

$$
{}^{\mathrm{I}}\boldsymbol{T}_{\mathrm{B}} = {}^{\mathrm{EFUEN}}\boldsymbol{T}_{\mathrm{B}} = \begin{bmatrix}1 & \sin\Phi\cos\Theta & \cos\Phi\sin\Theta\\ 0 & \cos\Phi & -\sin\Phi\\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta\end{bmatrix} \tag{3.15}
$$

As only the pitch dynamics are to be derived at this point, the equations in the x-z-plane can be extracted from equations 3.13 and 3.14. This yields the scalar equations

$$
\dot{\Theta} = \cos\Phi\ q - \sin\Theta\ r \tag{3.16}
$$

$$
\begin{aligned}
J_{\mathrm{yx}}\dot{p} + J_{\mathrm{yy}}\dot{q} + J_{\mathrm{yz}}\dot{r} = \sum \tau_{\mathrm{y}} &- r(J_{\mathrm{xx}}p + J_{\mathrm{xy}}q + J_{\mathrm{xz}}r) + \cdots\\
&+ q(J_{\mathrm{zx}}p + J_{\mathrm{zy}}q + J_{\mathrm{zz}}r) - (\dot{J}_{\mathrm{yx}}p + \dot{J}_{\mathrm{yy}}q + \dot{J}_{\mathrm{yz}}r)
\end{aligned} \tag{3.17}
$$

which are still generally applicable to any rigid body with the coordinate conventions that define the transformation matrix as in 3.15. Now, however, it is sensible to reduce these equations using a number of mission-specific simplifications that are valid not only for the CALLISTO demonstrations, but also for any other launch vehicle. As mentioned in section 3.2.2, the choice of coordinate system such that symmetries arise leads to a diagonal inertia matrix $\boldsymbol{J}$ where all products of inertia are approximately zero and the moments of inertia $J_{\mathrm{yy}} \approx J_{\mathrm{zz}}$. Applying this modification to the unsimplified scalar pitch dynamics in 3.17 results in the much shorter equation

$$J_{yy}\dot{q} = \sum \tau_y - J_{xx}pr + J_{zz}qr - \dot{J}_{yy}q \tag{3.18}$$

If further the roll angle $\Phi$ and rate $p$ are assumed to be zero by means of control with the RCS and the final simplification of assuming a rate of change of the moment of inertia close to zero is applicable, the reduced EOMs

$$\dot{\Theta} = q \tag{3.19}$$

$$J_{yy}\dot{q} = \sum \tau_y \tag{3.20}$$

which characterise the planar attitude dynamics of a slender vehicle with a negligible roll rate.

To finally obtain the non-linear differential equations that apply to CALLISTO's movement specifically, the external torques described in section 3.3.1 have to be inserted. The sum of external torques acting on the vehicle in the fixed configuration during the low-energy demonstrations can be represented by

$$\sum \tau_y = \tau_{Ty} + \tau_{Ay} + \tau_{RCSy} \tag{3.21}$$

all described in body coordinates B.

**Propulsive Torques**

The torque resulting from the main engine thrust and TVC deflection dominates the planar attitude dynamics. It can be constructed from the propulsive force vector and the position of the CoM with respect to the structural frame origin S as

$$^{B}\boldsymbol{\tau}_{T} = {}^{S}\boldsymbol{r}_{CoM} \times {}^{S}\boldsymbol{f}_{T} \tag{3.22}$$

$$= \begin{pmatrix} 0 \\ -{}^{S}r_{CoM,x}\,{}^{S}f_{T,z} \\ 0 \end{pmatrix} \tag{3.23}$$

$$= \begin{pmatrix} 0 \\ -{}^{S}l_{CoM}T\sin(\delta_y) \\ 0 \end{pmatrix} \tag{3.24}$$

with the main engine thrust magnitude $T$ and the scalar distance between the two coordinate systems, which collapses to the value of the longitudinal contribution of the position vector ${}^{S}l_{CoM} \approx {}^{S}r_{CoM,x}$, as the two origins are approximately aligned on the vehicle's longitudinal axis. This distance acts as the lever arm for the thrust force to create the torque around the CoM. For thrust vector attitude-controlled systems this is where the control input, the TVC deflection $\delta_y$, enters the system's attitude dynamics.

**Aerodynamic Torques**

The second term that contributes to the sum of external torques in equation 3.21 contains the aerodynamics. To allow for a simplified depiction of the aerodynamic torques, the origin of the coordinate system in which they are expressed in CALLISTO is not the CoP, which would physically be the point where all aerodynamic torques are zero, but instead the tip of the fairing [45]. This transforms the origin of the aerodynamic reference system from being a moving point to being a constant, well-defined coordinate in all body-fixed frames. Aerodynamic databases created for the vehicle are therefore transformed according to this convention to allow the aerodynamic forces and torques to be described at any coordinate without considering a moving reference. In the body coordinate system the aerodynamic force vector takes the shape

$$^{B}\boldsymbol{f}_{A} = Q_{dyn}S_{ref}\,{}^{B}\begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} = Q_{dyn}S_{ref}\,{}^{B}\boldsymbol{R}_{TF}\,{}^{TF}\begin{pmatrix} C_A \\ C_S \\ C_N \end{pmatrix} \tag{3.25}$$

with the index TF denoting the tip of fairing coordinate frame and the aerodynamic coefficients in this frame $C_\mathrm{A}$, $C_\mathrm{S}$ and $C_\mathrm{N}$ corresponding to the axial, side and normal forces. The characteristic surface of the aerodynamic body $S_\mathrm{ref}$ and the dynamic pressure $Q_\mathrm{dyn}$ are used to transform the coefficients into forces. The rotation matrix ${}^\mathrm{B}\boldsymbol{R}_\mathrm{TF}$ simply describes a rotation by 180° around the y-axis [3] such that the available aerodynamic data can be easily transformed from the TF frame to the body frame.

Aerodynamic characterisation with both computational methods [46] and wind tunnel tests [45] has been performed to find the characteristic surface $S_\mathrm{ref}$ and length $l_\mathrm{ref}$ of the vehicle. These investigations also yield an aerodynamic database as a multidimensional point matrix that can be interpolated for different flight conditions to find the approximate current aerodynamic coefficients. The dynamic pressure is a function of the surrounding air density $\rho$ and the free-stream velocity $V_\mathrm{A}$ according to the formula $Q_\mathrm{dyn} = \frac{1}{2}\rho_\mathrm{d} V_\mathrm{A}^2$ [3, 44].

Since the aerodynamic forces produce zero torque only at the CoP, the aerodynamic torques at the CoM have to be derived using both the forces ${}^\mathrm{TF}\boldsymbol{f}_\mathrm{A}$ and torques ${}^\mathrm{TF}\boldsymbol{\tau}_\mathrm{A}$ described in the adjusted reference frame TF. The resulting formula for the vector describing the aerodynamic torque in the body coordinate system can be found by following the derivation in the appendix of [3], which describes the steps (with adjusted nomenclature)

$$
\begin{aligned}
{}^\mathrm{B}\boldsymbol{\tau}_\mathrm{A} &= Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref} \; {}^\mathrm{B}\begin{pmatrix} C_\mathrm{l} \\ C_\mathrm{m} \\ C_\mathrm{n} \end{pmatrix} && (3.26)\\
&= {}^\mathrm{B}\boldsymbol{R}_\mathrm{TF} \left( {}^\mathrm{TF}\boldsymbol{\tau}_\mathrm{A} + {}^\mathrm{TF}\boldsymbol{r}_\mathrm{B,TF} \times {}^\mathrm{TF}\boldsymbol{f}_\mathrm{A} \right) && (3.27)\\
&= {}^\mathrm{B}\boldsymbol{R}_\mathrm{TF} \left( Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref} \; {}^\mathrm{TF}\begin{pmatrix} C_\mathrm{Mx} \\ C_\mathrm{My} \\ C_\mathrm{Mz} \end{pmatrix} + Q_\mathrm{dyn} S_\mathrm{ref} \; {}^\mathrm{TF}\begin{pmatrix} r_\mathrm{B,TFx} \\ r_\mathrm{B,TFy} \\ r_\mathrm{B,TFz} \end{pmatrix} \times {}^\mathrm{TF}\begin{pmatrix} C_\mathrm{A} \\ C_\mathrm{S} \\ C_\mathrm{N} \end{pmatrix} \right) && (3.28)\\
&= Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref} \begin{pmatrix} -{}^\mathrm{TF}C_\mathrm{Mx} \\ {}^\mathrm{TF}C_\mathrm{My} + \frac{{}^\mathrm{B}l_\mathrm{TF}\,{}^\mathrm{TF}C_\mathrm{N}}{l_\mathrm{ref}} \\ -{}^\mathrm{TF}C_\mathrm{Mz} + \frac{{}^\mathrm{B}l_\mathrm{TF}\,{}^\mathrm{TF}C_\mathrm{S}}{l_\mathrm{ref}} \end{pmatrix} && (3.29)
\end{aligned}
$$

Besides the variables already used to describe the aerodynamic forces in the body frame in equation 3.25, the final formula for the aerodynamic torques given in equation 3.29 contains the aerodynamic coefficients for the torques in the TF frame ${}^\mathrm{TF}C_\mathrm{Mx}$, ${}^\mathrm{TF}C_\mathrm{My}$ and ${}^\mathrm{TF}C_\mathrm{Mz}$, the lever arm between the CoM and the origin of the aerodynamic reference frame TF ${}^\mathrm{B}l_\mathrm{TF}$ in body coordinates as well as the aerodynamic reference length of the vehicle $l_\mathrm{ref}$. Carefully assessing the signs in this formula is paramount to correctly describing the aerodynamic influences on the dynamics of the system as there are many potential sources of error in its derivation. To this end it should be noted that the lever arm vector ${}^\mathrm{TF}\boldsymbol{r}_\mathrm{B,TF}$ is dominated by its entry along the TF x-axis ${}^\mathrm{TF}r_\mathrm{B,TFx}$ for the CALLISTO vehicle, as the CoM and the origin of the TF frame lie approximately within the same axis. The resulting lever arm can either be expressed in the body or in the TF frame with the relation ${}^\mathrm{B}l_\mathrm{TF} = -{}^\mathrm{TF}l_\mathrm{B}$ and by convention the former is chosen.

### RCS Torques

The contribution of the RCS can be neglected, as the operation cycle is designed to control roll motion by simultaneous activation of opposite engines with equal force, leading to an approximate net zero torque in the pitch and yaw directions respectively.

### Non-linear Planar Attitude EOMs

Having fully described the torques acting on the system, the respective pitch plane constituents can now be inserted into the slender vehicle planar EOMs from equation 3.20 to yield the CALLISTO pitch dynamics

$$
\begin{aligned}
\dot{\Theta} &= q && (3.30)\\
\dot{q} &= J_\mathrm{yy}^{-1} \left( Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref} \left( {}^\mathrm{TF}C_\mathrm{My} + \frac{{}^\mathrm{B}l_\mathrm{TF}\,{}^\mathrm{TF}C_\mathrm{N}}{l_\mathrm{ref}} \right) - {}^\mathrm{S}l_\mathrm{CoM} T \sin\left(\delta_\mathrm{y}\right) \right) && (3.31)
\end{aligned}
$$

A few details are of note considering the applicability of this formulation for inversion-based linearisation implementations. So far no simplifications as to the non-linearity of the system equations have been made. Equation 3.31 is non-linear with respect to the control input $\delta_\mathrm{y}$, which, as stated in chapter 2, requires further effort to transform the system into an input linearisable form.

Further, the aerodynamic coefficients are a function of the angle of attack $\alpha$, the side-slip angle $\beta$, the free-stream velocity $V_\mathrm{A}$ and the altitude $h_\mathrm{alt}$. This relation is highly non-linear and difficult to describe analytically, which leads to a linearisation technique using stability derivatives $\frac{\partial C}{\partial \alpha}$ in classical linear control methods [44]. Since inversion controllers do not require the system to behave linearly with respect to the states, this is not required here. Using available aerodynamic databases and interpolating them based on measured states provides the required information and in turn retains the non-linear behaviour.

To express the dynamics in a form akin to the generic non-linear equation 2.5, the vector fields $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ can be introduced as

$$\begin{bmatrix}\dot{\Theta}\\ \dot{q}\end{bmatrix} = \underbrace{\begin{bmatrix}q\\ \mu_\alpha(\alpha,...)\end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix}0\\ -\mu_{\delta\mathrm{y}}(\delta_\mathrm{y})\end{bmatrix}}_{\boldsymbol{g}(u)} \tag{3.32}$$

with the functions $\mu_\alpha(\alpha,...)$ and $\mu_{\delta\mathrm{y}}(\delta_\mathrm{y})$ abbreviating the constituent terms following

$$\mu_\alpha(\alpha,...) = \frac{Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref}}{J_\mathrm{yy}} \left( {}^{\mathrm{TF}}C_{\mathrm{My}}(\alpha,...) + \frac{{}^{\mathrm{B}}l_{\mathrm{TF}}}{l_\mathrm{ref}} {}^{\mathrm{TF}}C_{\mathrm{N}}(\alpha,...) \right)$$
$$\mu_{\delta\mathrm{y}}(\delta_\mathrm{y}) = \frac{{}^{\mathrm{S}}l_{\mathrm{CoM}} T}{J_\mathrm{yy}} \sin\delta_\mathrm{y}$$

akin to the definitions previously introduced by [3]. Here, however, the aerodynamic coefficients are non-linear functions of the angle of attack $\alpha$, the side-slip angle $\beta$, the free-stream velocity $V_\mathrm{A}$ and the altitude $h_\mathrm{alt}$ instead of linearised relations. Thus, the system coefficient $\mu_\alpha$ is also a function of these implicit states. From this point on, for ease of readability, the angle of attack $\alpha$ is the only depicted dependence of the aerodynamic coefficients and aerodynamic system coefficients, as it dominates their relation for the flight regime during the low-energy demonstrations. It is treated as a state given the fact that it implicitly depends on actual states following a geometric decomposition

$$\alpha \approx \Theta + \frac{w}{V_\mathrm{T}} - \frac{w_\mathrm{w}}{V_\mathrm{A}}$$

which can be done under the assumption of a small angle of attack. This goes to show that $\mu_\alpha(\alpha)$ is indeed a state-dependent term without any influence of the system input $\delta_\mathrm{y}$, logically retaining its place in the drift vector field $\boldsymbol{f}(\boldsymbol{x})$[1].

The vector field description in equation 3.32 is the basis for the derivation of the NDI linearisation and control loops for the pitch dynamics. It will be expanded to include translational drift dynamics in the following section.

### 3.3.3 Complete Non-linear Planar Dynamics

In order to derive the planar equations of motion including directional drift motion dynamics, the rigid-body translational kinematic and dynamic equations 3.8 and 3.9 are again decomposed into their vector constituents using the states in equations 3.12, resulting in

[1] Drift vector field refers here only to the non-linear system constituent $\boldsymbol{f}(\boldsymbol{x})$ in a general sense and not to the lateral or directional drift $y$ or $z$ of the vehicle.

$$
{}^{\mathrm{I}}\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix} = {}^{\mathrm{I}}\boldsymbol{T}_{\mathrm{B}}\ {}^{\mathrm{B}}\begin{pmatrix}u\\ v\\ w\end{pmatrix} \tag{3.33}
$$

$$
{}^{\mathrm{B}}\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} = \frac{1}{m}\ {}^{\mathrm{B}}\begin{pmatrix}\sum f_{\mathrm{x}}\\ \sum f_{\mathrm{y}}\\ \sum f_{\mathrm{z}}\end{pmatrix} - \begin{pmatrix}p\\ q\\ r\end{pmatrix} \times \begin{pmatrix}u\\ v\\ w\end{pmatrix} \tag{3.34}
$$

again reducing the number of displayed coordinate system indices to improve readability. The rules apply analogously, position variables being defined in the inertial frame and velocities and in turn forces being described in body coordinates. Using the EFUEN coordinate system as inertial reference again gives the transformation matrix from body to EFUEN coordinates ${}^{\mathrm{EFUEN}}\boldsymbol{T}_{\mathrm{B}}$ from equation 3.15. Extracting from this the planar dynamics in the pitch plane yields

$$
\begin{aligned}
\dot{x} = {} & \cos\Theta\sin\Psi u + (\sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi)v + \cdots \\
& + (\cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi)w
\end{aligned} \tag{3.35}
$$

$$
\dot{z} = -\sin\Theta u + \sin\Phi\cos\Theta v + \cos\Phi\cos\Theta w \tag{3.36}
$$

$$
\dot{u} = \frac{1}{m}\sum f_{\mathrm{x}} - qw + rv \tag{3.37}
$$

$$
\dot{w} = \frac{1}{m}\sum f_{\mathrm{z}} - pv + qu \tag{3.38}
$$

which still contains out-of-plane dependencies. Fully assuming planar motion removes terms containing the out-of-plane variables by setting $v = 0\,\frac{\mathrm{m}}{\mathrm{s}}$ and $\Psi = 0°$. Further adding the assumption of static roll motion, as for the derivation of the attitude dynamics, setting $\Phi = 0°$ and $p = 0°/\mathrm{s}$ reduces the equations to the substantially shorter form

$$
\dot{x} = \cos\Theta u + \sin\Theta w \tag{3.39}
$$

$$
\dot{z} = -\sin\Theta u + \cos\Theta w \tag{3.40}
$$

$$
\dot{u} = \frac{1}{m}\sum f_{\mathrm{x}} - qw \tag{3.41}
$$

$$
\dot{w} = \frac{1}{m}\sum f_{\mathrm{z}} + qu \tag{3.42}
$$

It can be seen that there is a coupling between the longitudinal and directional dynamics driven by their respective velocities $u$ and $w$. To avoid this coupled behaviour, a last assumption can be made. Assuming a gravity turn trajectory or, equivalently, a pitch angle of $\Theta = 0°$ and a pitch angular rate of $q = 0°/\mathrm{s}$ removes the coupled terms, leaving only

$$
\dot{x} = u \tag{3.43}
$$

$$
\dot{z} = w \tag{3.44}
$$

$$
\dot{u} = \frac{1}{m}\sum f_{\mathrm{x}} \tag{3.45}
$$

$$
\dot{w} = \frac{1}{m}\sum f_{\mathrm{z}} \tag{3.46}
$$

which holds for the low-energy demonstration flights of CALLISTO. During these flights, the vehicle always points straight up, making this a valid assumption. For trajectories that do not follow a gravity turn trajectory or a low-altitude hop trajectory, the coupling terms have to be considered [3].

The gravitational, aerodynamic and thrust components contribute to the sum of forces in equations 3.45 and 3.46 as shown in figure 3.3. These can be expressed in vector notation as

$$
\sum {}^{\mathrm{B}}\boldsymbol{f} = {}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{G}} + {}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{T}} + {}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{A}} \tag{3.47}
$$

**Gravitational Force**

Modelling the gravitational force ${}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{G}}$ can take a complex shape by considering a high-order approximation of the gravitational field of the planet. As the considered demonstration flights remain in a low-altitude domain, approximating the gravitational acceleration using the sea-level constant $g_0 = 9.806\frac{\mathrm{m}}{\mathrm{s}}$ yields sufficient accuracy. The resulting gravitational force vector is

$$
\begin{aligned}
{}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{G}} &= {}^{\mathrm{B}}\boldsymbol{T}_{\mathrm{I}} \overset{\mathrm{I}}{\begin{pmatrix} -mg \\ 0 \\ 0 \end{pmatrix}} && (3.48) \\
&= \overset{\mathrm{B}}{\begin{pmatrix} -mg\ \cos\Theta \\ 0 \\ -mg\ \sin\Theta \end{pmatrix}} && (3.49) \\
& && (3.50)
\end{aligned}
$$

which accounts for the attitude of the vehicle relative to the gravitational force vector in the pitch plane only, leading to the transformation matrix ${}^{\mathrm{B}}\boldsymbol{T}_{\mathrm{I}}$ simply being a rotation around the y-axis by the pitch angle $\Theta$.

**Propulsive Force**

Decomposing the propulsive force vector ${}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{T}}$ into its contributions along the axes in body coordinates yields

$$
{}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{T}} = \overset{\mathrm{B}}{\begin{pmatrix} T\ \cos\delta_{\mathrm{y}} \\ 0 \\ -T\ \sin\delta_{\mathrm{y}} \end{pmatrix}} \tag{3.51}
$$

with the main engine thrust magnitude $T$ and the TVC pitch deflection angle $\delta_{\mathrm{y}}$. Out-of-plane contributions from the yaw direction are neglected.

**Aerodynamic Force**

The description of the aerodynamic force vector follows the same logic applied in the derivation of the aerodynamic torques in section 3.3.2. It is described in the TF frame, and their description in the body frame thus requires a transformation. Section 3.3.2 built the relation

$$
{}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{A}} = Q_{\mathrm{dyn}} S_{\mathrm{ref}} {}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{TF}} \overset{\mathrm{TF}}{\begin{pmatrix} C_{\mathrm{A}} \\ C_{\mathrm{S}} \\ C_{\mathrm{N}} \end{pmatrix}} \tag{3.52}
$$

Inserting the simple transformation between TF and body frames that was previously introduced yields the description of the aerodynamic force vector in body coordinates

$$
{}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{A}} = Q_{\mathrm{dyn}} S_{\mathrm{ref}} \overset{\mathrm{TF}}{\begin{pmatrix} -C_{\mathrm{A}} \\ C_{\mathrm{S}} \\ -C_{\mathrm{N}} \end{pmatrix}} \tag{3.53}
$$

leaving the aerodynamic coefficients as functions of the angle of attack $\alpha$, the side-slip angle $\beta$, the free-stream velocity $V_{\mathrm{A}}$ and the altitude $h_{\mathrm{alt}}$, of which the angle of attack is the dominant factor in the pitch plane.

**Non-linear Planar EOMs**

The pitch dynamics continue to behave according to equations 3.30 and 3.31 derived in section 3.3.2. With the forces modelled according to the expressions above, the directional and longitudinal translational motion for the CALLISTO vehicle can be expressed by

$$\dot{x} = u \tag{3.54}$$
$$\dot{z} = w \tag{3.55}$$
$$\dot{u} = -g\ \cos(\Theta) - \frac{1}{m} Q_{\mathrm{dyn}} S_{\mathrm{ref}} C_{\mathrm{A}}(\alpha) + \frac{1}{m} T \cos(\delta_{\mathrm{y}}) \tag{3.56}$$
$$\dot{w} = -g\ \sin(\Theta) - \frac{1}{m} Q_{\mathrm{dyn}} S_{\mathrm{ref}} C_{\mathrm{N}}(\alpha) - \frac{1}{m} T \sin(\delta_{\mathrm{y}}) \tag{3.57}$$

Bringing these equations into the generic non-linear form using vector field notations and combining them with the corresponding attitude formulation leads to

$$\begin{bmatrix} \dot{\Theta} \\ \dot{q} \\ \dot{x} \\ \dot{z} \\ \dot{u} \\ \dot{w} \end{bmatrix} = \underbrace{\begin{bmatrix} q \\ \mu_{\alpha}(\alpha) \\ u \\ w \\ -\Gamma_{\mathrm{X}}(\Theta) - X_{\alpha}(\alpha) \\ -\Gamma_{\Theta}(\Theta) - Z_{\alpha}(\alpha) \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 \\ -\mu_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) \\ 0 \\ 0 \\ X_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) \\ -Z_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) \end{bmatrix}}_{\boldsymbol{g}(u)} \tag{3.58}$$

with the new coefficients

$$\begin{aligned} \Gamma_{\mathrm{X}}(\Theta) &= g\ \cos\Theta \\ \Gamma_{\Theta}(\Theta) &= g\ \sin\Theta \\ X_{\alpha}(\alpha) &= \frac{1}{m} Q_{\mathrm{dyn}} S_{\mathrm{ref}} C_{\mathrm{A}}(\alpha) \\ Z_{\alpha}(\alpha) &= \frac{1}{m} Q_{\mathrm{dyn}} S_{\mathrm{ref}} C_{\mathrm{N}}(\alpha) \\ X_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) &= \frac{1}{m} T \cos\delta_{\mathrm{y}} \\ Z_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) &= \frac{1}{m} T \sin\delta_{\mathrm{y}} \end{aligned}$$

where the aerodynamic coefficients $C_{\mathrm{i}}(\alpha, ...)$ and in turn again the aerodynamic system coefficients $X_{\alpha}(\alpha, ...)$ and $Z_{\alpha}(\alpha, ...)$ depend not just on the angle of attack $\alpha$, but also on the side-slip angle $\beta$, the free-stream velocity $V_{\mathrm{A}}$ and the altitude $h_{\mathrm{alt}}$. Again, for ease of readability, merely the dependence on the dominant contributor, the angle of attack, is displayed.

Further it should be noted that because this thesis is centred around the control of the system dynamics using the TVC actuators specifically, the longitudinal channel consisting of the kinematics $\dot{x}$ and the dynamics $\dot{u}$ can be omitted. The influence of the TVC control input $\delta_{\mathrm{y}}$ on this channel is negligible compared to the thrust modulation control input. This can be seen by comparing the magnitudes of corresponding terms in the B matrix in the EOMs considering the thrust as a system input [3]. The remaining two degrees-of-freedom in the pitch and directional drift dynamics will provide the basis for the derivation of the inversion law in the planar case in the following chapter.

### 3.3.4 Combined Pitch, Yaw, Lateral and Directional Dynamics

During the derivation of the pitch plane dynamics, the equations were fully decoupled from any out-of-plane influences. Thanks to this approach, expanding the system to four degrees-of-freedom and considering the yaw plane attitude dynamics as well as the lateral drift poses no challenge. Following the same derivation for the EOMs in the yaw plane yields similar results. Only a slight difference in the form of sign changes due to changed transformations between coordinate axes emerges. To remove out-of-plane coupling, here the pitch plane contributions $w = 0\,\frac{\mathrm{m}}{\mathrm{s}}$ and $\Theta = 0°$ are set to zero instead.

The contributing forces and torques remain unchanged, following the description given in equations 3.47 and 3.21, but the definition of the elements differs. Under the assumption of a fully planar flight and the omission of all out-of-plane terms, the adjusted gravitational and propulsive force vectors in the yaw plane become

$$ {}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{T}} = \begin{pmatrix} T\cos\delta_{\mathrm{z}} \\ -T\sin\delta_{\mathrm{z}} \\ 0 \end{pmatrix} \tag{3.59} $$

$$ {}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{G}} = \begin{pmatrix} -mgcos\Psi \\ mg\sin\Psi \\ 0 \end{pmatrix} \tag{3.60} $$

with the aerodynamic force vector ${}^{\mathrm{B}}\boldsymbol{f}_{\mathrm{A}}$ from equation 3.53 still being applicable as it was not derived using the exclusively planar simplification. Using the adjusted definition of the propulsive force in equation 3.59 and applying the cross product with the relative position of the body frame origin with respect to the structural frame origin results in the propulsive torque

$$ {}^{\mathrm{B}}\boldsymbol{\tau}_{\mathrm{T}} = \begin{pmatrix} 0 \\ 0 \\ -{}^{\mathrm{S}}l_{\mathrm{CoG}}T\sin\delta_{\mathrm{z}} \end{pmatrix} \tag{3.61} $$

Again, the definition of the aerodynamic torques ${}^{\mathrm{B}}\boldsymbol{\tau}_{\mathrm{A}}$ given in equation 3.29 remains valid. The reasoning for the influence of the RCS torques ${}^{\mathrm{B}}\boldsymbol{\tau}_{\mathrm{RCS}}$ equally remains unchanged, making their consideration unnecessary.

These changes can be inserted into the vector fields of equation 3.58 and, if the longitudinal dynamics are disregarded, because of the limited influence of the TVC system's control inputs $\delta_{\mathrm{y}}$ and $\delta_{\mathrm{z}}$, a combined formulation can be constructed which gives the resulting non-linear system in vector field notation

$$ \begin{bmatrix} \dot{\Theta} \\ \dot{\Psi} \\ \dot{q} \\ \dot{r} \\ \dot{y} \\ \dot{z} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \underbrace{\begin{bmatrix} q \\ r \\ \mu_{\alpha}(\alpha) \\ \nu_{\beta}(\beta) \\ w \\ v \\ \Gamma(\Psi) + Y_{\beta}(\beta) \\ -\Gamma(\Theta) - Z_{\alpha}(\alpha) \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -\mu_{\delta y}(\delta_{\mathrm{y}}) \\ -\nu_{\delta z}(\delta_z) \\ 0 \\ 0 \\ -Y_{\delta_z}(\delta_z) \\ -Z_{\delta_{\mathrm{y}}}(\delta_{\mathrm{y}}) \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})} \tag{3.62} $$

with the new yaw plane coefficients $\nu_{\beta}$, $\nu_{\delta_{\mathrm{z}}}$, $Y_{\beta}$ and $Y_{\delta_{\mathrm{z}}}$, which are defined as

$$ \nu_{\beta}(\beta) = \frac{Q_{\mathrm{dyn}}S_{\mathrm{ref}}l_{\mathrm{ref}}}{J_{\mathrm{zz}}}\left(-{}^{\mathrm{TF}}C_{\mathrm{Mz}}(\beta) + \frac{{}^{\mathrm{B}}l_{\mathrm{TF}}\,{}^{\mathrm{TF}}C_{\mathrm{S}}(\beta)}{l_{\mathrm{ref}}}\right) \tag{3.63} $$

$$ \nu_{\delta_{\mathrm{z}}}(\delta_{\mathrm{z}}) = \frac{{}^{\mathrm{S}}l_{\mathrm{CoM}}T}{J_{\mathrm{zz}}}\sin\delta_{\mathrm{z}} \tag{3.64} $$

$$ Y_{\beta}(\beta) = \frac{Q_{\mathrm{dyn}}S_{\mathrm{ref}}\,{}^{\mathrm{TF}}C_{\mathrm{S}}(\beta)}{m} \tag{3.65} $$

$$ Y_{\delta_{\mathrm{z}}}(\delta_{\mathrm{z}}) = \frac{T}{m}\sin\delta_{\mathrm{z}} \tag{3.66} $$

In the yaw plane, the dominant contributor to the aerodynamic coefficients is no longer the angle of attack. Instead, the side-slip angle is the only displayed dependence. This set of equations carries the same non-linearities seen in the pitch plane: in the aerodynamic coefficients, as implicitly state-dependent non-linear functions, and in the control input coefficients, which contain trigonometric functions depending on the TVC deflection angle. Similar to the angle of attack, the side-slip angle can be approximated as a function of the yaw angle $\Psi$, the lateral drift velocity $v$ and the wind velocity $v_{\mathrm{w}}$ following

$$ \beta \approx -\Psi + \frac{v}{V_{\mathrm{T}}} - \frac{v_{\mathrm{w}}}{V_{\mathrm{A}}} \tag{3.67} $$

when small angles are assumed, making the implicit state dependence given in equations 3.63 and 3.65 an explicit one if desired.

This last set of non-linear equations completes the system model required to understand and design a control architecture for the TVC system. It encompasses four of the six degrees-of-freedom of the CALLISTO vehicle motion, not describing the longitudinal dynamics in both translation and attitude, as these remain almost unaffected by the TVC control action.

### 3.3.5 Model Validity

Because a model is merely an approximation of the real behaviour of a physical system and FBL-based control architectures are especially prone to modelling mismatches, this section will highlight the most prominent simplifications and assumptions made during the derivation of the rigid-body, non-linear CALLISTO EOMs.

The basis of the derivation is already a strong simplification. By assuming a rigid body, changes in shape of the vehicle are assumed non-existent [44]. This neglects structural deformations that would introduce high-frequency system dynamics like structural vibrations. Without the description of these effects, no information about their dynamic nature is contained within the system model. Thus, any control system derived from it cannot consider these effects, and the produced control action could accidentally drive resonance in the structural oscillations, which ultimately could lead to catastrophic system failure. In practice, the structural oscillations and their natural frequency are determined in Finite Element Method (FEM) simulations. Upon designing the control system, correct rigid-body bandwidth placement away from the identified resonant frequencies is a design driver. For the CALLISTO vehicle, the structural vibration frequencies are well known from numerical simulations and sufficient closed-loop bandwidth separation is ensured [47]. Another consequence arising from the rigid-body assumption is the neglect of the so-called tail-wag-dog effect. This effect describes the relative motion of the main engine to the body of the launch vehicle and induced torques and changes in Moment of Inertia (MoI) and CoM. Propellant sloshing also remains unconsidered in this study. If all these effects were considered, it is foreseeable that multi-body description applicable to CALLISTO vehicle would drastically increase the amount of states in the non-linear equations of motion. This would in turn require measurements of these additional states to account for them in the linearising state feedback matrix $\boldsymbol{a}_{\mathrm{v}}(\boldsymbol{x})$. This raises the question whether the performance increase obtained by the consideration of such higher-order effects even outweighs the increase in model complexity or whether simpler models are sufficient.

Beyond the rigid-body simplification, several vehicle and mission-specific assumptions are made. Starting with the perfect vertical flight scenario. Assuming small angles for the pitch and yaw angles and small magnitudes for the corresponding rates with respect to the reference trajectory restricts the proposed solution to flight domains where this holds. Besides the hover scenarios described in this work, this also covers a gravity turn manoeuvre for which the lateral and directional centrifugal forces cancel with the gravitational force components. Further, the cross-couplings between control channels are neglected, simplifying a key part of RLV dynamics. This is done by assuming that the roll channel control solution is perfect and that all products of inertia are negligible due to geometric symmetries of the vehicle.

Other plant dynamics, such as actuator dynamics and sensor delays also remain unconsidered. Since the plant aerodynamic model is built around look-up tables, flight domains with large uncertainties like transonic flight and vehicle configuration transitions require well-bounded envelopes by the aerodynamics.

All of these restrictions have to be carefully considered when deriving the inversion law. Outside the validity region of the plant model, it will no longer accurately describe the system, and any linearisation and control solution applied to it will have decreased performance.

# 4 NDI Based Controller Design

Based on the methods presented in chapter 2 and the CALLISTO system dynamics derived in chapter 3, the feedback linearisation-based control architecture can be derived. This will allow the tracking of guidance commands without restrictions to any predetermined trajectory, as would be the case for gain-scheduled solutions. An NDI-based control architecture is separated into the linearisation loop and the outer error controller, and this chapter presents their derivation and implementation applied to an increasing number of DOFs of the CALLISTO system dynamics.

## 4.1 Control Strategy

To begin, an NDI-based control structure is first derived only for the planar attitude dynamics of the CALLISTO vehicle in the pitch plane described in subsection 3.3.2. Upon expanding the implementation to also consider directional drift, the system architecture becomes a cascaded loop structure, as is customary in aeronautical applications [13, 21, 36, 48]. This technique's applicability depends on the separation of the bandwidths of the cascaded closed-loops. The outer loop is designed such that it assumes perfect tracking of the inner loop with respect to its given reference value, making the dynamics of the inner loop invisible to the outer loop. This allows both loops to be designed independently. For the CALLISTO system dynamics in one plane, this results in two SISO systems, keeping the design approach presented in chapter 2 applicable while avoiding design complexities resulting from a MIMO system. To guarantee that this approximation is valid and stability is retained, sufficient time-scale separation is required [20]. Time-scale separation is a tested principle in NDI-based control solutions with a wide range of examples [21, 27, 41, 48].

Figure 4.1 depicts the desired structure for the full four TVC controllable DOFs. Pitch and yaw plane system dynamics are controlled via a parallel, cascaded architecture that features a controller for each degree of freedom and their derivatives, corresponding to states in the chosen form of the plant. Each of these controllers is shown as a subsystem that contains its own linearisation loop and an error controller.

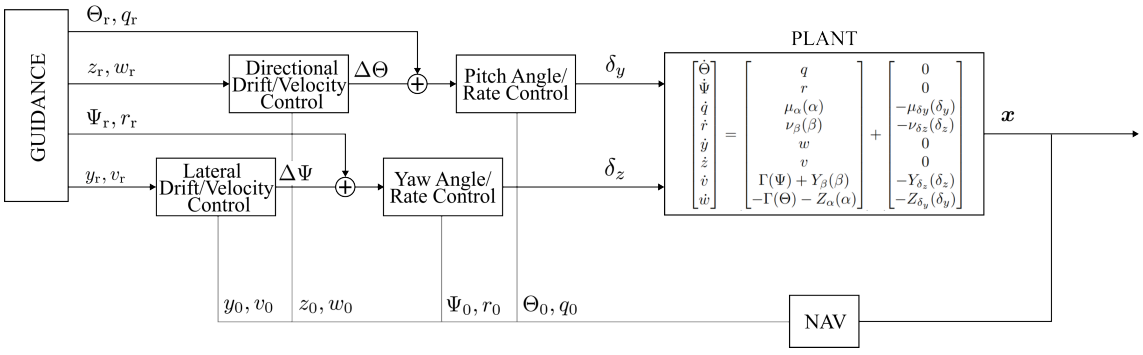



Figure 4.1: Full four-degrees-of-freedom cascaded control architecture.

The guidance system is treated as a pre-existing unit which supplies the optimal reference signal $\boldsymbol{x}_r$ for a given trajectory. In case of the inner attitude loops, these reference signals are combined with the outer loop correction values $\Delta\boldsymbol{\epsilon}$ to generate a new reference that is later used to calculate error signals $\boldsymbol{x}_e$. Finally, it is assumed that the navigation subsystem provides measurements of all states $\boldsymbol{x}_0$ to the error controllers. They construct the state error $\boldsymbol{x}_e$ for each of the states internally

and use it to derive the corrective control action. Both the guidance and navigation subsystems are assumed to be solved, and their structure is not discussed further in this work. What each of the control blocks in figure 4.1 contains specifically will be elaborated in the following sections, starting with the inner loops, which control the attitude dynamics.

## 4.2 Planar Attitude Control

### 4.2.1 Inner Linearisation Loop

At the heart of the linearisation loop, which provides the pseudo LTI to which the linear error controller is later applied, lies the inversion law. It is generally described for input-affine, SISO systems by equation 2.7. To apply it to the CALLISTO system dynamics, the non-linear, planar attitude EOMs from equation 3.32 are first transformed into an input-affine form. This can be achieved by assuming small values for the TVC deflection angles $\delta_\mathrm{y}$ and $\delta_\mathrm{z}$, which is valid as the maximum range of motion of the TVC gimbal actuators is restricted to this domain.

The resulting set of differential equations in vector field notation becomes

$$\begin{bmatrix} \dot{\Theta} \\ \dot{q} \end{bmatrix} = \underbrace{\begin{bmatrix} q \\ \mu_\alpha(\alpha) \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 \\ -\mu_{\delta_\mathrm{y}} \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x})} \delta_\mathrm{y} \tag{4.1}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ q \end{bmatrix}}_{h(\boldsymbol{x})} \tag{4.2}$$

with the drift vector field $\boldsymbol{f}(\boldsymbol{x})$, the input vector field $\boldsymbol{g}(\boldsymbol{x})$ and the output function $h(\boldsymbol{x})$. The choice of the output function and its contained control variables is free, but as stated in chapter 2, this choice can greatly influence the behaviour of the internal dynamics, if any remain after feedback linearisation. Here, the pitch angle $\Theta$ is chosen as the system output to facilitate the implementation of a PD error controller in later design phases. This enables the use of system heritage from previous control architectures for the CALLISTO vehicle.

Before moving to the derivation of the linearisation loop, it is worth taking a moment to characterise the non-linear system presented above. It contains the system coefficients $\mu_\alpha(\alpha)$ and $\mu_{\delta_\mathrm{y}}$. As presented in chapter 3, the aerodynamic system coefficient $\mu_\alpha$ depends on a number of variables, but since the contribution of the angle of attack $\alpha$ dominates, only this one is displayed. The angle of attack implicitly depends on the chain of rotations associated with the pitch angle $\Theta$, the flight-path angle $\gamma$ and the aerodynamic angle of attack $\alpha_\mathrm{w}$ in non-linear form. Populating $C_{\mathrm{M}_\mathrm{y}}$ and $C_\mathrm{N}$ within $\mu_\alpha$ can follow one of two approaches: either it can be obtained via a Taylor expansion, which is reduced to only consider dominant terms, yielding a term containing the stability derivative times angle of attack. The decomposition of the angle of attack in equation 3.33 can then be inserted to obtain a state-dependent form. The other option is to interpolate the aerodynamic coefficients within $\mu_\alpha$ from an aerodynamic database. The latter is chosen in the implementation in this work to circumvent the approximation made in the Taylor expansion. Returning to the non-linear set of equations above, the drift vector field $f(\boldsymbol{x})$ is implicitly a state-dependent vector field, thereby making its depiction as a function of states accurate. The input vector field $g(\boldsymbol{x})$, on the other hand, does not contain a dependence on the system states $\Theta$ and $q$, yet it is built from extended state measurements of the main engine thrust $T(t)$ and the mass $m(t)$ of the vehicle, making it a time-varying function instead. It will continue to be described as $g(\boldsymbol{x})$, but the fact that it only depends on extended states should be kept in mind.

Now, to begin deriving the NDI law, the relative degree of the system is required. Repeated differentiation of the output function $h(\boldsymbol{x})$ using the Lie derivatives may accomplish this feat. Applying the process described in chapter 2 yields

$$\dot{y} = \mathcal{L}_{\boldsymbol{f}}\Theta + \mathcal{L}_{\boldsymbol{g}}\Theta\delta_\mathrm{y} \tag{4.3}$$

with the contributing terms

$$\mathcal{L}_{\boldsymbol{f}}\Theta = \nabla\Theta\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial\Theta}{\partial\Theta} & \frac{\partial\Theta}{\partial q} \end{pmatrix} \begin{bmatrix} q \\ \mu_\alpha(\alpha) \end{bmatrix} = q$$

and

$$\mathcal{L}_{\boldsymbol{g}}\Theta\delta_{\mathrm{y}} = \nabla\Theta\boldsymbol{g}(\boldsymbol{x})\delta_{\mathrm{y}} = \begin{pmatrix} \frac{\partial\Theta}{\partial\Theta} & \frac{\partial\Theta}{\partial q} \end{pmatrix} \begin{bmatrix} 0 \\ -\mu_{\delta_{\mathrm{y}}} \end{bmatrix} = 0$$

Because the second term remains zero, no relation between the system output equation $y$ and the control input $\delta_{\mathrm{y}}$ is found. Thus, the relative degree of the system has to be larger than one, leading to a repetition of the derivation according to

$$\ddot{y} = \mathcal{L}_{\boldsymbol{f}}q + \mathcal{L}_{\boldsymbol{g}}q\delta_{\mathrm{y}} \tag{4.4}$$
$$= \mu_\alpha(\alpha) - \mu_{\delta_{\mathrm{y}}}\delta_{\mathrm{y}} \tag{4.5}$$

which corresponds exactly to the planar attitude dynamics in the second row of the non-linear system 4.1. This result contains the desired relation between the system's input and output, leading to a relative degree of $\rho = 2$. This result depends on the choice of the output function $h(\boldsymbol{x})$: if, for instance, the angular velocity is chosen as the system output instead, this relative degree would reduce to $\rho = 1$. Following the derivation with this alternative approach obtains a very similar result to the presented variant. Consequently, it is not discussed in further detail.

Since the relative degree is not one, the NDI law from equation 2.26 is not applicable and the derivation has to default to the more general inversion law from equation 2.7. If applied to the non-linear system at hand, one obtains

$$\delta_{\mathrm{y}} = \underbrace{\frac{1}{-\mu_{\delta_{\mathrm{y}}}}}_{B_{\mathrm{v}}(\boldsymbol{x})^{-1}} (v - \underbrace{\mu_\alpha(\alpha)}_{a_{\mathrm{v}}(\boldsymbol{x})}) \tag{4.6}$$

where the control effectiveness matrix $\boldsymbol{B}_{\mathrm{v}}(\boldsymbol{x})$ and the feedback linearisation matrix $\boldsymbol{a}_{\mathrm{v}}(\boldsymbol{x})$ reduce to scalars. This is always the case for SISO systems. Instead of requiring a non-singular $\boldsymbol{B}_{\mathrm{v}}(\boldsymbol{x})$ matrix, such systems have to adhere to the restriction that the value of $B_{\mathrm{v}}(\boldsymbol{x})$ is always non-zero for it to remain invertible. Equation 4.6 further introduces the new pseudo control input with the definition $v = \ddot{y}$. Its relation to the system output becomes apparent upon comparing the inversion law and the final derivative of the output equation in 4.5.

When the inversion law is inserted into the non-linear system equations in 4.1, the following relation emerges

$$\begin{bmatrix} \dot{\Theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} q \\ \mu_\alpha(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_{\delta y} \end{bmatrix} \frac{1}{-\mu_{\delta_{\mathrm{y}}}} (v - \mu_\alpha(\alpha)) \tag{4.7}$$
$$= \begin{bmatrix} q \\ v \end{bmatrix} \tag{4.8}$$

which corresponds again to a chain of $\rho = 2$ integrators between the new pseudo control input $v$ and the system output $y = \Theta$. The new pseudo LTI can be rewritten in state-space form as

$$\begin{bmatrix} \dot{\Theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \tag{4.9}$$

Figure 4.2 depicts both the linearisation loop with its feedback linearisation term $a_{\mathrm{v}}(\boldsymbol{x})$ and control effectiveness $B_{\mathrm{v}}(\boldsymbol{x})$ as well as the equivalent double integrator system as seen from the perspective of the new pseudo control input $v$. The system coefficients that build the control effectiveness and feedback linearisation terms are provided by state measurements given by the navigation solution.

This simple system can be used as a baseline onto which any linear control strategy can be applied. To demonstrate its potential, a pole-placement PD controller is presented in the following section.

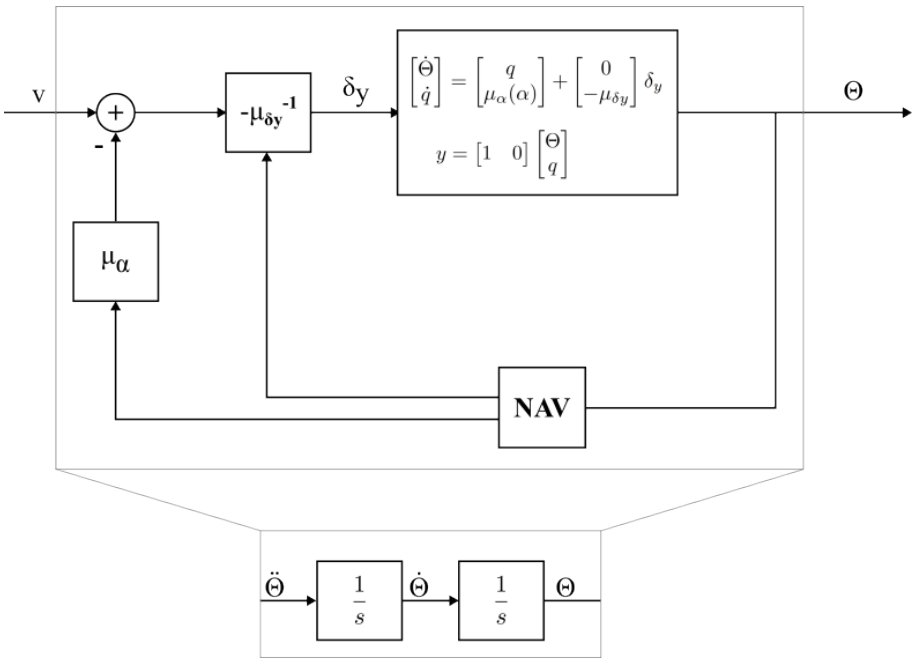



Figure 4.2: Planar attitude NDI linearisation loop with equivalent double integrator representation.

### 4.2.2 Inner Control Loop

Section 2.2 describes how to design the error control loop around a generic, feedback linearised, non-linear system. Applying this fundamental approach to the result obtained above follows the same logic, starting from the pitch angle tracking error

$$\Theta_\mathrm{e} = \Theta_\mathrm{r} - \Theta_0 \tag{4.10}$$

where the index 0 denotes a measurement of the system output $y$ provided by the navigation subsystem. To arrive at a relation with the pseudo control input $v = \ddot{y}$, equation 4.10 is differentiated twice according to the relative degree $\rho = 2$, leading to

$$\ddot{\Theta}_\mathrm{e} = \ddot{\Theta}_\mathrm{r} - \ddot{\Theta}_0 \tag{4.11}$$

where the error dynamics $\ddot{\Theta}_\mathrm{e}$ govern the evolution of the error over time and the second derivative of the measurement of the current system output is assumed to be equal to the second derivative of the true output $\ddot{\Theta}$. As for the generic example in section 2.2, the goal is to achieve an asymptotically stable error with a steady-state value of zero. The error dynamics

$$\ddot{\Theta}_\mathrm{e} = K_\mathrm{p}\Theta_\mathrm{e} + K_\mathrm{d}\dot{\Theta}_\mathrm{e} \tag{4.12}$$

can fulfil this requirement given that the control gains $K_\mathrm{p}$ and $K_\mathrm{d}$ are chosen accordingly. Inserting the error dynamics into equation 4.11 and reforming it for the system output yields

$$v = \ddot{\Theta} = \ddot{\Theta}_\mathrm{r} - K_\mathrm{p}\Theta_\mathrm{e} - K_\mathrm{d}\dot{\Theta}_\mathrm{e} \tag{4.13}$$

With this relation, the complete NDI-based control architecture can be built akin to the generic structure given in figure 2.3. The resulting architecture is depicted in figure 4.3. When compared to a typical PD control implementation, the only difference is the feed-forward signal $\ddot{\Theta}_\mathrm{r}$. It is sometimes referred to as the *reference pseudo control* and as seen in equation 4.13, together with the stabilising feedback loop, it plays a crucial role in the construction of the pseudo control input [26]. The depicted architecture only considers measurements of the angle in the navigation subsystem, but if measurements of the angular velocity are available, the derivative error can alternatively be

constructed from these measurements and a reference value provided by the guidance subsystem. This avoids the use of the derivative of the $\Theta_e$ signal shown in figure 4.3, which would amplify noise and might therefore require filtering, which in turn can introduce additional dynamics.

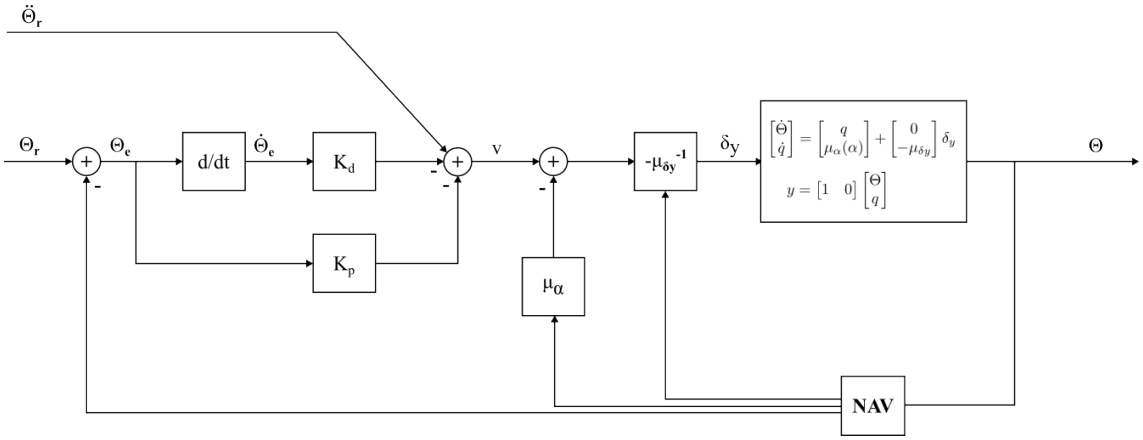



Figure 4.3: Pitch plane stabilising feedback loop applied to the planar attitude NDI linearisation loop.

With this control architecture in mind, the problem of choosing the control gains can be approached. Finding not just stabilising but optimal control gains is a broad field of research and can be influenced by a number of considerations. For this investigation, the actuator natural frequency $\omega_{\mathrm{n,TVC}}$ and the time-scale separation $\sigma$ are assumed as design factors. As the current linear control strategy of the CALLISTO mission considers time-scale separation of about one order of magnitude [49], a value of $\sigma = \frac{1}{10}$ will be used in the design of the NDI-based controllers. Together with the known architecture, the TVC natural frequency and time-scale separation magnitude provide the necessary information to tune the control gains to follow any desired dynamic behaviour. Here, the transfer behaviour from reference to output is approximated by the dynamics of a second-order system, which can be described by the transfer function

$$T(s) = \frac{y(s)}{r(s)} \approx \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2} \tag{4.14}$$

with the system's natural frequency $\omega_{\mathrm{n}}$ and the damping ratio $\zeta$. Introducing the time-scale separation requirement based on the natural frequency of the TVC subsystem results in

$$T(s) = \frac{y(s)}{r(s)} \approx \frac{(\sigma\,\omega_{\mathrm{n,TVC}})^2}{s^2 + 2\,\zeta\,\sigma\,\omega_{\mathrm{n,TVC}}\,s + (\sigma\,\omega_{\mathrm{n,TVC}})^2} \tag{4.15}$$

where the damping ratio is chosen to be $\zeta = \frac{1}{\sqrt{2}} \approx 0.7071$ to obtain a trade-off between resonant peak magnitude and settling time [50].

Now a relationship between the error control gains and the desired dynamic behaviour is required to correctly place the poles of the system. To this end, consider again the block diagram in figure 4.3. The transfer behaviour from the reference value $\Theta_{\mathrm{r}}$ to the system output $\Theta$ can be described by investigating the signals constituents in the time domain and transferring them to the frequency domain by means of Laplace transformation. Because the linearisation loop transforms the non-linear plant dynamics into a chain of integrators, which is a linear system, the description with transfer functions applies. Decomposing the pseudo control input in the time domain leads to

$$\ddot{\Theta} = \ddot{\Theta}_{\mathrm{r}} - \ddot{\Theta}_{\mathrm{e}} \tag{4.16}$$

$$= \ddot{\Theta}_{\mathrm{r}} - K_{\mathrm{p}}\,\Theta_{\mathrm{e}} - K_{\mathrm{d}}\dot{\Theta}_{\mathrm{e}} \tag{4.17}$$

$$= \ddot{\Theta}_{\mathrm{r}} - K_{\mathrm{p}}\,(\Theta_{\mathrm{r}} - \Theta) - K_{\mathrm{d}}\,(\dot{\Theta}_{\mathrm{r}} - \dot{\Theta}) \tag{4.18}$$

which can be simplified if the reference values for the angular rate and acceleration $\dot{\Theta}_\mathrm{r}$ and $\ddot{\Theta}_\mathrm{r}$ are assumed to be zero. The resulting reduced variant is transformed into the frequency domain using the Laplace transformation

$$\ddot{\Theta} = -K_\mathrm{p}\,(\Theta_\mathrm{r} - \Theta) + K_\mathrm{d}\,\dot{\Theta} \tag{4.19}$$

$$\downarrow \mathcal{L}\{\cdot\}$$

$$s^2\Theta = -K_\mathrm{p}\,\Theta_\mathrm{r} + K_\mathrm{p}\,\Theta + K_\mathrm{d}\,s\Theta \tag{4.20}$$

Reforming this relation to display the transfer behaviour from reference $\Theta_\mathrm{r}$ to system output $\Theta$ yields the closed-loop transfer function

$$T(s) = \frac{\Theta}{\Theta_\mathrm{r}} = \frac{-K_\mathrm{p}}{s^2 - K_\mathrm{d}\,s - K_\mathrm{p}} \tag{4.21}$$

By means of comparison with equation 4.15, a description of the control gains as functions of the driving design variables $\omega_{\mathrm{n,TVC}}$ and $\sigma$ can be found according to

$$K_{\mathrm{p,att}} = -(\sigma\,\omega_{\mathrm{n,TVC}})^2 \tag{4.22}$$

$$K_{\mathrm{d,att}} = -2\,\zeta\,\sigma\,\omega_{\mathrm{n,TVC}} \tag{4.23}$$

This approach yields the magnitudes of the control gains, but the signs of the current solution are a mere result of the choice of signs in the definition of the error dynamics. If it is considered that the linearisation loop always yields a positive double integrator, the chosen signs yield asymptotic error dynamics.

## 4.3 Inclusion of Directional Drift Control

With the planar attitude control structure and the control strategy presented at the beginning of this chapter in mind, an outer loop can be designed such that it controls the directional drift $z$. To begin, the plant is expanded by the two additional states $z$ and $w$ and their corresponding kinematic and dynamic behaviour, which was previously presented in equation 3.58. Following the same argumentation as during the derivation of the overall system dynamics in section 3.3.3, the longitudinal kinematics $\dot{x}$ and dynamics $\dot{u}$ are neglected. As the remaining system is again not affine in input, the small-angle approximation for the TVC deflection angle $\delta_\mathrm{y}$ is used once more to transform it into the form

$$\begin{bmatrix} \dot{\Theta} \\ \dot{q} \\ \dot{z} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} q \\ \mu_\alpha(\alpha) \\ w \\ -\Gamma_\Theta(\Theta) - Z_\alpha(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_{\delta y} \\ 0 \\ -Z_{\delta_y} \end{bmatrix} \delta_\mathrm{y} \tag{4.24}$$

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ q \\ z \\ w \end{bmatrix} \tag{4.25}$$

which is a MIMO system. This is where the cascaded control strategy reveals its greatest advantage. To retain the ability to follow the SISO derivations presented in chapter 2 and to avoid complications from dealing with the MIMO system structure, an equivalent transformation enables the separated treatment of translational and attitude dynamics. By finding an expression for the control input $\delta_\mathrm{y}$ that depends on attitude states only and inserting it into the last row of the system presented in equation 4.24, the attitude and translational plants can be separated into a cascaded structure. Figure 4.4 shows the plant decomposition.

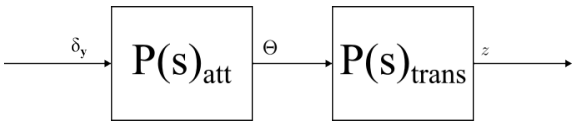



Figure 4.4: Cascaded plant structure using the pitch plane as an example.

Then, the attitude plant remains as shown at the beginning of this chapter, repeated here for completeness as

$$\begin{bmatrix}\dot{\Theta}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} q \\ \mu_\alpha(\alpha)\end{bmatrix} + \begin{bmatrix} 0 \\ -\mu_{\delta y}\end{bmatrix}\delta_\mathrm{y} \tag{4.1}$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\Theta\\ q\end{bmatrix} \tag{4.2}$$

while the directional drift plant takes the form

$$\begin{bmatrix}\dot{z}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} w \\ -\Gamma_\Theta(\Theta) - Z_\alpha(\alpha)\end{bmatrix} + \begin{bmatrix} 0 \\ -Z_{\delta_y}\end{bmatrix}\delta_\mathrm{y}(\Theta, q) \tag{4.26}$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}z\\ w\end{bmatrix} \tag{4.27}$$

with their respective system and output equations. They will subsequently be used to derive the inversion law for the outer loop. First, however, the function $\delta_\mathrm{y}(\Theta, q)$ must be found. Consider the inversion law of the inner loop given in equation 4.6. It gives the control input as a function of the system coefficients $\delta_\mathrm{y}(\mu_{\delta_\mathrm{y}}, \mu_\alpha)$, of which the aerodynamic system coefficient $\mu_\alpha(\alpha)$ implicitly depends on system states through the angle of attack $\alpha$ and the thrust system coefficient $\mu_{\delta_\mathrm{y}}$ only depends on extended states like the thrust. Therefore, the inversion law represents a function of the control input $\delta_\mathrm{y}$ depending only on attitude system states. This can be inserted into the translational plant given in equation 4.26 to obtain

$$\begin{bmatrix}\dot{z}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} w \\ -\Gamma_\Theta(\Theta) - Z_\alpha(\alpha)\end{bmatrix} + \begin{bmatrix} 0 \\ -Z_{\delta_y}\end{bmatrix}\left(\frac{1}{-\mu_{\delta_\mathrm{y}}}(v_\mathrm{att} - \mu_\alpha(\alpha))\right) \tag{4.28}$$

where the gravitational contribution in the drift vector field $f(\boldsymbol{x})$ has the shape $\Gamma_\Theta = g\sin\Theta$. Extracting the dependence on the pitch angle $\Theta$ in this system coefficient using the small-angle approximation and reforming for the new input $\Theta$ yields

$$\begin{bmatrix}\dot{z}\\ \dot{w}\end{bmatrix} = \underbrace{\begin{bmatrix} w \\ -Z_\alpha(\alpha) - \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}}\mu_\alpha(\alpha) + \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}}v_\mathrm{att}\end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 \\ -\Gamma_\Theta\end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x})}\Delta\Theta \tag{4.29}$$

$$y = \underbrace{\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}z\\ w\end{bmatrix}}_{h(\boldsymbol{x})} \tag{4.30}$$

Now that the dependence on the system input $\delta_\mathrm{y}$ is removed, a non-linear, input-affine system in vector field notation is obtained that can be used to derive an inversion law akin to the form found in section 4.2 for the pitch dynamics.

### 4.3.1 Outer Linearisation Loop

As previously presented, the first step is to find the relative degree $\rho$ of the system. Differentiating the output equation twice results in the expression

$$\ddot{y} = -\Gamma_\Theta\Delta\Theta - Z_\alpha(\alpha) - Z_{\delta_y}\left(\frac{1}{-\mu_{\delta_\mathrm{y}}}(v_\mathrm{att} - \mu_\alpha(\alpha))\right) \tag{4.31}$$

which does not depend on the original control input $\delta_\mathrm{y}$. Yet, because of the transformation of the plant into two nested structures, the outer loop no longer supplies a TVC deflection angle. It must instead supply the required pitch angle $\Delta\Theta$ to drive the drift coordinate $z$ to the reference $z_\mathrm{r}$. This can be seen when the overall control architecture shown in figure 4.1 is considered again. At the interface between inner and outer control loop, the attitude reference value $\Theta_\mathrm{r}$ supplied by the guidance subsystem is added to the outer loop attitude reference $\Delta\Theta$, which becomes the new reference. This way, the inner loop considers both contributing effects and constructs the control error $\Theta_\mathrm{e}$ accordingly. Both contributions propagate to the actual system control input $\delta_\mathrm{y}$.

To determine the relative degree, merely a dependence on the outer loop system output $\Delta\Theta$ must be found. Equation 4.31 already contains such a relation, leading to the system relative degree $\rho = 2$. Having found this property, the inversion law can be defined using the generic formulation in equation 2.7, which results in

$$\Delta\Theta = \underbrace{-\frac{1}{\Gamma_\Theta}}_{B_\mathrm{v}(\boldsymbol{x})^{-1}} \left( v_\mathrm{trans} - \underbrace{\left( -Z_\alpha(\alpha) - \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}}\mu_\alpha(\alpha) + \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}} v_\mathrm{att} \right)}_{a_\mathrm{v}(\boldsymbol{x})} \right) \tag{4.32}$$

To assess the resulting pseudo LTI, the inversion law is inserted into the underlying system equation 4.29 which leads to

$$\begin{aligned} \begin{bmatrix} \dot{z} \\ \dot{w} \end{bmatrix} = & \underbrace{\begin{bmatrix} w \\ -Z_\alpha(\alpha) - \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}}\mu_\alpha(\alpha) + \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}} v_\mathrm{att} \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{x})} + \cdots \\ & + \underbrace{\begin{bmatrix} 0 \\ -\Gamma_\Theta \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x})} \underbrace{-\frac{1}{\Gamma_\Theta}}_{B_\mathrm{v}(\boldsymbol{x})^{-1}} \left( v_\mathrm{trans} - \underbrace{\left( -Z_\alpha(\alpha) - \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}}\mu_\alpha(\alpha) + \frac{Z_{\delta_y}}{\mu_{\delta_\mathrm{y}}} v_\mathrm{att} \right)}_{a_\mathrm{v}(\boldsymbol{x})} \right) \end{aligned} \tag{4.33}$$

The resulting simplified system has the same shape as for the inner loop. It is given as

$$\begin{bmatrix} \dot{z} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} w \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ v_\mathrm{trans} \end{bmatrix} \tag{4.34}$$

which describes a double integrator transfer behaviour from the outer pseudo control input $v_\mathrm{trans}$ to the directional drift $z$. In state-space form the linearised system can be described as

$$\begin{bmatrix} \dot{z} \\ \dot{w} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{A}} \begin{bmatrix} z \\ w \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\boldsymbol{B}} v_\mathrm{trans} \tag{4.35}$$

To summarise the outer linearisation loop structure, figure 4.5 depicts its application to the already stabilised inner loop from section 4.2. It features the same signal sources as the inner loop: extended state measurements from the navigation subsystem for the control effectiveness and linearising feedback terms. As in the control strategy described at the beginning of this chapter, the reference signal for the inner loop $\Theta_\mathrm{r,new}$ is constructed from a superposition of the guidance attitude reference $\Theta_\mathrm{r}$ and the outer loop corrective contribution $\Delta\Theta$.

As long as the closed-loop bandwidth of the inner loop is correctly separated from the closed-loop bandwidth of the outer loop via time-scale separation, this structure is valid. From the perspective of the slower outer loop, the inner loop can be seen as providing immediate commands, reaching its equilibrium much faster than the outer loop.

Further of note is the feedback of the inner pseudo control input $v_\mathrm{att}$, which is a constituent of the linearising feedback term $a_\mathrm{v}(\boldsymbol{x})$ of the outer loop. Due to time-scale separation, this signal contains higher-frequency dynamics at the bandwidth of the inner loop. The outer loop cannot

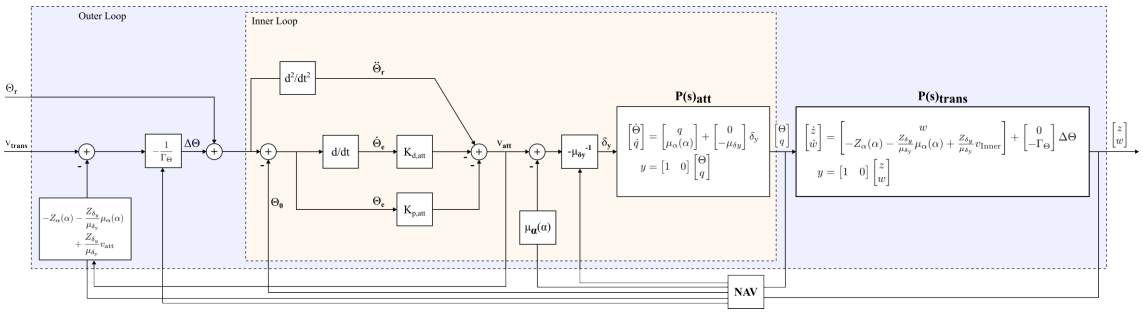



Figure 4.5: Outer NDI linearising feedback loop with the inner stabilised attitude dynamics and nested plants.

follow these fast dynamics, effectively making them noise on the signal. To alleviate this problem, a first-order low-pass filter at the natural frequency of the outer loop

$$G_{\mathrm{LPF}}(s) = \frac{\omega_{\mathrm{n,trans}}}{s + \omega_{\mathrm{n,trans}}} \tag{4.36}$$

can be applied to the signal, rendering high-frequency behaviour invisible to the outer loop while only retaining the low-frequency feedback information.

### 4.3.2 Outer Control Loop

Finding the stabilising control loop for the LTI generated by the inversion law in equation 4.32 follows the same approach as for the attitude closed-loop. The driving design parameters $\zeta$ and $\omega_{\mathrm{n}}$ are again contained in the desired dynamics described by the second-order transfer function in equation 4.14. Since the relative degree of the outer loop is also $\rho = 2$, the same closed-loop structure arises as for the inner loop, leading to the gains

$$K_{\mathrm{p,trans}} = -(\sigma\,\omega_{\mathrm{n,att}})^2 \tag{4.37}$$

$$K_{\mathrm{d,trans}} = -2\,\zeta\,\sigma\,\omega_{\mathrm{n,att}} \tag{4.38}$$

Now the time-scale separation baseline frequency is no longer the TVC system natural frequency $\omega_{\mathrm{n,TVC}}$, but instead the inner loop natural frequency $\omega_{\mathrm{n,att}}$. The same time-scale separation is chosen between inner and outer loops as had been chosen for the separation of the inner loop and actuator dynamics, leading to the relation

$$\omega_{\mathrm{n,trans}} = \sigma^2 \omega_{\mathrm{n,TVC}} \tag{4.39}$$

Figure 4.6 shows the outer stabilising feedback loop applied to the outer pseudo LTI resulting from the linearising feedback. To improve readability, the inner loop dynamics are shown according to the assumption that, from the perspective of the outer loop under time-scale separation, they perfectly track their given reference $\Theta_{\mathrm{r,new}} = \Delta\Theta + \Theta_{\mathrm{r}}$.

Returning to the initial control strategy given in figure 4.1, the resemblance becomes apparent. The control architecture described so far stabilises two of the four TVC system controllable degrees-of-freedom, corresponding to the upper half of the parallel structure described in figure 4.1. The remaining parallel control architecture is derived in the following.

## 4.4 Combined Pitch and Yaw Plane Control

As demonstrated in the derivation of the CALLISTO system dynamics in chapter 3, due to their symmetry, the pitch and yaw plane dynamics are very similar. This similarity can be exploited to rapidly generate a four-degrees-of-freedom stabilising control architecture. The derivation follows the exact procedure presented for the pitch plane and is therefore only given in short form for both the linearisation loops and the stabilising control loops in yaw plane attitude and translation, starting with the linearisation loops.

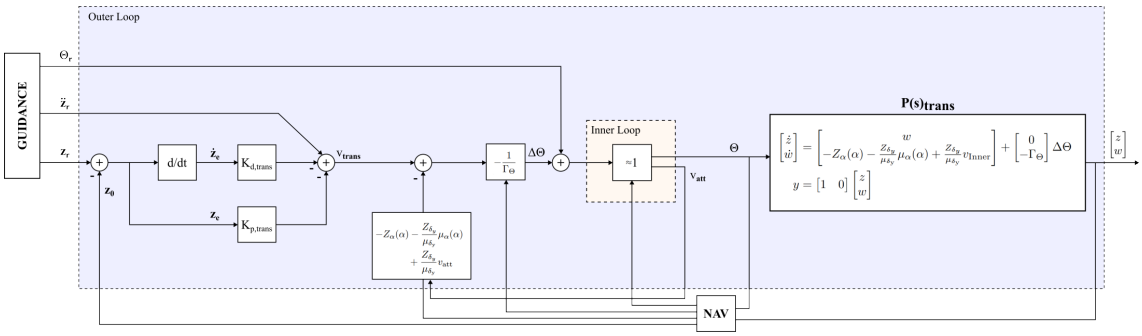



Figure 4.6: Pitch plane directional drift controller applied to the planar directional drift NDI linearisation loop with time-scale separated inner loop transfer behaviour.

### 4.4.1 Yaw Plane Linearisation Loops

Extracting the yaw plane translational and attitude dynamics from equation 3.62 and transforming the system into an input-affine representation using the small-angle approximation for the TVC deflection angle $\delta_\mathrm{z}$ gives

$$\begin{bmatrix} \dot{\Psi} \\ \dot{r} \\ \dot{y} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} r \\ \nu_\beta(\beta) \\ v \\ \Gamma_\Psi(\Psi) + Y_\beta(\beta) \end{bmatrix} + \begin{bmatrix} 0 \\ -\nu_{\delta_\mathrm{z}} \\ 0 \\ -Y_{\delta_z} \end{bmatrix} \delta_\mathrm{z} \tag{4.40}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Psi \\ r \\ y \\ v \end{bmatrix} \tag{4.41}$$

which serves as the starting point for the derivation of the attitude inversion law and the separated, cascaded plant structure akin to figure 4.4 in the yaw plane. Upon finding the relative degree to be $\rho = 2$, the yaw plane attitude inversion law takes the form

$$\delta_\mathrm{z} = \underbrace{-\frac{1}{\nu_{\delta_\mathrm{z}}}}_{B_\mathrm{v}(\boldsymbol{x})} (v_\mathrm{att} - \underbrace{\nu_\beta(\beta)}_{a_\mathrm{v}(\boldsymbol{x})}) \tag{4.42}$$

with the control effectiveness term $B_\mathrm{v}(\boldsymbol{x})$ and the feedback linearising term $a_\mathrm{v}(\boldsymbol{x})$.

Now, to achieve the plant decomposition into separated attitude and translational dynamics, the inner loop inversion law can be inserted into the translational dynamics, resulting in the yaw plant

$$\begin{bmatrix} \dot{\Psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} r \\ \nu_\beta(\beta) \end{bmatrix} + \begin{bmatrix} 0 \\ -\nu_{\delta_z} \end{bmatrix} \delta_\mathrm{z} \tag{4.43}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Psi \\ r \end{bmatrix} \tag{4.44}$$

and the lateral drift plant

$$\begin{bmatrix} \dot{y} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ Y_\beta(\beta) - \frac{Y_{\delta_z}}{\nu_{\delta_\mathrm{z}}} \nu_\beta(\beta) + \frac{Y_{\delta_z}}{\nu_{\delta_\mathrm{z}}} v_\mathrm{att} \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_\Psi \end{bmatrix} \Delta\Psi \tag{4.45}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} \tag{4.46}$$

To arrive at the lateral drift plant in equation 4.45, the translational contribution in equation 4.40 is reformed to take the inner loop yaw angle $\Psi$ as a control input by assuming a small yaw

angle in the $\Gamma_\Psi(\Psi)$ term as before in the pitch plane for the $\Gamma_\Theta(\Theta)$ term. From this decoupled outer plant, upon differentiating the output equation twice, results the inversion law

$$\Delta\Psi = \underbrace{\frac{1}{\Gamma_\Psi}}_{B_\mathrm{v}(\boldsymbol{x})^{-1}} \left( v_\mathrm{trans} - \underbrace{\left( Y_\beta(\beta) - \frac{Y_{\delta_z}}{\nu_{\delta_z}} \nu_\beta(\beta) + \frac{Y_{\delta_z}}{\nu_{\delta_z}} v_\mathrm{att} \right)}_{a_\mathrm{v}(\boldsymbol{x})} \right) \tag{4.47}$$

which is again based on a relative degree of $\rho = 2$. Note that beyond the differing yaw plane system coefficients, the only differences to the outer loop inversion law in the pitch plane are inverted signs in the control effectiveness $B_\mathrm{v}(\boldsymbol{x})$ and in the first contributing coefficient of the feedback linearising term $a_\mathrm{v}(\boldsymbol{x})$. With the two inversion laws given in equations 4.42 and 4.47, the respective pseudo LTIs are formed and the linear error controllers for both attitude and translational dynamics can be derived.

### 4.4.2 Yaw Plane Control Loops

As previously done in sections 4.2.2 and 4.3.2, the approach taken to derive the respective error controllers for the translation and the attitude dynamics in the yaw plane is again based on the assumption that inner and outer closed-loop bandwidths are correctly time-scale separated, allowing each to be treated individually. The desired dynamics of the inner loop are to follow the second-order system in equation 4.15. This equation is, however, not completely plane-agnostic. The natural frequency of the correct TVC actuator, responsible for either pitch or yaw plane deflections, has to be selected as a bandwidth baseline. Beyond that, the dynamics separation between inner and outer loop is again assumed to be equal to that between the inner loop and actuator bandwidths. Under these conditions, the control gains for the outer and inner loop error controllers are calculated exactly according to the relations for the pitch plane, given in equations 4.22, 4.23, 4.37 and 4.38.

With this in mind, each element in the initial control strategy given at the beginning of this chapter can be implemented. Because the derivation of this control architecture and its components was an iterative process, the following section describes some explored design solutions.

## 4.5 Explored Design Choices

The iterative process touched on architectural variants as well as filtering and signal processing options to improve control performance. The gathered experience in applying these alternative design choices is presented below, starting with the control architecture.

### 4.5.1 Parallel Control Architecture

The choice of a cascaded control structure was not immediately obvious at the beginning of this thesis. Solutions to consider the full MIMO system in the derivation of the inversion laws are also explored in the literature [36, 51, 22]. Such an alternative structure is shown in figure 4.7 at the example of the CALLISTO pitch plane dynamics. All previous scalar properties now become tensors, whose shape depends on the dimensions of the system inputs and outputs as well as the choice of control variables, if internal dynamics need to be avoided.

For CALLISTO, the system dynamics are not square, meaning that the number of system inputs $u \in \mathbb{R}^r$ is not equal to the number of system outputs $y \in \mathbb{R}^m$ with $r \neq m$ and specifically the system is under-actuated with $r < m$. This directly leads to the control effectiveness matrix

$$\boldsymbol{B}_\mathrm{v}(\boldsymbol{x}) = \begin{bmatrix} -\mu_{\delta_\mathrm{y}} \\ -Z_{\delta_\mathrm{y}} \end{bmatrix} \tag{4.48}$$

being singular and thus not invertible. It is a common restriction for the design of NDI linearisation loops that the system dynamics must be square in order to retain an invertible control effectiveness

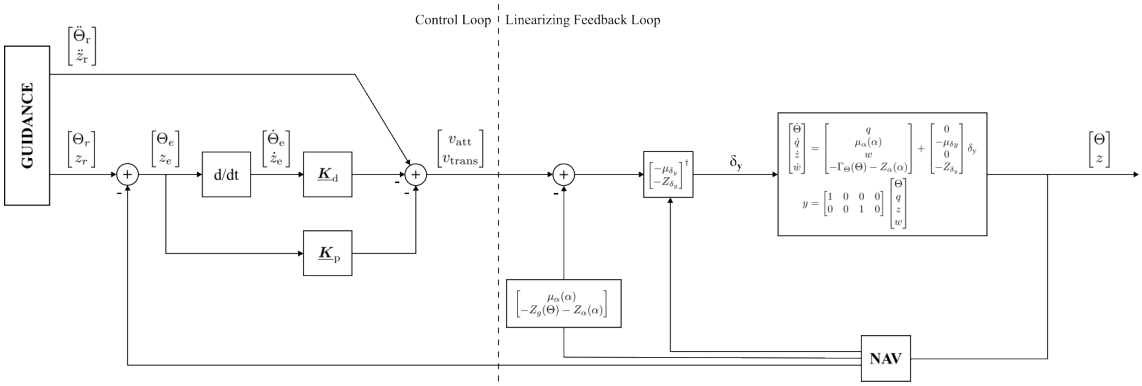



Figure 4.7: Parallel control architecture for attitude and translation using the pitch plane as an example.

matrix. The literature does, however, provide solutions for the opposing case, though mostly over-actuated systems are covered [26]. In the case of a singular control effectiveness, two solutions are presented. Either methods for the determination of control allocation operations are proposed to map the available control inputs to a set that would render the system square or a pseudo-inverse is applied to the singular $\boldsymbol{B}_{\mathrm{v}}(\boldsymbol{x})$ matrix, leading to considerable computation overhead if the pseudo-inverse must be calculated online. Within the scope of this thesis, a parallel control architecture was explored which used the Moore–Penrose pseudo-inverse to deal with the system's under actuation. Because the cascaded control architecture yielded better results faster, the MIMO-based structure was discarded, yet its potential advantages should not be underestimated, making it a worthwhile direction for future investigations.

Some of the envisioned advantages include the independence from the time-scale separation assumption. Not relying on time-scale separation may hinder the possibility of assigning bandwidths separately, which poses the risk of closed-loop actuation bandwidths overlapping with high-frequency dynamics like bending modes or propellant sloshing. Nevertheless, it could reduce the use of extra filters and their induced dynamics in certain situations.

Further, in the case of the proposed cascaded loop architecture, the rewritten outer loop plant has to be transformed into an input-affine system. To this end, the magnitudes of the Euler angles $\Theta$ and $\Psi$ were assumed to be small enough to enable the use of the small-angle approximation. This is only valid for trajectories that either retain a vertical vehicle attitude or strictly follow a gravity turn manoeuvre. By avoiding this assumption, the control solution can be applied to a much wider range of trajectories. Alternatives to the transformation to an input-affine system by means of approximating trigonometric functions are available for certain system forms, which might however limit the applicable range of the control solution [32].

### 4.5.2 Filters

There are two instances where filtering is advised in the presented control architecture. The first is the attitude pseudo control input feedback $v_{\mathrm{att}}$ in the linearising feedback term $a_{\mathrm{v}}(\boldsymbol{x})$ of the outer loop. Simulations have shown that not filtering this signal causes the overall system to behave very oscillatory, which is likely to be attributed to the time-scale separated bandwidth of the inner and outer loop. Here, it was found that a discrete (the simulation framework is discrete in nature) first-order Low-Pass Filter (LPF) with a cut-off frequency equal to the outer loop natural frequency yields the best performance given its low complexity. Other solutions, such as second-order filters and other cut-off frequencies, were also evaluated. The simulation of discrete, second-order filters in a SIMULINK environment for the desired frequencies yielded numerical inaccuracies, which made them unusable in the chosen setup. The origin of these effects remained unexplained. Other cut-off frequencies either performed worse both for faster dynamics, approaching the inner loop natural frequency, and for slower dynamics, which rendered the signal effectively too delayed and misaligned with the outer loop. The result that the outer loop natural frequency produces the best behaviour reinforces the rationale behind requiring a filter in this location at all.

The second instance where filtering was identified to be required can be seen in figure 4.5. Here, the pseudo control reference signal in the inner loop $\ddot{\Theta}_{\mathrm{r}}$ is obtained by means of differentiating the

new reference signal $\Theta_{r,new} = \Theta_r + \Delta\Theta$ twice. This differentiation amplifies any noise on the reference signal. Since the outer loop operates at the known time-scaled natural frequency $\omega_{n,trans}$, a discrete, first-order filter with this frequency as the cut-off was chosen to alleviate the noise amplification. The same variation of filtering solutions presented above was applied at this point, leading to the same conclusion.

# 5 Implementation and Simulation

In order to verify the performance of the NDI-based control architecture presented in the previous chapters, the proposed design is implemented in two MATLAB-SIMULINK frameworks. The first is a non-linear framework for testing in the proprietary high-fidelity non-linear simulator of the CALLISTO mission, where all the models and Guidance, Navigation and Control (GNC) subsystems are available. The simulation environment is used for verification and validation during Monte Carlo (MC) campaigns at the DLR. The simulation campaigns presented in this work are evaluated against requirements comparable to those used in a mission like CALLISTO. Secondly, a simplified linear verification framework was created solely for the purpose of testing the stability of the NDI-based design, evaluating its time and frequency response using classic indicators used for linear systems. These results can help isolate specific influences; for instance, the application of only the attitude linearisation loop on the system poles. They can further assist in developing an understanding of trends induced by small adjustments of the controller structure.

Both frameworks are presented in the following sections, providing the context for the results obtained with them in the following chapter.

## 5.1 Nonlinear CALLISTO Simulator

The high-fidelity non-linear CALLISTO simulator is the result of the joint development effort of the DLR and JAXA. They collaboratively develop the GNC system for the mission, which contains the guidance and control algorithms. The validation and verification campaigns required for the mission motivated the creation the framework [12]. It features a six-DOF simulation of the CALLISTO vehicle and the high-frequency dynamics of all the effects associated with a launch vehicle. It is derived from an acausal MODELICA-based set of VTVL physics models. The MODELICA models provide the complex multi-body interactions and interface with the MATLAB-SIMULINK implementation of the simulator. The advantage and motivation of such a combined architecture is the facilitation of high-fidelity plant modelling in combination with conventional SIMULINK-based GNC development processes to enable rapid prototyping and achieve accelerated system verification and validation across the mission [5]. Within this combined framework, the dynamic effects given in table 5.1 are considered.

Parametric uncertainties exist in almost all parameters, including the MoI and CoM, aerodynamic coefficients, actuator models and communication delays [3]. These are used in Monte Carlo simulations to investigate the impact of model uncertainties on the overall system performance and stability. In the perturbed case, uncertainty can also be injected in the external perturbations, e.g., sampling random wind profiles during MC campaigns.

### 5.1.1 Implementation Strategy

Since the presented NDI-based control architecture is designed specifically for the TVC deflections, it is implemented in isolation from other control channels. It interfaces with the guidance and navigation subsystems to obtain the various reference values and error signals. Both subsystems are treated as black boxes, assuming their bandwidths to be well separated from that of the closed-loop so as to act instantaneously from the perspective of the controller.

In order to create a modular design for the implementation of the proposed architecture, a SIMULINK library is created to facilitate the potential application to other vehicles. The control structure is segmented into modules and a dedicated library block is created for each. They interconnect to form the global control solution or, if wanted, subsets thereof. Figure 5.1 displays

Table 5.1: Dynamic effects represented within the six-degrees-of-freedom, high-fidelity CALLISTO simulator (based on [5, 6]).

| **Dynamic Effect** | **Modelling Strategy/Considerations** |
|---|---|
| Aerodynamics | Look-up tables, modelled as external forces/torques, wind included |
| Fins | Aerodynamic actuation properties, deployment MoI and aerodynamic changes |
| Dry Vehicle Dynamics | Baseline multi-body inertial model |
| Main Engine | Interpolation of look-up tables, modelled as external force/torque |
| Ground Contact | Modelled as reaction force/torques |
| Landing Legs | Deployment MoI and aerodynamic changes, deployment reaction forces/torques |
| Propellant Depletion | Mass, MoI and CoM changes |
| Propellant Sloshing | Pendulum modelling, CoM and MoI changes |
| RCS Dynamics | Dynamic thrust modelling |
| Structural Flexibility | Linear combination of independent bending modes |
| Tail-Wag-Dog Effects | Separate body modelling of the engine and induced interactions with upper body |
| TVC Dynamics | Modelling of the electric motors |
| Variable Mass Effects | External forces/torques resulting from propellant outflow |
| Communication Delays | Sample holding |

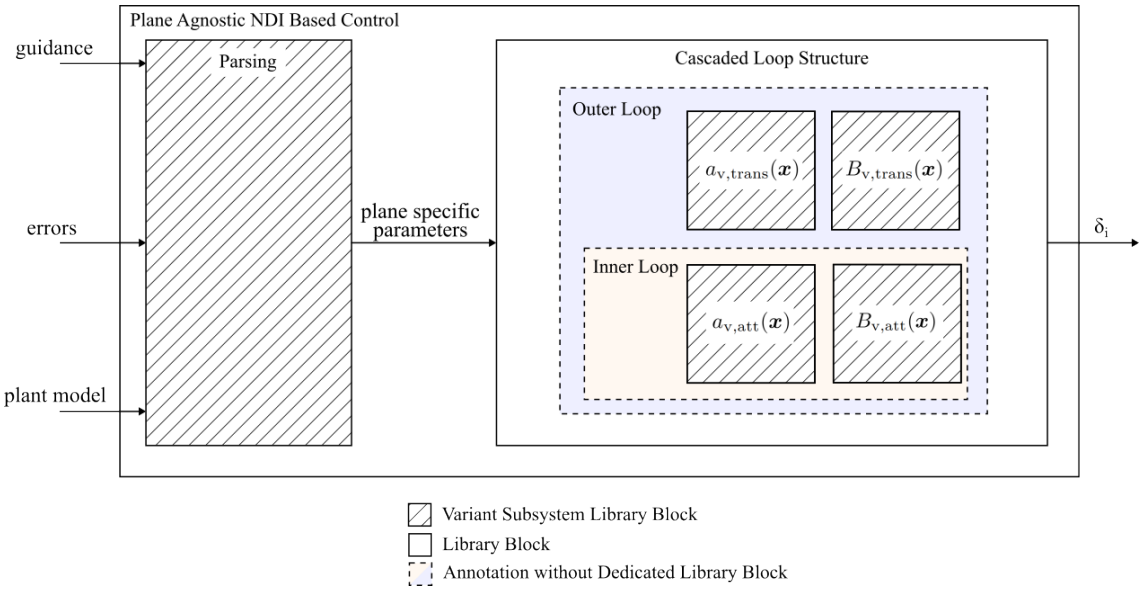



Figure 5.1: Modularised, plane-agnostic NDI control architecture for SIMULINK implementation purposes.

the separation and abstract relations between them. The *Plane Agnostic NDI Based Control* library block can be configured to either behave based on the pitch or yaw plane derivations, so that a full four-DOFs control structure consist of two of these blocks. Either configuration has its own parser logic and linearisation loop matrices, which are selected accordingly via variant subsystems. Because the cascaded loop structure is the same for both planes, it is not switched between each configuration. This would change if, e.g., different gains were used between planes. The external interfaces to the simulator are described in more detail in the following.

### 5.1.2 Module Interfaces

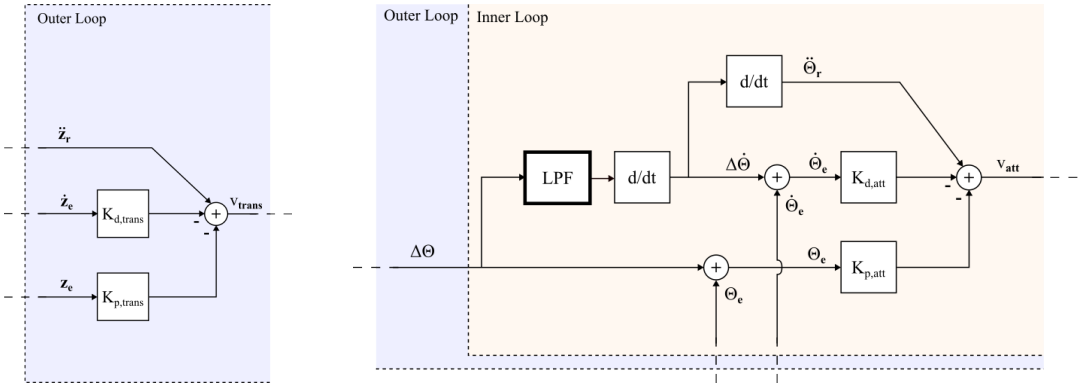
The signal requirements include the provision of live parameters to populate the control effectiveness and feedback linearisation terms $B_{\mathrm{v}}(\boldsymbol{x})$ and $a_{\mathrm{v}}(\boldsymbol{x})$ for both the inner and outer loops. The required signals must be contained within the navigation message. Some values are also provided by the guidance subsystem. Since the aerodynamic coefficients, though required for building the linearising feedback terms, are not directly available as measurements within the system, a reference aerodynamic database is used. It is interpolated with the current measurements of the angle of attack, the side-slip angle, the free-stream velocity and the altitude to generate estimates. These are subsequently passed to the corresponding blocks of the linearisation loops together with estimates of the dynamic pressure, the thrust force, the mass, the MoI, the CoM, correct frame transformations, the vehicle velocity, and aerodynamic geometry characteristics.

Besides the system model-specific signals in the construction of the linearisation loops, the pseudo control feed-forward is another intricacy of inversion-based implementations, which requires reference values for the translational and angular accelerations of the vehicle. When these are not available, the angular acceleration reference in both planes could be constructed by means of numeric differentiation of the respective angular velocity. This signal in turn could also be approximated by numerically differentiating the Euler angle reference signal.

As presented in chapter 4, to calculate the Euler angle reference, a contribution of the outer loop in the form of a reference correction $\Delta\Theta$ and the actual attitude reference provided by the guidance subsystem $\Theta_{\mathrm{r}}$ has to be considered. If these are not directly available, they can easily derived with the original error signal as follows

$$\Theta_{\mathrm{e,new}} = \Theta_{\mathrm{e}} + \Delta\Theta \tag{5.1}$$

and equally derivatives thereof. $\Theta_{\mathrm{e,new}}$ is approximately the outer loop reference correction $\Delta\Theta$ when a provided error $\Theta_{\mathrm{e}}$ of zero is assumed.




(a) (b)

Figure 5.2: Outer (a) and inner loop (b) reference and error signal construction.

Figure 5.2 depicts how the architecture changes compared to the generic formulation in chapter 4 as it is adjusted to the suite used in the non-linear simulations. The input interfaces for both the outer loop in sub-figure 5.2a and the inner loop in sub-figure 5.2b are given. In case of the outer loop, the availability of the precomputed error signal $z_{\mathrm{e}}$ and derivatives thereof means that the error calculation step is bypassed. The errors are directly passed to the error controller.

One of the consequences of these adjustments is the implementation of an LPF before the derivatives in the inner loop structure, highlighted in figure 5.2b and first presented at the end of the previous chapter. The chosen filter is a first-order LPF of the form in equation 4.36 with the natural frequency of the outer closed-loop $\sigma^2\omega_{\mathrm{n,TVC}}$ chosen as its cut-off frequency $\omega_{\mathrm{c}}$.

Another adjustment arises from the treatment of algebraic loops. Feeding back the inner loop pseudo control input signal $v_{\mathrm{att}}$ creates a dependency on itself. This issue can be navigated by introducing a unit delay on the feedback loop, enforcing the desired signal computation chronology in the discrete domain. With this change, on the first sample, the inner loop pseudo control input is calculated without the feedback loop and on the subsequent samples where a previous value is available, it is utilised.

On the output side of the NDI control module, the TVC deflections $\delta_\mathrm{y}$ and $\delta_\mathrm{z}$ are provided to the TVC system. The loop is closed via the output from the navigation subsystem.

### 5.1.3 Discretisation

As the simulator is based on a discrete, numerical solution scheme to propagate the system's evolution over time, the implementation has to be discretised accordingly. The design of the closed-loop itself was done in the continuous domain, and it is assumed that the simulated On-Board Computer (OBC) sampling rate is fast enough to retain the continuous control transfer behaviour in the discrete domain. Some elements, however, have to be explicitly transformed into a discrete formulation before implementation. The transfer functions of the delays and filters are discretised at the OBC sampling rate using a simple zero-order hold $z$-domain transformation scheme.

## 5.2 Linear Verification Framework

To verify stability, assess time- and frequency-domain performance, visualise trends and isolate error sources of the NDI-based control architecture, a linear verification model is iteratively designed alongside it. Its linear plant approximation as well as the architectural implementation are presented in the following section.

### 5.2.1 Linear Plant Model

A linearised version of the planar rigid-body dynamics derived in section 3.3.3 is used as the linear reference plant. To obtain it, the non-linear system given in equation 3.58 is transformed into a linear state space of the form

$$\begin{bmatrix} \dot{\Theta} \\ \ddot{\Theta} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \mu_\alpha & 0 & 0 & \mu_{\dot{z}} \\ 0 & 0 & 1 & 0 \\ -\Gamma_\Theta - Z_\alpha & 0 & 0 & -Z_{\dot{z}} \end{bmatrix} \begin{bmatrix} \Theta \\ \dot{\Theta} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\mu_{\delta_y} & -\mu_{\dot{z}} \\ 0 & 0 \\ -Z_{\delta_y} & Z_{\dot{z}} \end{bmatrix} \begin{bmatrix} \delta_\mathrm{y} \\ w_\mathrm{w} \end{bmatrix} \tag{5.2}$$

shown here at the example of the pitch plane. The linearisation is based on the derivation presented by Maces Hernandez et al. [3]. First, the aerodynamic coefficients are linearised by forming a Taylor expansion containing stability derivatives with respect to all changing variables and neglecting all terms except the dependence on the angle of attack $\frac{\partial C_\mathrm{i}}{\partial \alpha}$, as it has the dominant contribution. By then decomposing the angle of attack in the resulting linear relation $C_\mathrm{i} \approx \frac{\partial C_\mathrm{i}}{\partial \alpha}\alpha$ following equation 3.33 and assuming a small magnitude of the pitch and TVC deflection angles around the trim value, the linearised system coefficients become

$$\mu_\alpha = \frac{Q_\mathrm{dyn} S_\mathrm{ref} l_\mathrm{ref}}{J_\mathrm{yy}} \left( \frac{\partial^\mathrm{TF} C_\mathrm{My}}{\partial \alpha} + \frac{{}^\mathrm{B}l_\mathrm{TF}}{l_\mathrm{ref}} \frac{\partial^\mathrm{TF} C_\mathrm{N}}{\partial \alpha} \right) \qquad Z_\alpha = \frac{Q_\mathrm{dyn} S_\mathrm{ref}}{m} \frac{\partial^\mathrm{TF} C_\mathrm{N}}{\partial \alpha}$$

$$\mu_{\dot{z}} = \frac{\mu_\alpha}{V_\mathrm{T}} \qquad Z_{\dot{z}} = \frac{Z_\alpha}{V_\mathrm{T}}$$

$$\mu_{\delta_\mathrm{y}} = \frac{{}^\mathrm{S}l_\mathrm{CoM} T}{J_\mathrm{yy}} \qquad Z_{\delta_\mathrm{y}} = \frac{T}{m}$$

$$\Gamma_\Theta = g$$

A by-product of decomposing the angle of attack is the resulting separated dependence of the aerodynamic system coefficients $\mu_\alpha$ and $Z_\alpha$ on the resulting states $\Theta$ and $\dot{z}$. This further leads to an additional control input in the linear system equation 5.2 in the form of the wind velocity directional contribution $w_\mathrm{w}$. This input is used to model the effect of wind on the closed-loop dynamics.

In order to investigate the effect of uncertainties on the closed-loop performance, the linear plant model can be populated with uncertain parameters. A set of ranges is specified for each

uncertain variable, which can be chosen either to be the same as in the non-linear CALLISTO framework presented above or as a more conservative case with 20% on all variables. Each variable is then calculated according to

$$x = x_{\text{nominal}} + \Delta x \tag{5.3}$$

shown for a generic variable $x$, where the uncertainty is a uniformly sampled, random value within the specified percentage ranges.

Beyond the rigid-body dynamics, the TVC actuator dynamics are represented with a discrete transfer function. It is introduced to the signal flow before the plant. The transfer function describes a second-order system with the natural frequency of the TVC system $\omega_{\text{n,TVC}}$ and the damping ratio $\zeta = 0.7071$. The actuator delay is modelled with a discrete chain delay block, which holds the signal for a specified number of samples $n_{\text{samples}}$. An actuator bias is modelled with a DC gain $K_{\text{bias}}$ on the transfer behaviour. In the nominal case this gain is simply $K_{\text{bias}} = 1$. When uncertain values are used for the plant and actuator, the transfer function is built by adding uncertain deltas $\Delta\omega_{\text{n,TVC}}$, $\Delta\zeta$, $\Delta K_{\text{bias}}$ and $\Delta n_{\text{samples}}$ of all characteristic parameters. The final transfer behaviour can be described with

$$\begin{aligned} G_{\text{TVC}}(z) = {} & \frac{1}{z^{-(\text{n}_{\text{samples}}+\Delta\text{n}_{\text{samples}})}} \cdot \dots \\ & \cdot \mathcal{Z}\left\{ \frac{(1+\Delta K_{\text{bias}})(\omega_{\text{n,TVC}}+\Delta\omega_{\text{n,TVC}})^2}{s^2 + 2(\zeta+\Delta\zeta)(\omega_{\text{n,TVC}}+\Delta\omega_{\text{n,TVC}})s + (\omega_{\text{n,TVC}}+\Delta\omega_{\text{n,TVC}})^2} \right\} \end{aligned}$$

where the z-transform to the discrete domain of the transfer function is left unspecified to retain the parametrised second-order form.

### 5.2.2 Adjusted Inversion Law

A separate set of inversion laws for the outer and inner loops is derived based on the linear model presented above. Following the same process presented in chapter 4 yields the inversion law for the inner loop of the pitch plane

$$\delta_{\text{y}} = \underbrace{\frac{1}{-\mu_{\delta_{\text{y}}}}}_{B_{\text{v}}(\boldsymbol{x})^{-1}} \left(v_{\text{att}} - \underbrace{(\mu_\alpha \Theta + \mu_{\dot{z}} \dot{z})}_{a_{\text{v}}(\boldsymbol{x})}\right) \tag{5.4}$$

which contains linear dependencies on states $\Theta$ and $\dot{z}$ in the now decomposed $\mu_\alpha(\alpha)$. The contributions from the directional wind velocity $w_{\text{w}}$ are set to zero, because the linearisation loop is built solely from the nominal plant model. The outer loop inversion law in the pitch plane then takes the form

$$\Delta\Theta = \underbrace{-\frac{1}{(\Gamma_\Theta + Z_\alpha + \frac{Z_{\delta_{\text{y}}}}{\mu_{\delta_{\text{y}}}}\mu_\alpha)}}_{B_{\text{v}}(\boldsymbol{x})^{-1}} \left( v_{\text{trans}} - \underbrace{\left( \left( -Z_{\dot{\text{z}}} + \frac{Z_{\delta_y}}{\mu_{\delta_{\text{y}}}}\mu_{\dot{z}} \right) \dot{z} - \frac{Z_{\delta_y}}{\mu_{\delta_{\text{y}}}} v_{\text{att}} \right)}_{a_{\text{v}}(\boldsymbol{x})} \right) \tag{5.5}$$

The linear analysis framework can be configured to use either the non-linear plant model-based or the linear system model-based derivation of the inversion laws to compare results. Due to time limitations, however, the yaw plane adaptions were not carried out, making the analysis with the linear reference plant possible only for the pitch plane dynamics.

### 5.2.3 Linear Reference Model Architecture

With the plant model and adjusted inversion law at hand, the control architecture is now re-implemented in a separate MATLAB SIMULINK model which mimics the simulation environment of the non-linear simulator including sampling times and numerical solver choices. The interfaces that

were previously connected to the guidance and navigation subsystems in the non-linear simulator are now artificially created.

The navigation solution is assumed to be ideal, allowing perfect measurement of all system states, making them directly available from the plant output. The purpose of the linear reference model is to enable standard linear analysis techniques to be applied to the NDI-based control architecture, which leads to the artificial replacement of the guidance system by any reference signal, e.g., a step function.

Generating the time-varying extended-state data, which is used in the construction of the linearisation loop matrices $\boldsymbol{B}_{\mathrm{v}}(\boldsymbol{x})$ and $\boldsymbol{a}_{\mathrm{v}}(\boldsymbol{x})$ and the current plant approximation, is done by interfacing the reference trajectory of the scenario of interest. It contains the trim values for a range of parameters. The system coefficients which are calculated from this data for a specific sample along the trajectory are injected into a Simulink discrete state-space model, enabling the dynamics of a specific operating point of the plant to be frozen as an LTI and analysed as an isolated unit.

With the Simulink linear reference model comes a custom NDI linear analysis Matlab library which includes a model initialisation function `init_RM.m` and two analysis suites `analyse_linear_RM.m` and `analyse_non-linear_CLT_SIM.m` as well as referenced functions and utilities. The initialisation function merely extracts the trajectory data and populates the workspace with sampled coefficients. Both nominal and uncertain coefficients can be generated. The nominal coefficients are always used in the construction of the linearising loop terms, whereas the discrete state space and the actuator transfer function used to model the plant can be populated using either nominal or uncertain coefficients. The Simulink model accesses the resulting data structures in the Matlab workspace accordingly. To illustrate how the reference model is created, figure 5.3 shows its relationship to the linear analysis Matlab library as well as the reference trajectory data and a set of user settings.

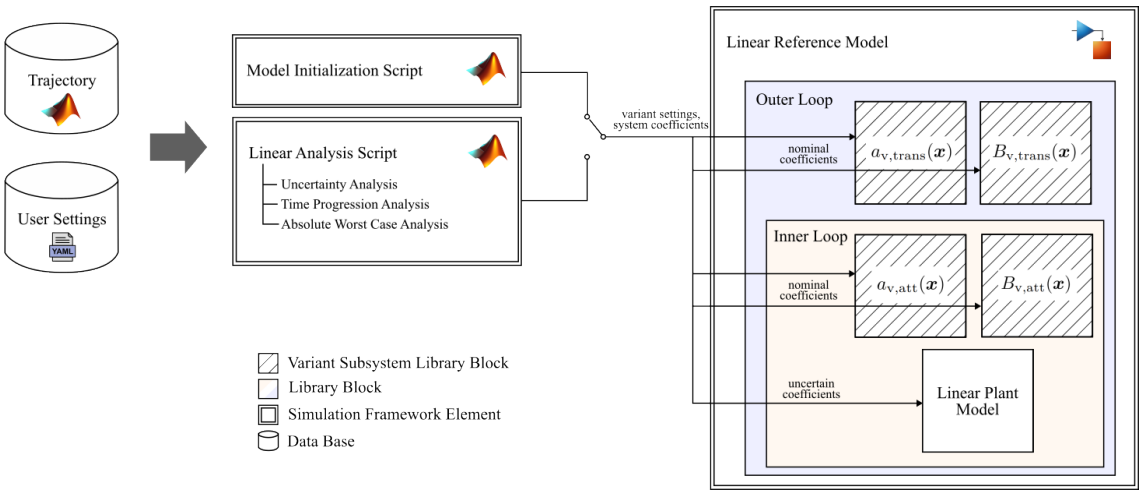



Figure 5.3: Linear reference model data flow diagram.

The reference trajectory is imported from a file into the workspace, where it can be accessed by processing functions that create the system coefficients. User settings are configured in a YAML file. They contain a number of variables which control how the Simulink reference model is built, and what data the analysis scripts generate. All settings are specified in table 5.2 with a short description as well as accepted values and data types.

The implementation is limited to the investigation of two DOFs at once, the drift and attitude dynamics in one plane. As such, the desired plane has to be specified before the verification is executed using the variable *plane*. The considered DOFs within the plane are also customisable, allowing the user to switch between the analysis of the inner loop as an isolated structure and the full cascaded architecture by setting the parameter *simDOF*. Actuator dynamics can be ignored or modelled as a second-order system with induced delay by switching the settings *useActuator* and *useActuatorDelay*. The nominal actuator delay can be configured under *nominalDelay*. The setting *useAlphaDecomp* is of special note: it switches between the linear model-based inversion laws given in this chapter and the non-linear model-based inversion laws from chapter 4. This is useful as it allows to investigate the impact of model discrepancies between the plant model and

Table 5.2: Reference model and analysis user settings for the linear verification framework.

| **Setting** | **Description** | **Values** | **Data-type** |
|---|---|---|---|
| Model Related Settings | | | |
| plane | Sets the modelled plane. | "pitch", "yaw" | string |
| simDOF | Sets the modelled degrees of freedom. | 1 → attitude<br>2 → +translation | integer |
| useActuator | Switches modelling of actuator dynamics. | true, false | boolean |
| useActuatorDelay | Switches modelling of actuator delays. | true, false | boolean |
| nominalDelay | Sets the nominal actuator delay in seconds. | Any positive floating point. | float |
| useAlphaDecomp | Switches between alpha decomposed and state inferred plant modelling. | true, false | boolean |
| Analysis Related Settings | | | |
| variant | Selects the analysis variant. | "uncertainty", "timeProgression", "absWorstMargins" | string |
| loopSelect | Selects the analysed loop structure. | "inner", "outer" | string |
| numRuns | Number of sets of uncertain variables for robust analysis. | Any positive integer. | integer |
| trjSamples | Number of discrete samples along the trajectory for time Progression Analysis | Any positive integer. | integer |
| uncertaintySample | Determines the time sample for uncertainty analysis. | "maxQ", "hover" | string |
| uncertaintySeverity | Determines the severity of modelled uncertainty. | "design", "conservative", "none" | string |
| smallValueFiltering | Filter all values in the system matrix below a threshold. | true, false | boolean |
| smallValueThreshold | Define small-value filtering threshold. | Any positive floating point | float |
| useMinReal | Filter all poles that cancel with zeros. | true, false | boolean |
| minRealTolerance | Define threshold for pole-zero cancelling. | Any positive floating point. | float |

the linearisation loop. Each of these settings configures the model structure by switching variant subsystems, making almost any combination a valid choice. Due to time limitations, an implementation of the linearisation loop structures for the alpha decomposed-based inversion laws in the yaw plane was not completed, making the setting combination plane=yaw and useAlphaDecomp=true invalid.

While the initialisation script simply populates the linear reference model, the analysis suites run a full linear analysis based on the provided user specifications. The following chapter will present the analysis strategy as well as the generated results at length.

# 6 Results

To answer the question of whether or not the NDI-based control architecture is a feasible option for the TVC control of VTVL RLVs, using the CALLISTO mission as a benchmark, the presented design is analysed in both the linear and non-linear domains using the previously presented verification frameworks. Across both domains, a set of performance requirements similar to those of the CALLISTO mission is used to judge the quality of the control solution. This analysis is not intended to qualify the solution for flight readiness, it merely serves to judge its feasibility. The requirements are summarised in the following section.

## 6.1 Performance Metrics

The NDI-based controller is benchmarked against a set of requirements similar to those of the CALLISTO mission. Stability is always a requirement and to assess how close a system is to instability and thus its robustness to uncertainties, phase and gain margins are analysed. The phase margin $\gamma$ is a measure of how much phase lag the system can withstand before it reaches the verge of instability at the frequency at which the open-loop magnitude is one. In the frequency domain, this is the case at the so called gain crossover frequency with $|G(j\omega)| = 1$. Further, the gain margin is defined as the inverse of the gain magnitude at the phase crossover frequency, when the phase angle is $\phi = -180°$ [50]. It is commonly expressed in decibels. Akin to the phase margin, it can be interpreted as the maximum gain a system can be exposed to before it reaches instability. Common values used in control engineering are given below [50].

- Phase Margin Requirement: $\gamma > 30°$
- Gain Margin Requirement: $K_{\mathrm{g}} > 6$ dB

In order to assess robustness against simultaneous gain and phase variations, meaning outside of the static $|G(j\omega)| = 1$ and $\phi = -180°$ cases, an elliptical region is defined around this point with a semi-minor axis equal to the gain margin 6 dB and a semi-major axis equal to the phase margin 30°. This can be depicted in a Nichols diagram [50]. The margins are obtained using the linear reference model and the linear analysis framework.

The non-linear simulator is used to assess the desired tracking performance of the controller. Tracking performance is measured by investigating the errors in all controllable states during a test scenario and a maximum error is proposed as a requirement. This is the case for the position, velocity, Euler angles and angular velocity errors. For the position, the maximum deflection requirement is related to the envelope for the landing ellipse at touchdown.

Another performance measure is the system's robustness against uncertainties in the system model or external perturbations. To investigate this characteristic, Monte Carlo simulations are conducted in the non-linear simulator. A Monte Carlo simulation generates sets of random variations within predefined ranges for all uncertain parameters of a model and simulates the corresponding responses, investigating an uncertain region around a nominal system. The more sets of random variations are created and analysed, the more confidently the impact of the uncertainties can be assessed. Because couplings between uncertain variables can be intricate and unpredictable, the number of required simulations rises strongly with the number of uncertain variables, leading to large computation requirements for modern validation campaigns. Under the influence of these uncertainties, the tracking performance is assessed. Alongside this analysis, using the linear reference model, random sets of uncertainties are applied to the investigated LTIs to assess margins and judge robustness.

Also investigated in both the linear reference model and the non-linear CALLISTO simulator is the disturbance rejection of the control architecture. In the linear reference model, wind is introduced as a disturbance in the plant dynamics as shown in equation 5.2. Here, the transfer behaviour of a nominal wind disturbance input to any controllable state can be investigated. The Monte Carlo campaigns in the non-linear simulator additionally consider uncertainties in these disturbances alongside plant uncertainties.

Though there is no requirement on the magnitude and rate of the control action in this thesis, the quality of a controller's behaviour can be assessed by these measures. Lower rates and magnitudes are generally better, reducing the stress on actuators and the energetic cost of operation. The range of motion of the TVC actuators is restricted to a maximum deflection. When the control action surpasses the possible range of motion, as a protective measure, the signal simply saturates.

With these measures in mind, the following sections will present the analysis strategy for both the linear and non-linear domains, starting with the linear verification framework.

## 6.2 Linear Verification Framework Results

To better understand how the linearising and stabilising feedback loops behave over the course of an entire trajectory, how model mismatch influences the linearisation quality and how uncertainties lead to a deterioration of the performance, the results obtained in the linear domain are presented in the following. First, however, the analysis strategy is presented. As mentioned in chapter 5, the linear reference model comes with a MATLAB NDI analysis library which contains the function `analyze_linear_RM.m`. It is the analysis suite used for all sets of investigations carried out with the reference model. The first mode of operation of the function produces a time progression analysis along the provided state trajectory. A second variant produces a robustness analysis by investigating randomised sets of uncertainties applied to a single, frozen sample along the state trajectory. Finally, a combination of both modes can be carried out, which investigates the impact of uncertainties on each of the specified discrete samples along the trajectory to find the overall worst-case margins. Selecting the mode of operation is done in the user settings YAML file under the setting *variant*.

Figure 6.1 shows a flow chart of the script's logical execution for the two major modes of operation *timeProgression* and *uncertainty*.

When the function is called, a set of initialising statements is executed, which load the user settings specified in the accompanying YAML file and the data from the reference trajectory. Since the linear SIMULINK reference model has access to the MATLAB workspace, after loading the user settings, all variant subsystems within are configured according to the specified simulation settings. After logging and verifying the loaded user settings, the initialisation process is finalised by a statement which loads the SIMULINK model as a variable, making it accessible from the script. Now, two settings from the analysis section of table 5.2 are queried. The *simDOF* and *loopSelect* govern which specific transfer functions are to be generated. Based on these settings, a set of input-output objects is created using the *Matlab* `linio()` function. They contain the input and output signals chosen from the *Simulink* model for each of the desired functions. These objects are created for the corresponding closed-loop $T(s)$, loop transfer function $L(s)$, open-loop plant $P(s)$ and open-loop pseudo-linear plant $P_v(s)$ transfer functions. The pseudo-linear plant refers here to the double integrator which results after applying the linearising feedback loop. Technically, the double integrator is a linear system, and especially in the linear transfer functions extracted using the linear verification framework this transfer behaviour is fully linear. Yet, because usually the linearisation loop supplies merely an approximation of a chain of integrators whose quality depends on the precision of the mathematical model at its core, the term pseudo-linear plant is more generally applicable. Here, the general term is used. Continuing along the script's logic, the setting defining the analysis operation mode *variant* is used to select the corresponding logic.

The *timeProgression* path first defines the discrete time grid along the trajectory using the `defineTimeGrid()` function. It samples in a uniform distribution according to the number of samples specified in the setting *trjSamples*. Because the actuator transfer behaviour does not change along the trajectory, its transfer function is calculated using the function `calcActuatorTF()` with nominal values. This nominal transfer function is accessed by a discrete transfer function block in the reference model, setting the dynamics for all subsequent investigations. The following for loop is the core of the *timeProgression* scenario. It iterates over each of the previously generated

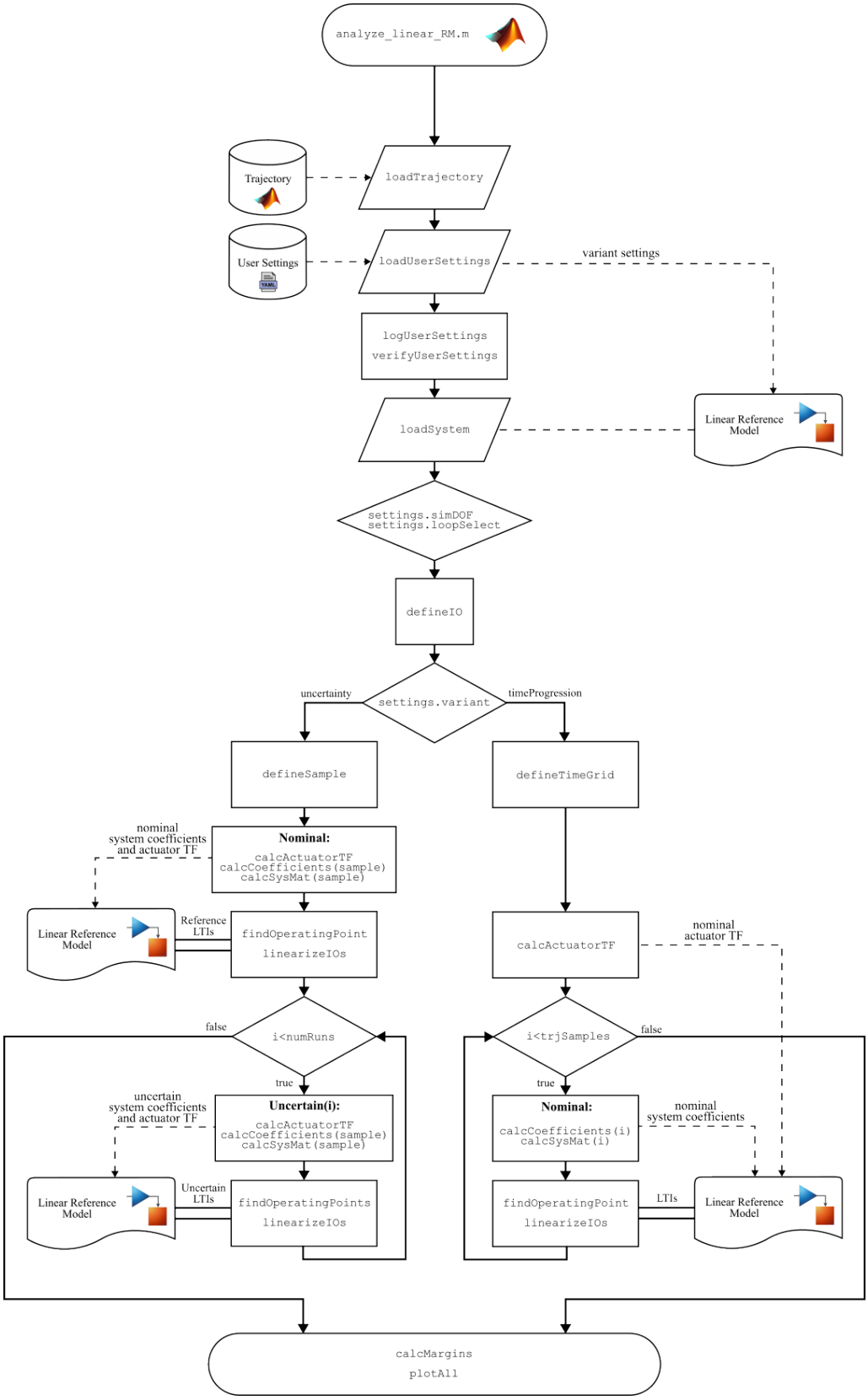



Figure 6.1: Flowchart depicting the logical progression of the linear analysis suite.

samples and calculates the nominal system coefficients and linear system matrices at each of them. In SIMULINK, the linear reference model updates the plant and linearisation loop accordingly. Then, all input-outputs sets are linearised using the MATLAB functions `findop()` and `linearize()`. The resulting LTIs are appended to system arrays for each of the transfer behaviours $T(s)$, $L(s)$, $P(s)$ and $P_{\mathrm{v}}(s)$ and the loop is repeated. When a set of LTIs at each of the sampled points along the trajectory is generated, the loop exits and margins are calculated for all systems using the `calcMargins()` function. Finally, step response, pole-zero, Bode and Nichols plots are created for each transfer behaviour using dedicated plotting functions.

The second operation mode *uncertainty* works in a similar manner. Instead of iterating over samples along the trajectory, the script iterates over randomised sets of uncertainties. Beginning again at the decision block of the analysis operation mode in the flowchart, instead of defining an array of samples along the trajectory, the function `defineSample()` returns a specific sample based on the setting *uncertaintySample*. It finds either the point in time of maximum dynamic pressure $Q_{\max}$ or the midpoint of the hover duration. Next, nominal reference LTIs are extracted to serve as a reference for later comparison to the uncertain behaviour and to populate the linearisation loops, as these are built exclusively on nominal data. The extraction process follows the same logic as before, first calculating nominal system characteristics with `calcActuatorTF()`, `calcCoefficients()` and `calcSysMat()`. System coefficients and matrices are created for the frozen dynamics at the chosen sample. The linear reference model updates automatically and the `findop()` and `linearize()` functions provide the LTIs for all transfer behaviours. Then the uncertainties are sampled using a for loop for the number of sets of uncertainties specified in the setting *numRuns*. For each of the sets, random uncertainties are generated using the `updateUncert()` function. The ranges used for each uncertain variable are controlled via the setting *uncertaintySeverity*. It allows either to use values similar to the design uncertainties for the mission, 20% maximum uncertainty on all uncertain variables or no uncertainties at all. Then all system characteristics are recomputed using the now familiar functions and the LTIs are extracted. When the specified number of uncertain LTIs has been generated, the loop exits and margins and plots are generated in the same way as for the *timeProgression* operation mode.

The analysis related section of the user settings also contains options for filtering. The *smallValueFiltering* setting determines whether or not values below a specified threshold for $\mu_\alpha$ are neglected in the system matrices of the linear reference model. The *useMinReal* setting controls whether or not zeros and poles of all transfer functions are numerically cancelled within the specified threshold. In the following results section neither filter is utilised.

A final remark must be made on the last operation mode and another feature script in the root directory of the library. The third operation mode *absWorstMargins* is not represented in the flowchart as its logic simply combines the other two operation modes, conducting an uncertainty sampling on each discrete time sample along the trajectory. The resulting margins are scanned to find the absolute worst case for the chosen linear reference model configuration. To conduct large quantities of simulations, the script `run_batch_analysis.m` provides the ability to queue multiple sets of model and analysis settings. Each combination of the provided settings is executed in succession, automating the analysis process across scenarios.

The following subsections present the results generated with this linear analysis suite, separating them into the time progression and robustness analysis regimes as well as a discussion on disturbance rejection behaviour.

### 6.2.1 System Evolution Analysis

Like the design of the NDI-based control architecture, the linear analysis progressively increases the model complexity starting with planar, attitude control only, moving to planar attitude and translational control and their evolution over the course of the trajectory. The presented results correspond to the pitch plane and the first demonstration flight scenario of CALLISTO. The yaw plane results are not presented, as preliminary analysis has shown that they differ only numerically from the pitch plane results.

Regardless of the chosen model variant and ensuing level of system complexity, the sampling along the trajectory remains unchanged. For the evolution analysis, 30 uniformly distributed samples are taken as depicted in figure 6.2. This was deemed sufficient to capture all relevant changes in system behaviour across the flight. Also uniform across all investigated scenarios is the plant model. It always adheres to the nominal linear state-space and actuator transfer behaviour

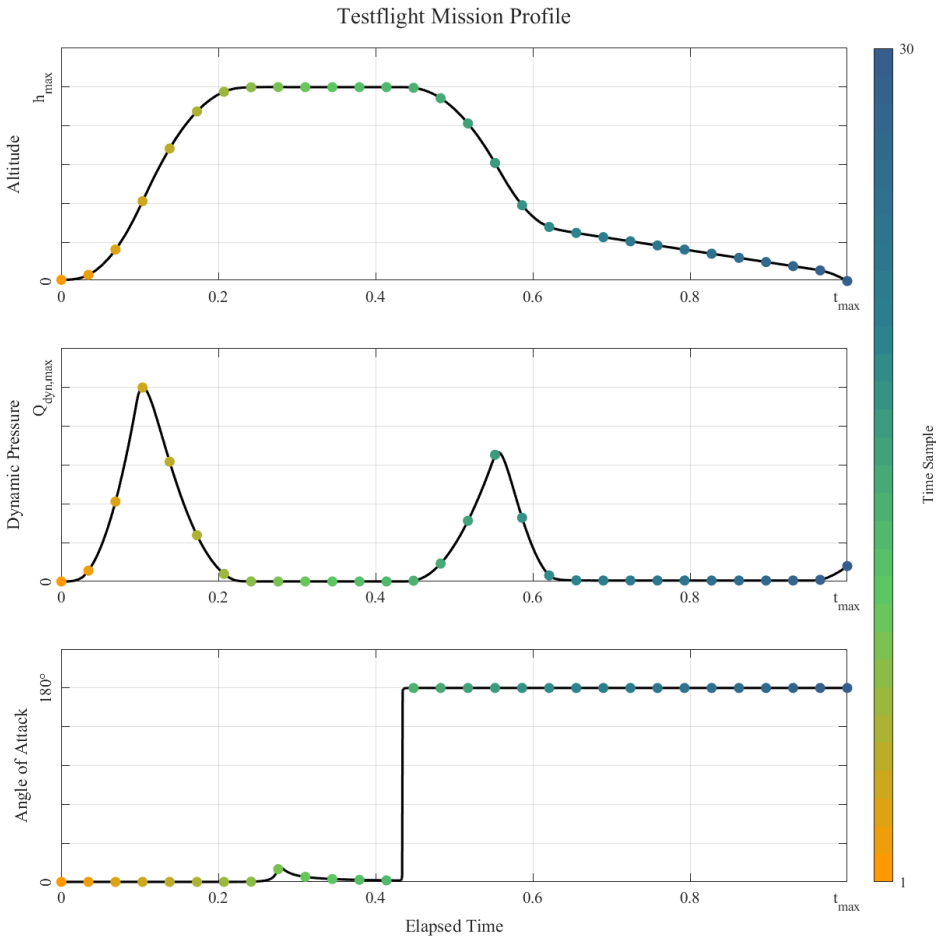



Figure 6.2: Mission Profile of the low-energy test flight demonstration scenario, with 30 discrete time samples along the nominal trajectory.

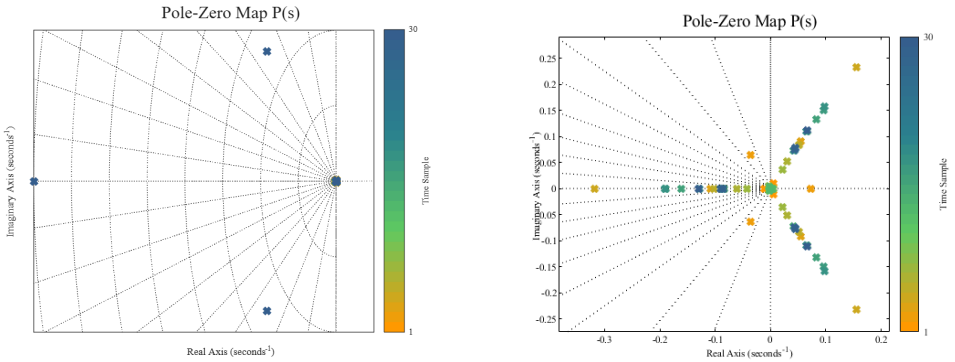
showcased in chapter 5. Because of this, the plant behaviour is presented first.

### Plant Evolution

As mentioned in the description of the analysis script, four distinct transfer functions are investigated. First, the open-loop plant $P(s)$ is analysed. In the pitch plane, it describes the transfer behaviour from the control input $\delta_y$ to the pitch angle $\Theta$, which includes the actuator dynamics. Figure 6.3 shows the system's poles in the complex plane. Figure 6.3a is the global depiction of the poles. The fast dynamics of the actuator have stable poles with large real part magnitudes whereas the plant poles lie closer to the origin. They dominate the behaviour of the uncontrolled plant. To better characterise the dominant poles, figure 6.3b shows a detail view of the region around the origin. The system is unstable across the entire trajectory, as there is no instance where all poles are exclusively in the left half of the complex plane.

The poles with non-zero imaginary parts, which follow a progression along diagonals around the origin, here represent the behaviour of the attitude degree of freedom in the pitch angle.

It is stable only for a short period of time in the ascent phase of the test flight, as can be seen from the orange marker on the left half of the complex plane. In contrast the behaviour of the translational degree of freedom in the directional drift, represented by the poles on the real axis, is mostly stable. Most of its poles lie in the left half of the complex plane. The two degrees of freedom have an inverse relation with regard to stability, leading to the attitude channel always being unstable when the translational channel is stable and vice versa. In the attitude, this behaviour is explained by the aerodynamic stability of the vehicle. During ascent, the CoP




(a) All poles.

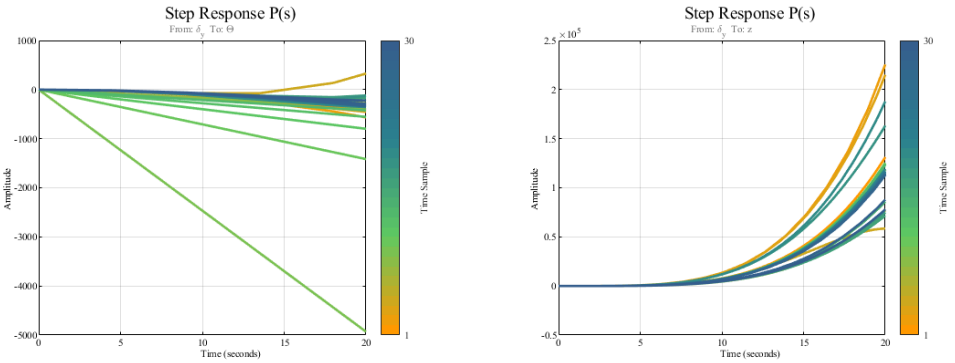
(b) Detail view.

Figure 6.3: Pole-zero map of the nominal linear reference plant and actuator dynamics $P(s)$ for 30 discrete time samples along the trajectory.

lies before the CoM as seen from the direction of the free-stream velocity. This is analogous to the stability behaviour of an inverted pendulum, which is unstable without active control. During the descent the opposite is true: the CoP lies behind the CoM as seen from the direction of the free-stream velocity. This configuration is stable, corresponding to a pendulum in equilibrium. During hover, the system's poles cluster around the origin.

The described behaviour cannot exactly be observed in the data in figure 6.3b. It is only correctly reflected for the ascent and hover phases of the test flight, which likely points to an error in the utilised plant model. It should therefore always be considered that the results obtained using this model are only representative of the ascent and hover phases of the vehicle.

As expected, when the open-loop time response is investigated, highly divergent behaviour can be observed across the trajectory in both attitude and translational channels as depicted in figure 6.4. Depending on the channel and flight phase, the behaviour is either oscillatory or monotone.




(a) Attitude channel.

(b) Directional Drift channel.

Figure 6.4: Step response of the nominal linear reference plant and actuator dynamics $P(s)$ for 30 discrete time samples along the trajectory.

### Closed Loop Evolution: Isolated Inner Loop

As previously introduced in chapter 2, an NDI-based control architecture is structured into two loops: the linearising feedback loop and the stabilising feedback loop. For the initial investigation, considering only the isolated attitude channel, the quality of the linearising feedback loop can be investigated by comparing the transfer behaviour from pseudo control input $v_{\text{att}}$ to the attitude state $\Theta$, here denoted as $P_{\text{v,att}}(s)$, to a double integrator $G_{\text{DI}}(s) = \frac{1}{s^2}$. Figure 6.5 shows a detail view of the corresponding poles, with the evolution of the pseudo-linear plant $P_{\text{v,att}}(s)$. This detail view does not depict the actuator poles, as they are not affected by the linearisation loop, since they are not considered in the derivation of the NDI law. It can be seen that the poles

move closer to the origin of the complex plane and, as such, they approximate the behaviour of the double integrator, which is the linear transfer behaviour obtained through the linearisation. Collectively, the magnitudes of the real parts become smaller, yet some outliers remain less affected by the linearisation effort. Least affected are of course the poles that correspond to the translational channel on the real axis, because only the attitude linearisation loop has been applied at this point. Beyond that, however, remain certain poles of the attitude dynamics whose complex absolute values remain relatively large. When these are traced back to the sample they originate in, the descent phase is identified. Again, this is likely a contribution of the error that also causes the plant model to be inaccurate in the descent phase.

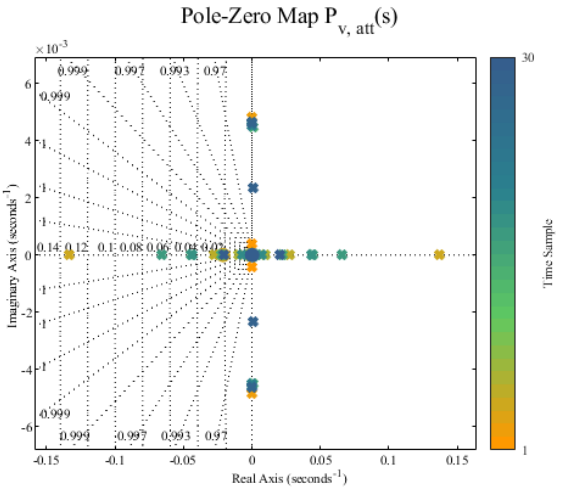



Figure 6.5: Pole-zero map around the origin of the inner pseudo-linearised plant $P_{\mathrm{v,att}}(s)$ for 30 nominal discrete time samples along the trajectory.

The discrepancy from the double integrator is further influenced by numerical inaccuracies. This becomes visible when the Bode plot of the pseudo-linearised system in figure 6.6 is considered.

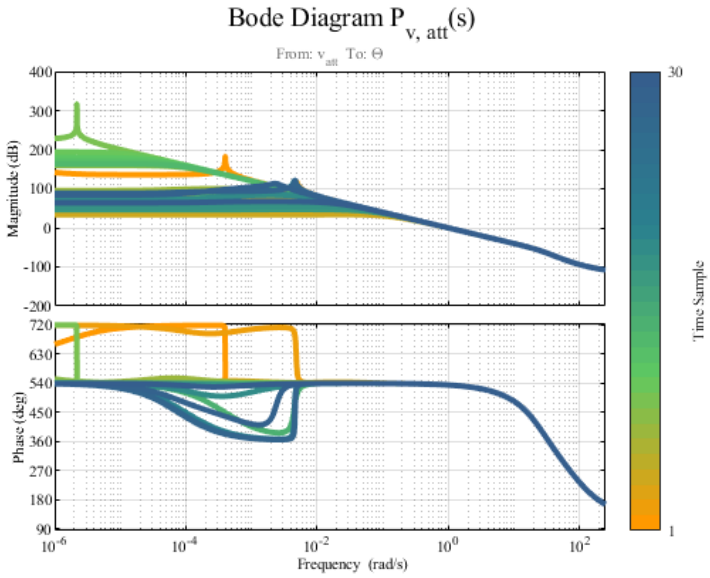


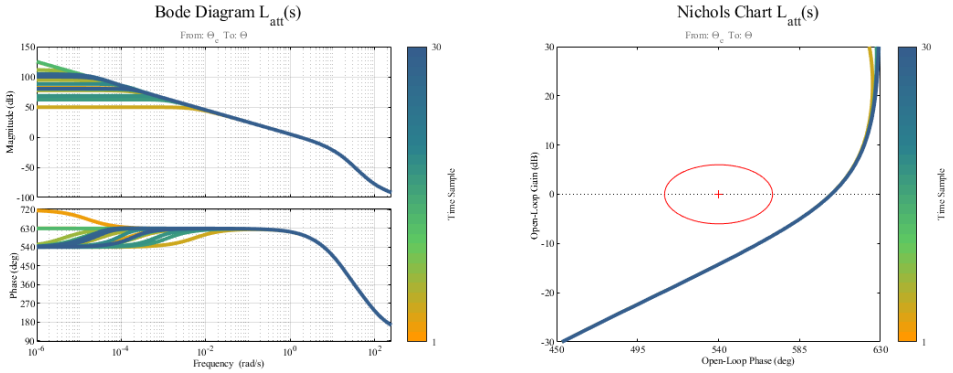

Figure 6.6: Bode diagram of the inner pseudo-linearised plant $P_{\mathrm{v,att}}(s)$ for 30 nominal discrete time samples along the trajectory.

The low-frequency domain exhibits the largest linearisation inaccuracies seen in the non-uniform frequency response below about 0.01 $\frac{\mathrm{rad}}{\mathrm{s}}$. Here, a roll-off behaviour is visible instead of the magnitude increasing without bound as would be the case for the double integrator. This dispersion

in the low frequencies originates in the open-loop system dynamics. The actuator dynamics can be seen as a slight roll-off at their natural frequency in the high end of the frequency domain.

Above about 0.01 $\frac{\text{rad}}{\text{s}}$, the linearisation reduces system dynamic dispersion. In this frequency range, the different LTIs along the trajectory behave almost the same. This is the reason why no gain scheduling is required in NDI-based architectures. While the linearisation remains accurate, a static controller can be used to control this uniform system.

To assess the impact of linearisation inaccuracies along the trajectory and investigate the effectiveness of an NDI-based controller, the following discussion presents the nominal system transfer behaviour when the stabilising feedback loop is applied. To this end, two characteristic LTIs are investigated: the loop transfer $L_{\text{att}}(s)$, which describes the open-loop transfer behaviour from the state error $\Theta_{\text{e}}$ to the state $\Theta$ under nominal conditions, and the closed-loop transfer function $T_{\text{att}}(s) = \frac{L(s)}{1+L(s)}$, which describes the transfer behaviour from the reference $\Theta_{\text{r}}$ to the state $\Theta$ under nominal conditions. Figure 6.7 shows the Bode diagram and the corresponding Nichols plot of the open-loop function $L_{\text{att}}(s)$.




(a) Bode plot.

(b) Nichols plot.

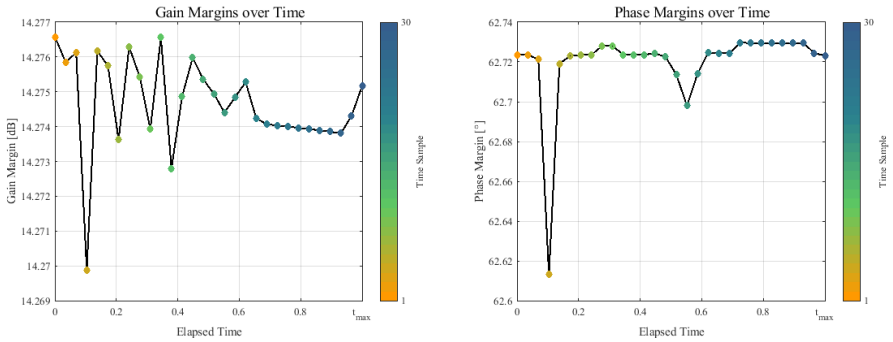
Figure 6.7: Bode and Nichols diagrams of the open-loop function $L_{\text{att}}(s)$ of the isolated attitude controller for 30 nominal discrete time samples along the trajectory.

Most noticeably in the frequency regime above 0.01 $\frac{\text{rad}}{\text{s}}$, the linearising feedback loop's reduced system dynamic dispersion is still present. Below this threshold, in the low-frequency domain, however, model mismatches in the linearisation loop propagate to the open-loop function. The Nichols plot in figure 6.7 displays the frequency regime where the linearisation is most accurate from 0.01 $\frac{\text{rad}}{\text{s}}$ onwards. The red ellipse indicates the region of stability deterioration where the gain and phase margins presented in section 6.1 are no longer upheld. The loop response can be seen to remain outside of this region, both in the gain- and phase-only margins, which correspond to the distances at the crossover frequencies for gain and phase, and in the combined margins. Because the phase is periodic, the crossover frequency, here defined at 540°, is equivalent to the -180° crossover frequency. The gain- and phase-only margins are further quantified using a MATLAB script, which extracts the evolution and worst-case results. The margin evolution for the nominal open-loop function of the purely attitude-controlled channel in the pitch plane is shown in figure 6.8.

Both margins vary only slightly in time. This trend is also observable in the Nichols plot in the limited dispersion of the sampled systems. An outlier can be observed in both margin evolutions which also appears in the Nichols plot of the system. It is likely a result of the calculation of the stability derivatives based on discontinuous data, which results in very large stability derivatives at that specific sample, artificially increasing the contribution of the aerodynamic term. The minimum margins further occur at the LTI along the trajectory that is least affected by the linearisation loop in the pole-zero maps in figure 6.5. Here the effect of model mismatches is directly visible, though admittedly the effect is minimal for this first scenario, which likely shows mostly numeric variation of the margins over the course of the trajectory.

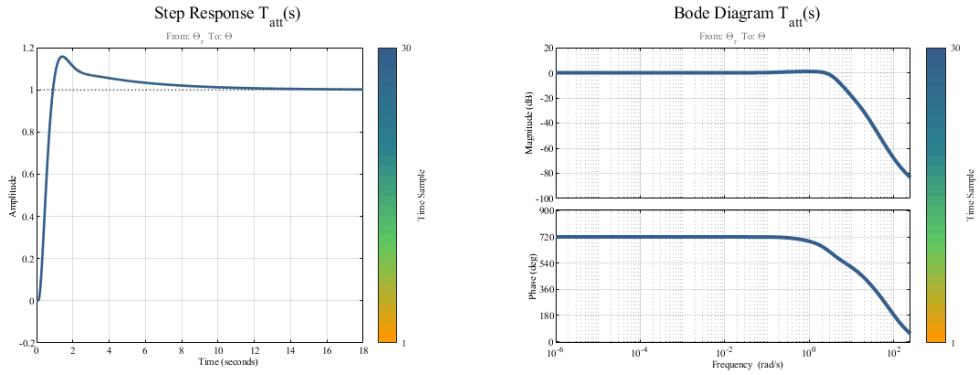
Finally, the time response of the closed-loop system is analysed by applying a unit step to the transfer behaviour $T_{\text{att}}(s) = \frac{\Theta}{\Theta_{\text{r}}}$. The result is shown in figure 6.9a.

Here, the strength of an NDI-based architecture is most prominent. The strongly dispersed open-loop plant response in figure 6.4 is both stabilised and unified into a single behaviour across




(a) (b)

Figure 6.8: Phase (a) and gain (b) margin evolution for the pitch plane attitude loop transfer function $L_{\text{att}}(s)$ for 30 nominal discrete time samples along the trajectory.



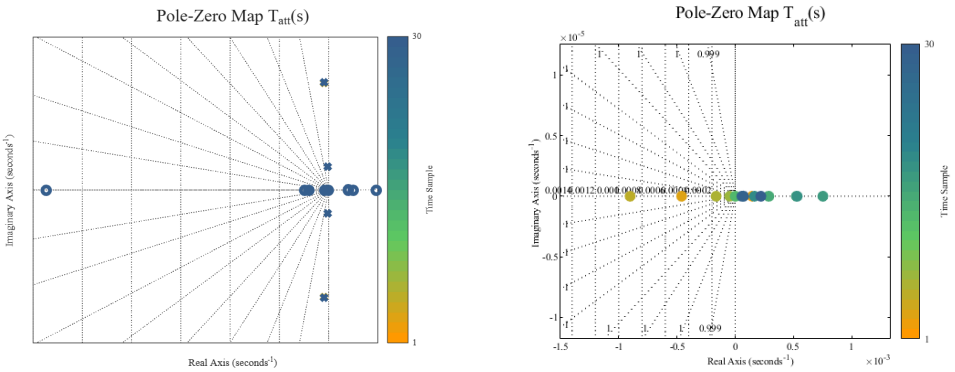

(a) Step response. (b) Frequency response.

Figure 6.9: Time and frequency response evolution for the pitch plane attitude closed-loop transfer function $T(s)$ for 30 nominal discrete time samples along the trajectory.

all flight regimes. Under these strictly nominal conditions without uncertainty or perturbation, the linearisation loop provides a simple LTI which the static linear controller stabilises globally. Comparing this response to that of the pure second-order system with the natural frequency $\sigma\omega_{\text{n,TVC}}$ and the damping ratio $\zeta$ which lies at the core of the control design shows that the response is not exactly the same. The NDI-based implementation has a longer settling time and a larger overshoot. Both can likely be attributed to the closed-loop zeros of the system. To investigate this relation further, the Bode diagram of the system closed-loop transfer function $T_{\text{att}}(s)$ is shown in figure 6.9b. Here, the roll-off frequency of the closed-loop system comes very close to the desired natural frequency, which underlines the methodology used to find the error control design presented in chapter 4.

To assess the stability of the closed loop a bit more precisely, figure 6.10 depicts its pole-zero map in an overview and a detailed view around the origin. It immediately becomes clear that not all poles lie in the left half of the complex plane. The unstable poles can be attributed to the neglected translational dynamics in the linearising and stabilising loops. This instability only affects the transfer behaviour from the reference $\Theta_{\text{r}}$ to the directional drift state $z$ though, making it negligible for the current investigation. When the zeros for the transfer behaviour in the attitude channel are also taken into consideration, it can be seen that they cancel exactly with the unstable poles as shown in figure 6.10b. This is why the attitude channel is stable even though unstable poles remain. Besides these unstable poles, it can be observed that the attitude poles are moved into the left half of the complex plane, with none remaining in the right half of the complex plane. This is consistent with the stable step response seen in figure 6.9a.

In light of the results obtained for the evolution of the simplest case, considering only the control of the attitude channel, the fundamental applicability of the NDI-based architecture to this part of the dynamics is verified. At this level of complexity, however, and without considering




(a) All poles and zeros.

(b) Detail view.

Figure 6.10: Pole evolution for the pitch plane attitude closed-loop transfer function $T(s)$ for 30 nominal discrete time samples along the trajectory.

robustness or disturbance rejection, the validity of the design against the requirements presented at the beginning of this chapter can hardly be judged. Therefore, the design is subsequently tested for the control of two degrees of freedom in the linear verification framework.

### Closed Loop Evolution: Cascaded Loops

When directional drift control is introduced into the nominal pitch plane attitude control architecture analysed above, the plant transfer behaviour and most of the inner loop dynamics remain unchanged such that the analysis in this extended scenario begins with the investigation of the transfer behaviour of the linearisation loop functions of the outer loop $P_{\mathrm{v,trans}}(s) = \frac{z}{\delta_{\mathrm{y}}}$. Figure 6.11 displays a detail view around the origin of the pole-zero map of the outer pseudo-linearised system. Since the control architecture is cascaded, the attitude channel is already fully stabilised when the pseudo control input $v_{\mathrm{trans}}$ enters the inner loop.

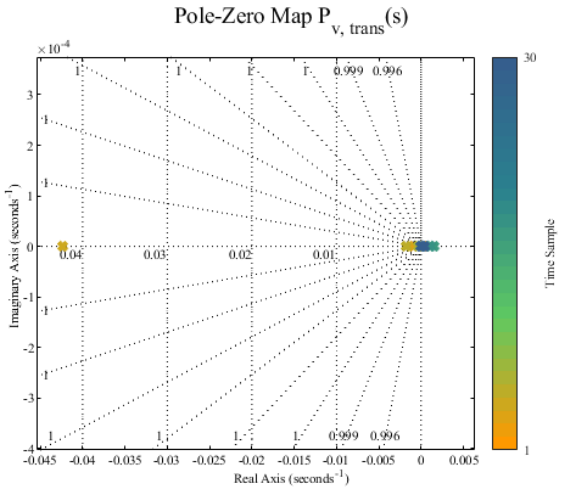
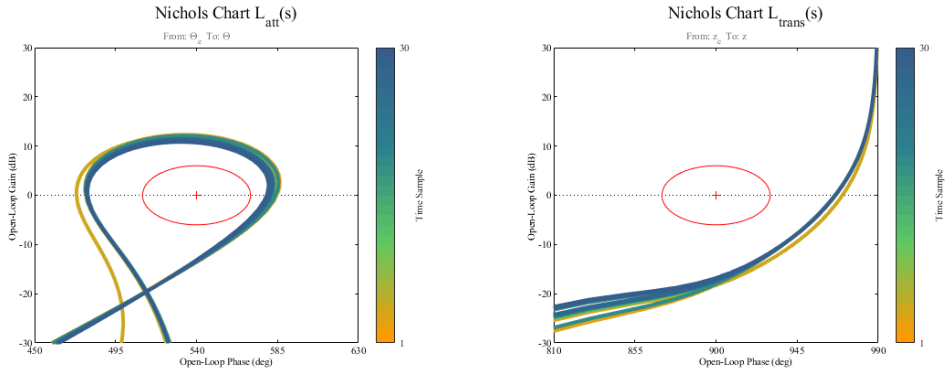



Figure 6.11: Pole-zero map of the directional drift pseudo-linearised plant $P_{\mathrm{v,trans}}(s)$ for 30 nominal discrete time samples along the trajectory.

When the translational poles in figure 6.11 are compared to the poles around the origin of the attitude solution in figure 6.10b, it can be seen that the magnitudes are drastically reduced. Like before, the linearisation loop moves the poles towards the origin, approximating the double integrator. One outlier can be seen to be less affected by this linearisation effort. It had previously caused the lowest margins in the attitude solution, and continues to likely originate from the calculation of the stability derivatives.

Moving from the effect of the outer linearising loop to the stabilising loop, figure 6.12 depicts




(a) Attitude.

(b) Translation.

Figure 6.12: Nichols diagrams of the open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ of the cascaded controller for 30 nominal discrete time samples along the trajectory.

the nominal attitude and translational frequency responses for excitations on the pitch reference $\Delta\Theta$ and the directional drift input $z_{\text{r}}$, respectively, in the inner and outer open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ as Nichols plots. None of the LTIs along the trajectory cross the low-margin region, indicating good margins for nominal conditions. It can be seen however that the outlier exhibits significant deviation from the other cases. To better quantify this result, the gain and phase margin evolutions over the trajectory for the nominal inner and outer open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ are presented in appendix B.1 and the worst-case results are shown here in table 6.1.

Table 6.1: Worst gain and phase margins along the trajectory for attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ based on the corresponding frequency responses under excitation of the $z_{\text{r}}$ signal.

| **Response Type** | **Margin** | **Trajectory Point** |
|---|---|---|
| Outer Loop $L_{\text{trans}}(s)$ | | |
| Gain | 16.803 dB | 30 |
| Phase | 65.922° | 18 |
| Inner Loop $L_{\text{att}}(s)$ | | |
| Gain | 13.142 dB | 30 |
| Phase | 38.993° | 30 |

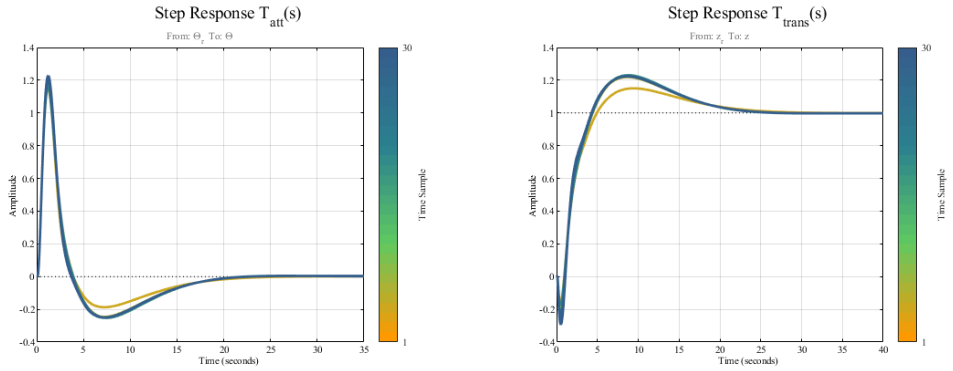
The step responses of the nominal closed-loop transfer functions $T_{\text{att}}(s)$ and $T_{\text{trans}}(s)$, which describe the transfer behaviour from the respective reference signal to the state, are depicted in figure 6.13. To produce this result, a unit step is applied to the reference signals $\Delta\Theta$ or $z_r$. The attitude response acts such that the outer loop can asymptotically approximate its reference signal through the passed outer loop control output $\Delta\Theta$. Due to the dynamic response of the attitude loop to the commanded $\Delta\Theta$, an expected undershoot at the beginning of the step response of the outer loop can be observed. As was the case for the isolated inner loop, the application of the stabilising loops causes the time responses of the system across the entire trajectory to have the same shape, excluding the outlier. This showcases once more the ability of the linearisation loops to provide a pseudo-linear system to which a time-invariant controller can be applied that stabilises the system across the entire trajectory.

It can further be observed that the outer loop peak time $t_{\text{p,trans}}$ is about 8.5s. The peak time is the time at which the maximum overshoot occurs and it can be used to infer a second-order system's natural frequency according to the relation

$$t_{\text{p}} = \frac{\pi}{\omega_{\text{d}}} \tag{6.1}$$

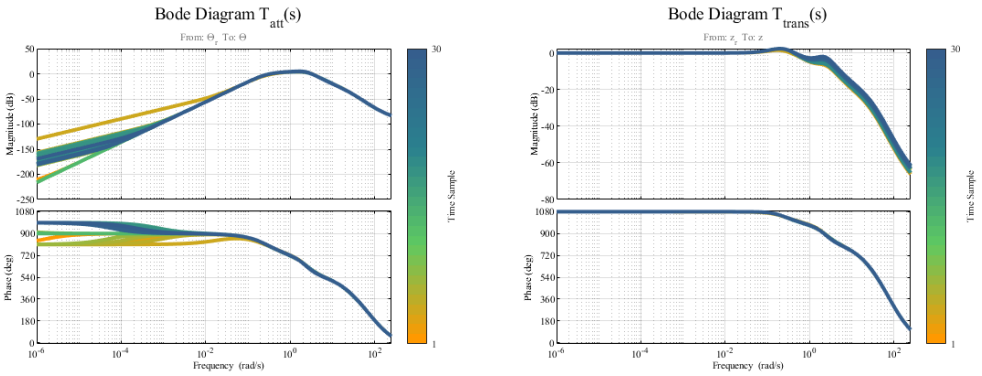
where $\omega_{\text{d}}$ is the damped natural frequency which is related to the natural frequency of a second-order system by $\omega_{\text{d}} = \omega_{\text{n}}\sqrt{1-\zeta^2}$ [50]. Using these relations, the time-scale separation can be

tested by comparing the natural frequencies of the inner and outer loop dynamics in the linear verification framework. From the outer loop peak time $t_{\mathrm{p,trans}} \approx 8.5\mathrm{s}$ follows a natural frequency of around $\omega_{\mathrm{n,trans}} \approx 0.523\frac{\mathrm{rad}}{\mathrm{s}}$ assuming the damping ratio to be $\zeta = 0.7071$. From figure 6.9a the inner loop peak time can be found to be around $t_{\mathrm{p,att}} \approx 1.5\mathrm{s}$. This results in an inner loop natural frequency of around $\omega_{\mathrm{n,att}} \approx 2.962\frac{\mathrm{rad}}{\mathrm{s}}$, which is roughly one order of magnitude faster than the outer loop.




(a) Attitude. (b) Translation.

Figure 6.13: Time response evolution for the cascaded closed-loop transfer functions $T_{\mathrm{att}}(s)$ and $T_{\mathrm{trans}}(s)$ with a unit step on the translational reference signal for 30 nominal discrete time samples along the trajectory.



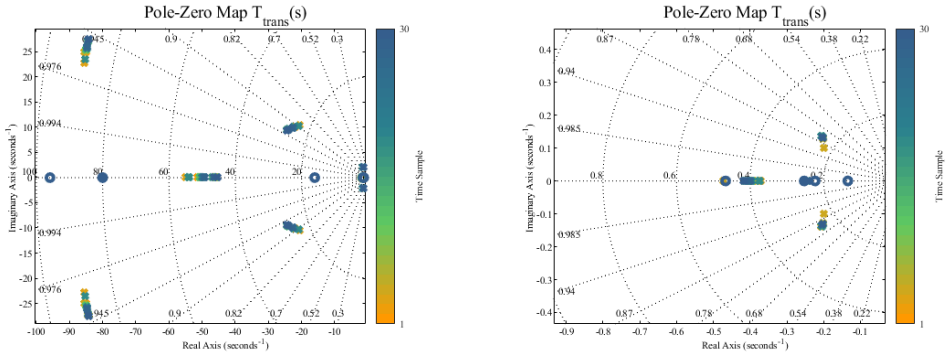

(a) Attitude. (b) Translation.

Figure 6.14: Frequency response evolution for the cascaded closed-loop transfer functions $T_{\mathrm{att}}(s)$ and $T_{\mathrm{trans}}(s)$ with a unit step on the translational reference signal for 30 nominal discrete time samples along the trajectory.

To add to the analysis of the time-scale separation of the system, figure 6.14 shows the Bode diagrams for both closed-loop transfer functions under excitation of the reference signals $\Delta\Theta$ and $z_{\mathrm{r}}$ respectively. In figure 6.14b, it can be observed that the outer loop natural frequency, the first of three observable roll-off frequencies, is around $0.5\frac{\mathrm{rad}}{\mathrm{s}}$ which indeed is around one decade slower than the observed natural frequency of the inner loop. It is represented by the second roll-off around $2.5\frac{\mathrm{rad}}{\mathrm{s}}$ and can also be found in the Bode diagram of the isolated attitude loop analysis in figure 6.9b. These observed natural frequencies fit well with the analytical solutions found above and further attest to a correctly time-scale-separated system. The last roll-off in figure 6.14b is unaddressed up to this point. It occurs due to the actuator dynamics and is observed to be about one decade faster than the inner loop dynamics.

The deviations from the time-scale separation of one decade chosen during the design phase are due to the fact that the analysed systems are not ideal. Here, the largest downside of the linearisation method used to extract the LTIs becomes apparent. By using the MATLAB function `linearize()` all dynamics along an input-output map are linearised and considered in the extracted

state-space model. The filters and simulation inherent objects like rate transitions have states in the result and induce their own overlaying dynamics. Furthermore, the design of the gains of both loops assumed their reference signals in the first and second derivatives to be zero. This is not always the case, especially for the inner loop, as its reference signals are constructed from the outer loop reference correction $\Delta\Theta$. The differentiation and filtering of this signal likely influences the inner loop transfer behaviour and causes the slight deviation from the desired natural frequency and time-scale separation. Because the margins are within the required ranges and the time-scale separation remains about one decade, the introduction of these additional dynamics does not pose a critical risk.




(a) All poles.

(b) Detail view.

Figure 6.15: Pole evolution for the pitch plane cascaded closed loop for 30 nominal discrete time samples along the trajectory.

The additional dynamics also become visible in the overall system pole-zero maps in figure 6.15, leading to a larger than expected number of poles. As before, a global map of the poles and zeros is shown in figure 6.15a and a detail view close to the origin is shown in figure 6.15b. The full order system is expected to have six poles, two for the inner loop dynamics, two for the outer loop dynamics and two for the actuator dynamics. The rest corresponds to other elements along the signal path like filters or delays, which makes it difficult to identify individual contributors. Regardless, among all poles none remain in the right half of the complex plane after applying the outer stabilising loop, making the system stable across the entire trajectory. This fits well with the asymptotically stable behaviour of the time responses.

A discussion of the most important results obtained with the same trajectory evolution analysis carried out on the non-linear system equation-based NDI architecture is presented in appendix B.2. It arrives at the conclusion that during the ascent phase, for which the utilised plant model remains representative, the obtained results are equivalent. The small discrepancies which arise stem from the approximation inaccuracies within the linear model, which are minimal for the initial test flight scenario of the CALLISTO vehicle. It is expected that for the demonstration scenarios with higher maximum velocities and larger aerodynamic contributions to the system dynamics, the difference between the two implementations would become more apparent.

Though the linear analysis of the system dynamic evolution does visualise trends of the effects of the inversion-based architecture elements over the course of the trajectory in nominal conditions, the robust linear analysis under uncertain conditions really investigates the method's weakest point, a mismatch between the plant and the linearisation loops. It is presented in the following.

### 6.2.2 Robustness Analysis

The robustness analysis is carried out on the pitch plane dynamics and the sample corresponding to the maximum dynamic pressure $Q_{\max}$ along the trajectory because it contains the largest contribution of the aerodynamic forces and torques to the system dynamics. Classically, the corresponding aerodynamic term has the largest influence on the stability of the system, making this a critical phase with high stress on the accuracy of the linearisation. The time of maximum dynamic pressure occurs during the ascent, which falls in line with the disclaimer that the plant model remains most representative during this phase due to the identified modelling error. The yaw plane

is again not presented here because the differences for the investigated flight regimes were found to be negligible. Figure 6.16 displays once more the test flight mission profile for the initial flight demonstration of the CALLISTO vehicle with the sample of maximum dynamic pressure marked on all graphs.

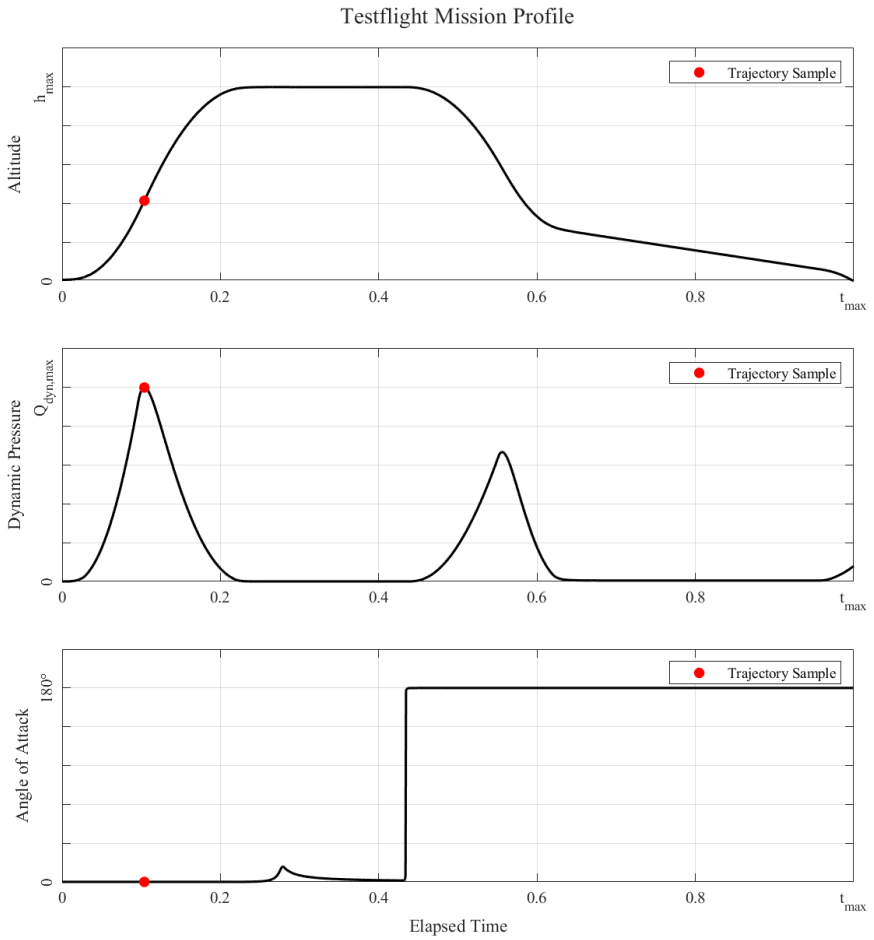



Figure 6.16: Mission Profile of the first low-energy test flight demonstration scenario, $Q_{\max}$ marked as the nominal system choice.

The investigated uncertainty ranges are chosen to be similar to those in the CALLISTO mission. To give an introduction to the effects of these uncertainties on the plant, figure 6.17 displays the pole-zero map of each plant transfer behaviour $P(s)$, generated from 20 randomised sets of uncertainties and the nominal system. The uncertain systems are shown in blue whereas the nominal system poles are marked with a red marker. As explained before, the ascent phase leads to unstable attitude and stable drift dynamics, which are well represented by the nominal poles. It can be seen that the uncertain cases cluster around the nominal case while remaining on a straight line towards the origin.

How the linearisation loops are impacted by the induced plant uncertainties can be seen in figure 6.18. It depicts the pole-zero maps after applying the attitude linearisation loop in figure 6.18a and after applying the translational linearisation loop in figure 6.18b. As expected, the magnitudes of the real parts of the poles reduce, yet it is of note, that after applying the inner linearisation loop, for some sets of uncertainties the attitude poles are shifted to the left half of the complex plane, stabilising them. Likewise, some of the formerly stable translational poles are moved to the right half of the complex plane. This by-product of the linearising feedback loop is somewhat unexpected as it should merely cause the approximation of the pseudo-linear transfer behaviour in the form of the double integrator.

Moving again from the linearisation loop's to the assessment of the stabilising loop's robustness,

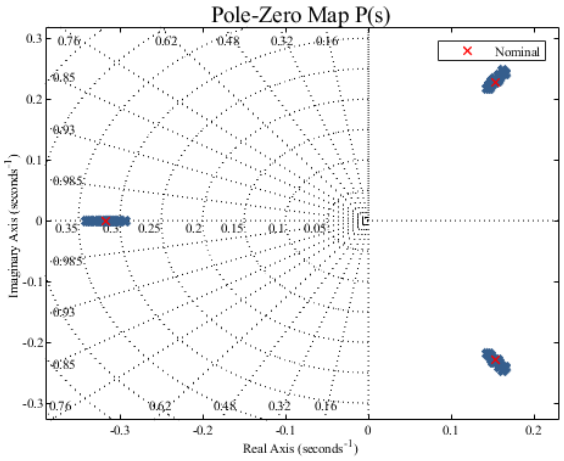
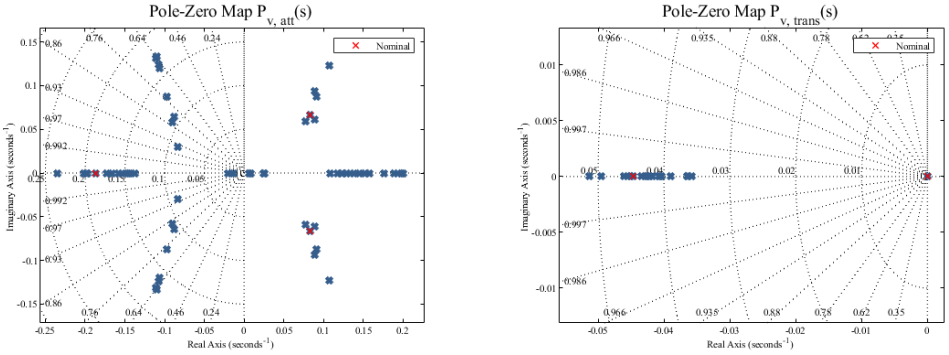



Figure 6.17: Pole-zero map around the origin of the uncertain linear reference plant $P(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal poles are marked in red.




(a) Attitude.

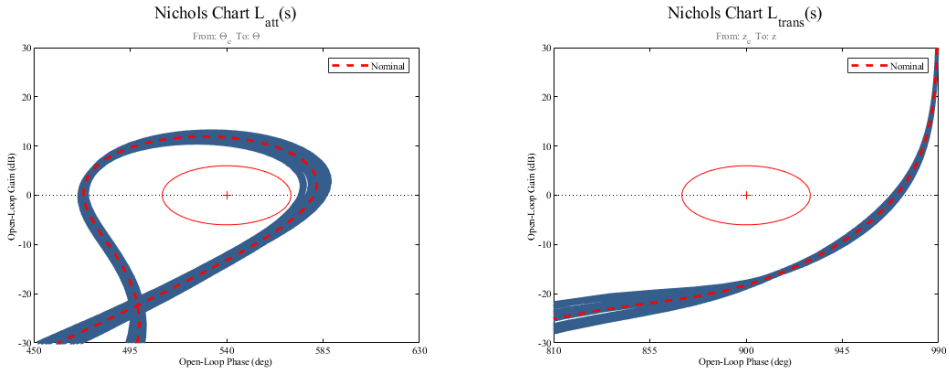
(b) Translation.

Figure 6.18: Pole-zero map around the origin of the uncertain pseudo-linear plants $P_{\mathrm{v,att}}(s)$ and $P_{\mathrm{v,trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal poles are marked in red.

figure 6.19 displays the Nichols diagrams for the outer and inner open-loop functions $L_{\mathrm{att}}(s)$ and $L_{\mathrm{trans}}(s)$ with applied excitations to the pitch angle reference $\Delta\Theta$ and the drift reference signal $z_{\mathrm{r}}$. The region denoting the margin requirements can be seen not to be crossed by any of the uncertain sets of variables, attesting to the system's robustness, even when applied to an uncertain plant model. The effect of model mismatches is evidently small enough in the investigated cases for the NDI-based implementation to remain valid. Table 6.2 gives the worst-case results for the gain and phase margins under the 20 considered cases at $Q_{\max}$ in a more quantified manner.

The worst gain and phase margins are within the requirements, underlining the behaviour depicted in the Nichols charts of figure 6.19. When the worst-case results are compared to the nominal worst-case margins along the trajectory in table 6.1, it seems that in the case of the outer loop, the found uncertain worst margins are actually further away from instability compared to the found worst margins along the trajectory under nominal conditions. This is not entirely accurate, because the investigated sample at $Q_{\max}$ does not correspond to the sample with the worst margins along the trajectory in nominal conditions. The figures in appendix B.1 underline this, as for the cascaded structure the margin evolution is not just numeric in nature. To fundament that the overall worst-case margins are really found, the third operating mode of the analysis suite *absWorstMargins* is utilised.

Before moving to the search for the global worst case, however, the analysis of the system closed-loop behaviour $T_{\mathrm{att}}(s)$ and $T_{\mathrm{trans}}(s)$ at $Q_{\max}$ under uncertain conditions is done by investigating



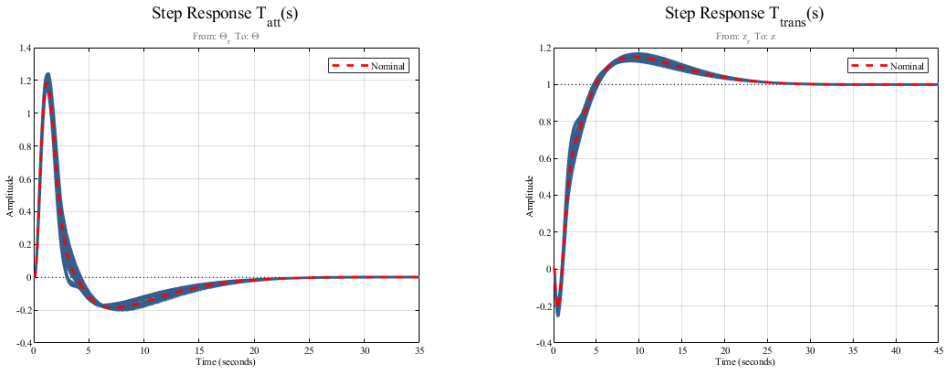

(a) Attitude. (b) Translation.

Figure 6.19: Nichols diagrams of the uncertain open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal responses are marked in red.

Table 6.2: Worst gain and phase margins among 20 sets of uncertain variables for attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ at $Q_{\max}$ based on the corresponding frequency responses under excitation of the $z_{\text{r}}$ signal.

| **Response Type** | **Margin** | **Uncertainty Sample** |
|---|---|---|
| Outer Loop $L_{\text{trans}}(s)$ | | |
| Gain | 17.535 dB | 19 |
| Phase | 68.290° | 10 |
| Inner Loop $L_{\text{att}}(s)$ | | |
| Gain | 11.828 dB | 12 |
| Phase | 34.091° | 19 |

the step responses depicted in figure 6.20. It can be seen that the model mismatch between linearisation loops and plant causes the step responses to appear more dispersed. Though all systems remain stable, as already indicated in the preceding margin analysis, the time-invariant stabilising feedback loops act on an inaccurately linearised pseudo LTI which they were never designed for. In some cases the step response even exhibits slight damped oscillations, which can be expected to be worse for unconsidered, extreme combinations of uncertainties. As long as these oscillations remain damped, they do not pose a threat to the system's stability.




(a) Attitude. (b) Translation.

Figure 6.20: Step responses of the uncertain closed-loop functions $T_{\text{att}}(s)$ and $T_{\text{trans}}(s)$ at $Q_{\max}$ for 20 randomised uncertain sets of variables. The nominal responses are marked in red.

To take into account that the preceding robustness analysis only covers the sample corresponding to the maximum dynamic pressure, the results of the global minimum margin search for both the outer and inner open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ and the isolated inner loop are

presented in table 6.3.

Table 6.3: Global worst gain and phase margin search results among 20 sets of uncertain variables at 30 discrete points along the trajectories. Applied to the attitude and translational open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ based on the corresponding frequency responses under excitation of the $\Delta\Theta$ and $z_{\text{r}}$ signals.

| **Response Type** | **Margin** | **Trajectory Point** | **Uncertainty Sample** |
|---|---|---|---|
| Isolated Inner Loop $L_{\text{att}}(s)$ | | | |
| Gain | 13.321 dB | 4 | 2 |
| Phase | 58.566° | 1 | 3 |
| Cascaded Outer Loop $L_{\text{trans}}(s)$ | | | |
| Gain | 15.939 dB | 30 | 19 |
| Phase | 63.898° | 1 | 10 |
| Cascaded Inner Loop $L_{\text{att}}(s)$ | | | |
| Gain | 11.285 dB | 30 | 12 |
| Phase | 31.298° | 30 | 19 |

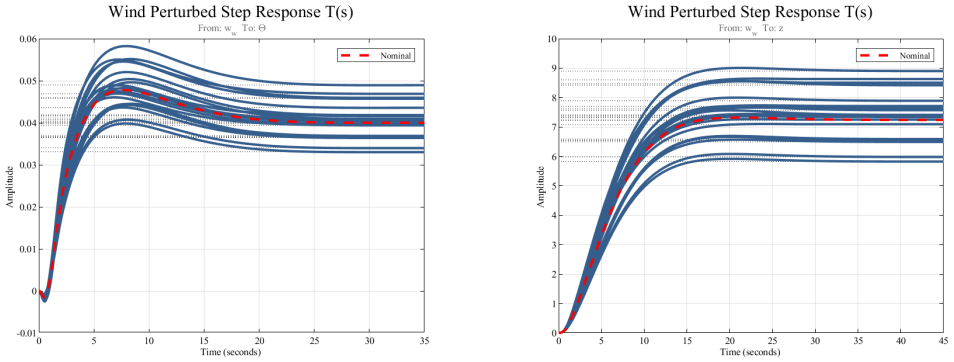
Within the investigated sets of uncertainties, and under the chosen discretisation in time along the trajectory, the worst gain and phase margins for each of the open-loop functions remain well within the desired ranges. Interestingly, the samples which produce the lowest values correspond to the initial and final points on the trajectory. This can be understood if the margin evolution over time in nominal conditions for the cascaded loop structure is considered. The evolution is presented in appendix B.1. Both gain and phase margin evolutions for the inner loop and the gain margin evolution for the outer loop follow the thrust magnitude profile over the course of the flight. The last instance of flight right before touchdown has a low thrust magnitude. Coupled with an observed overall declining behaviour, the last sample becomes the sample in the nominal cases with the lowest margins. If uncertainties are now applied, the margin variations over time outweigh the margin variations induced by the uncertainties, making the last sample still the closest to instability. For the phase margin evolution in the outer loop, shown in figure B.1d, the dependence on the thrust magnitude is inverted. Because now the first instances of flight have the lowest margins across the trajectory, the uncertainties determine which specific sample is the worst case. In the investigated 20 cases, the first sample was the resulting worst case.

It should be kept in mind that a low number of uncertain sets of variables were generated and analysed to showcase trends in the system's response under uncertainties and assess the feasibility of the presented design. This set most likely does not contain the absolute worst cases. Since the gain and phase margins retain merely a small distance from the required margins in the worst presented cases, especially in the cascaded inner loop, it is probable another worse set could be found that causes the requirements to be violated. Beyond this, the results are only representative for real test flight conditions during the first demonstration scenario of CALLISTO in a limited manner, as the used linear reference model and the linearisation loop alongside it are based on a crude approximation of the real system dynamics.

### 6.2.3 Disturbance Rejection

Analysing the effect of a wind disturbance with the linear reference model is less elaborate than in the non-linear simulator. Here, it is analysed by injecting the maximum expected wind magnitude at the largest altitude and assessing the step response for a number of uncertain plants. Again, 20 uncertain sets of variables are generated and analysed. The nominal system corresponds again to the time of maximum dynamic pressure. The result is shown in figure 6.21.

In the attitude response, only a small steady-state deflection of around 0.04° is observed. For different sets of uncertainties this deflection varies within a range of around 0.01°. The drift channel shows a much greater response of around 7m with a variation of up to 2m under the considered sets of uncertainties. When this result is compared to the steady-state errors in the position channel obtained in the non-linear simulator, similar magnitudes are observed.



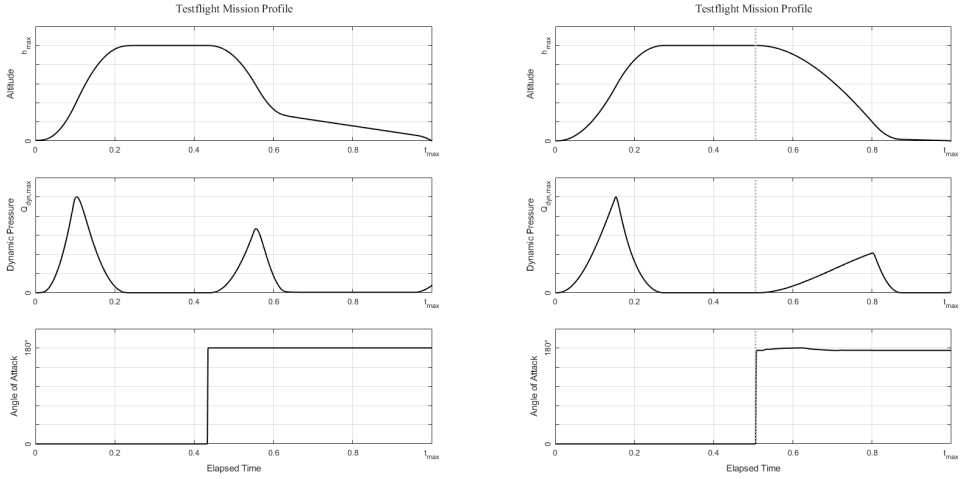

(a) Attitude.

(b) Translation.

Figure 6.21: System response of the planar controller of the linear reference model for a $20\frac{m}{s}$. wind disturbance applied to the $Q_{\max}$ uncertain plant for 20 randomised uncertain sets of variables. The nominal responses are marked in red.

## 6.3 Nonlinear Simulation Results

The non-linear simulator represents the most accurate environment for the benchmark analysis of the time-domain performance of the NDI-based control architecture and is as such of greater relevance to its verification in this domain than the linear verification framework.




(a) Initial demonstration flight.

(b) Second demonstration flight.

Figure 6.22: Relevant parameters from the reference trajectory for the first and second flight demonstration scenarios of the CALLISTO vehicle. Normalised altitude, dynamic pressure and angle of attack over time are shown. A grey, dotted line indicates the guidance and control phase change from hover to landing in the second scenario.

Two scenarios are investigated: The initial low-energy flight demonstration of the CALLISTO vehicle, whose normalised mission profile is shown in figure 6.22a. The normalisations of both presented trajectories are done with respect to their corresponding maximum values. The first scenario features a static vehicle configuration and can therefore be controlled by the proposed NDI-based architecture for the entire flight duration. The second demonstration features a higher maximum altitude and flight configuration changes and is depicted in normalised form in figure 6.22b. It is introduced to judge how the control solution performs under more dynamic conditions. The maximum velocities are the most impactful change here. The system response of an RLV is largely dominated by its aerodynamics, and since higher velocities directly translate in the

magnitudes of the aerodynamic forces and torques, the system's stability is stressed more during the second scenario. However, one important restriction originates from the implementation of the NDI-based control architecture in the non-linear simulator. The aerodynamic database, which is interpolated to find the aerodynamic coefficients to build the linearising feedback loop in both attitude and translational controllers, is fixed to the vehicle configuration. Even though the second scenario starts with unfolded landing legs in the same configuration as the first, now the aerodynamic control surfaces are deployed after the hover phase. Therefore, the inversion-based controller is only applied to the ascent and part of the hover phase during the second scenario. An indication is added to each plot in figure 6.22b to highlight the flight configuration change, from which point on another control architecture is employed. Accordingly, the presented results for the second scenario only contain the tracking performance up to this point in time.

### 6.3.1 Initial Flight Demonstration Scenario

To give a first impression of the results obtained in the non-linear simulator with the NDI-based control solution, the normalised errors in position, velocity, angles and angular velocities in pitch and yaw are presented in nominal conditions for the first demonstration flight. Nominal conditions refer here to a scenario where no wind perturbations are applied and all uncertain parameters remain at their nominal value. Figure 6.23 displays the results for this scenario normalised with respect to the individual requirements of each quantity. As the longitudinal and roll channels are not controlled by the TVC actuators, they are omitted. The bounds for each degree of freedom are marked in red. They are defined in the scope of this thesis and are similar to the values used for the benchmark mission.

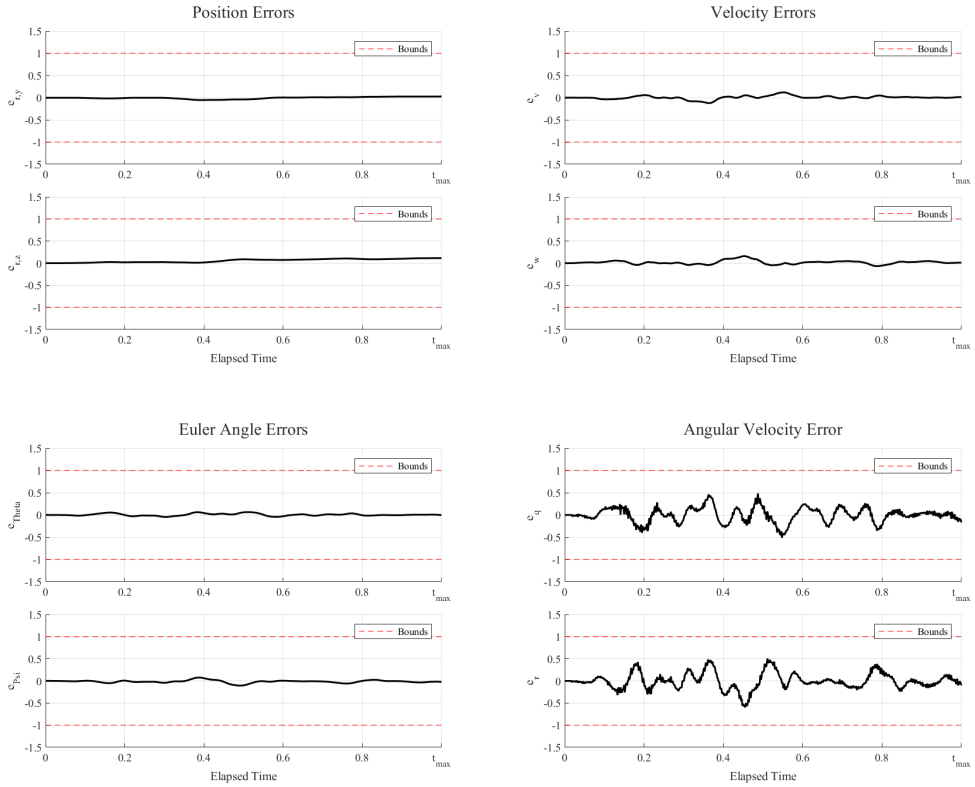



Figure 6.23: Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of the nominal, initial demonstration of CALLISTO.

Since this nominal case is a mere approximation of the unknown conditions during the real flight, the effect of uncertainties has to be assessed. To showcase how the NDI-based control solution performs under uncertain conditions, a Monte Carlo campaign with 50 unique, randomly sampled sets of uncertainties, which are applied to the parameters described in section 5.1, is

conducted. Added to those 50 randomised samples are three corner cases, which were previously identified to have the largest wind velocities in a bin of 250 random sets of uncertainties. They cover the largest wind velocities along two geographic directions as well as the largest wind velocity norm. With these 53 cases, the validity of the control implementation is further assessed.

Figure 6.24 shows once more the normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes, now under uncertain conditions.

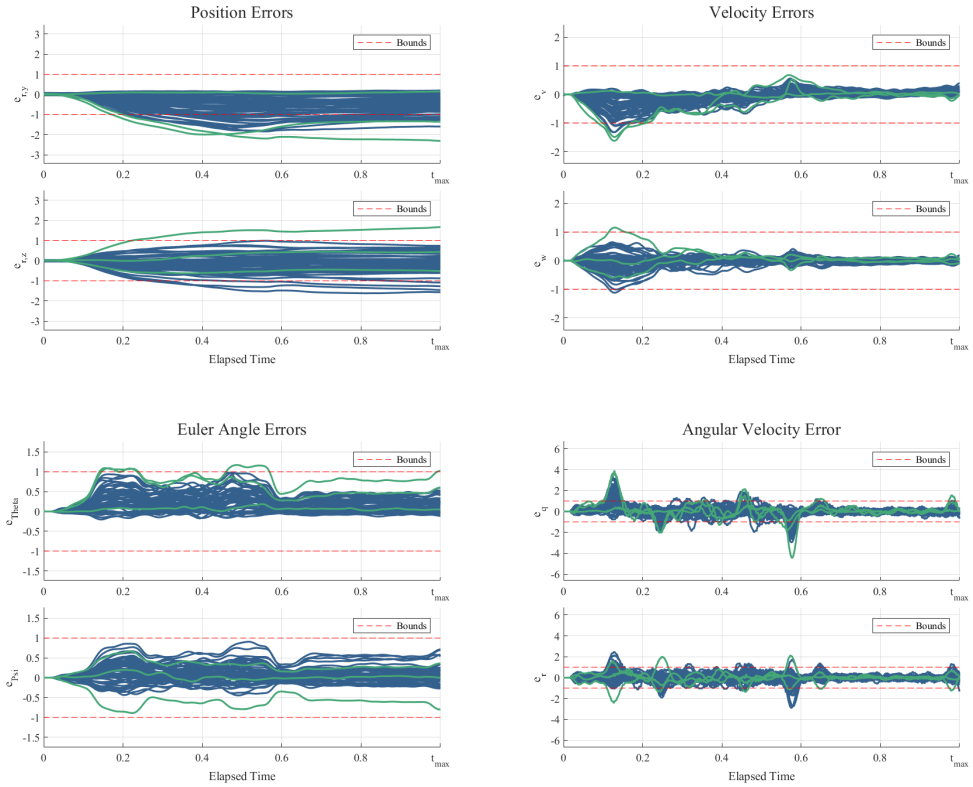



Figure 6.24: Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified corner cases (green) for the initial demonstration of CALLISTO.

It can be seen that under uncertain conditions the bounds are not always upheld and especially in the position errors, deflections lead to a constant offset rather than a gradual decline after a perturbation has occurred. This can be attributed to the structure of the error controller. It contains contributions for the proportional and derivative control of the drifts and angular errors $y_e, z_e$ and $\Theta_e, \Psi_e$. With this setup, it is not guaranteed that the steady-state accuracy of the system response is such that no offset remains. If an integral error controller were to be introduced, such steady-state errors could reduce, and the system would likely reach an equilibrium closer to the reference. Such penalising behaviour can be observed in the velocity and angular velocity errors $v_e, w_e$ and $q_e, r_e$. This is because these two physical properties indirectly have an integral contribution. Integrating the velocity and angular velocity errors leads to the position and angular errors. Since these are controlled with a PD structure, their derivative contribution becomes the proportional contribution and equally the proportional contribution becomes an integral control on the velocities. Accordingly, the steady-state accuracy of the system response in the velocities is much better. Another observed effect is a bias in the progression of both the lateral drift $y_e$ and the pitch angle error $\Theta_e$. This can be attributed to typical launch site wind profiles. For the targeted launch site, an inherent bias in the wind direction persists and this visibly propagates to the state response.

As presented in the previous section, a measure of the quality of a control solution is the resulting control action. Figure 6.25 presents the normalised TVC deflection angles which result for the uncertain cases of the MC campaign introduced above. The values are normalised with respect to the TVC system saturation limit. Mostly, around 50% of the saturation limit is reached

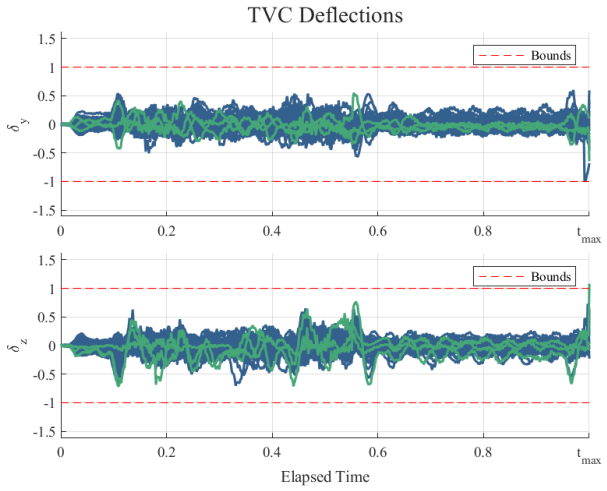



Figure 6.25: Normalised TVC deflection angles in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified corner cases (green) for the initial demonstration of CALLISTO.

in the worst cases covered by the campaign. An exception can be observed for the last instance of flight. In some cases, the landing leads to a strong increase in TVC control action to the point of saturation. A potential explanation lies in the dependency of the inversion law on the mathematical model. Because the last instance before touch down could cause interactions with the ground, which are not considered in the model, a mismatch occurs which leads to the steep increase in control action. Because it takes time for the control action to propagate to the states, a resulting deflection is not observed in the errors. This behaviour should be kept in mind as a potential risk for future implementations.

Because of the relatively low number of MC simulations, with 53 sets of uncertainties, it cannot be guaranteed that the cases investigated in this work contain true worst cases. The coupling between the numerous uncertain parameters is complex and unpredictable and to increase the confidence in the control solution in uncertain scenarios, larger campaigns should be carried out.

### 6.3.2 Second Flight Demonstration Scenario

So far, only the initial flight demonstration of CALLISTO has been considered as a reference scenario. The second flight scenario, with a higher hover altitude and configuration changes, is presented in the following. It should be noted that the linear analysis presented at the beginning of this chapter does not cover this test flight. Merely the initial test flight is showcased and the margins, time and frequency characteristics extracted for this flight are not directly transferable to the second scenario. Therefore, only the nominal results for the ascent and hover phases during which the NDI-based control architecture is applied are presented here, and the results of the MC campaign are documented in appendix B.3. Without the accompanying linear analysis the robustness is difficult to interpret, leaving these results for future investigations.

Figure 6.26 displays the normalised errors obtained for the second CALLISTO demonstration under nominal conditions. The controller performs similarly to the initial demonstration flight. Larger deviations than before can be observed during the ascent at around 20% of the elapsed time for the directional drift $z$ and the corresponding velocity $w$. The angular velocity errors have a reduced magnitude with more noise compared to the first demonstration flight. In both flights, the noise is largest around the maximum velocity, making the effect of a larger aerodynamic contribution on the system dynamics visible. This nominal case merely speaks to the performance of the NDI-based control architecture where it is most potent. As long as the system model at the core of the linearisation effort remains close to the plant, the expected performance is very good. To assess the linearisation loop's behaviour under more realistic conditions, an elaborate robust analysis should be carried out. In the limited scope of this work, the linear analysis of the second scenario was not completed, providing only the same MC campaign carried out on the

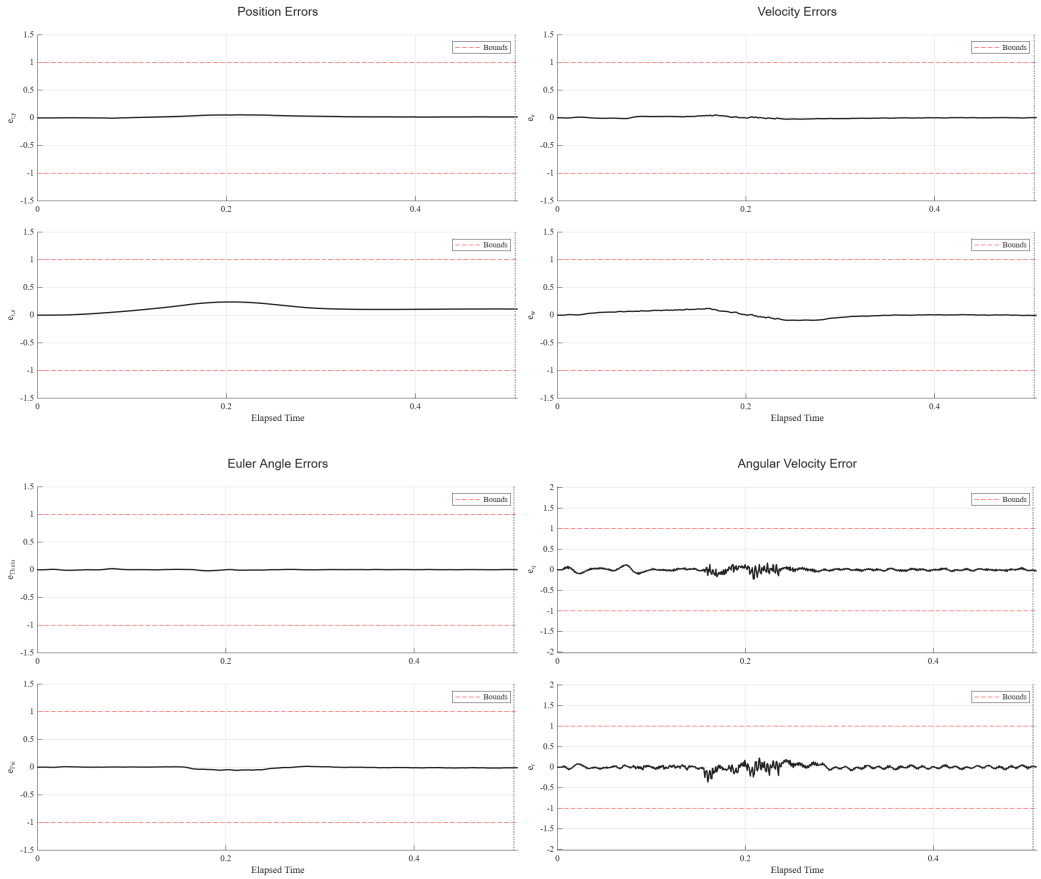



Figure 6.26: Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of the nominal, second demonstration of CALLISTO. Shown are the ascent and hover phases during which the NDI-based control architecture is applied.

second test flight in the non-linear simulator as had been done on the initial test flight before. It can be found in appendix B.3 and can be taken as a reference once a full analysis of the second flight is conducted.

# 7 Summary and Outlook

In this work, the design and analysis of a non-linear Dynamic Inversion (NDI)-based control architecture is presented and the question of whether or not it is a feasible approach for VTVL RLVs is investigated. An exemplary control architecture for TVC actuation of the CALLISTO reusable rocket demonstrator is derived and tested in a high-fidelity non-linear simulation environment and a linear verification framework. The analysis in the high-fidelity, non-linear simulator, built to support the GNC development of the CALLISTO vehicle, is used here to assess tracking, robustness and disturbance rejection of the controller for two low-energy hover scenarios. Alongside it, analysis in the linear verification framework is done to visualise trends in frequency and time response behaviour with classical linear analysis techniques. This auxiliary verification is shown to be representative only to a limited extent, yet it has been proven to be an invaluable tool for the isolation of specific behaviour in the design process of the controller.

The obtained results clearly showcase the potential of the method in eradicating the need for time-varying, trajectory-dependent control solutions. Nonetheless, its most notorious drawback, the strong dependence on the accuracy of the mathematical plant model used to derive the inversion law, poses a risk to a successful application in a real flight demonstration. Under uncertain conditions the performance of the control architecture degrades and it is expected that for more dynamic conditions, as would be present during a typical mission profile of an orbital class reusable first stage, this performance degradation would be even more drastic.

What is, however, showcased in both the non-linear simulation results and even more so in the linear verification results under nominal conditions, is the presented solution's ability to provide a uniform pseudo-linear plant under changing conditions. The linearisation loop at the heart of the NDI-based control architecture successfully approximates a double integrator in a domain where the assumptions used during the derivation of the inversion laws hold. In the proposed solution, these include small-angle approximations for the pitch and yaw angles, making it only applicable in the presented hover trajectories or during a gravity turn, where centrifugal forces and the gravity force components cancel out. Other simplifications include the neglect of terms in the rigid-body equations of motion which contain changes in moments and products of inertia, limiting the solution's accuracy during vehicle configuration transitions. Further, geometric properties of the vehicle are exploited to reduce the matrix of inertia to diagonal form. This limits the solution's applicability to slender, rotationally symmetric vehicles. In inversion-based control solutions, these simplifications play a crucial role and should be kept in mind when considering their feasibility.

An advantage of the consideration of the system's non-linear behaviour lies in the estimation of the aerodynamic forces and torques in the linearisation loop. In the derivation based on the non-linear equations presented in chapter 4, the aerodynamic coefficients are left as implicit functions of the states, allowing the classic approach of Taylor expansion combined with alpha decomposition to be circumvented to obtain the aerodynamic contributions. Not having to form the Taylor expansion retains information lost when higher-order terms are neglected.

From these considerations emerges a trade-off. An NDI-based control implementation in the presented form requires a very detailed system model, which is often difficult to acquire for space transportation systems because system identification campaigns are costly and exceedingly rare during development. Yet, especially for reusable systems with fast turnover periods, a trajectory agnostic solution could save development costs and reduce the required time between launches. The increase in complexity from the additional linearisation loops comes with a possible decrease in complexity for the formerly required different flight phase controllers.

In its current form, the proposed control solution is unlikely to outperform state-of-the-art control solutions based on linear models. The advantages of the pure NDI linearisation method, especially in combination with the rudimentary pole-placement-based controller used here, are

simply not enough to justify the decreased robustness in uncertain conditions. This conclusion cannot be drawn for inversion-based methods as a whole. There is potential in the approach for the application to VTVL RLVs and their overall feasibility is clearly demonstrated in this work.

The most promising future investigation based on the proposed solution is its expansion to an incremental formulation. INDI promises to reduce the mathematical model dependency, allowing the approach to possibly remain applicable even in high-energy flight scenarios coupled with the limited availability of system identification data in the technological domain of launch vehicles. The incorporation of measurements of the pseudo control input and the actual system input does come with its own complications, yet the result of a trajectory agnostic control solution with a static controller remains the same. More advanced variants of INDI like E-INDI and ANDI additionally allow the actuator dynamics to be considered in the plant model, increasing the accuracy of the linearisation effort.

Besides the adaptation to INDI, considering additional dynamic effects in the mathematical model could offer another means of improvement. Considering e.g. coupled dynamics between control channels by treating the pure MIMO system directly or considering non-zero products of inertia would increase the domain to which the solution is applicable. Additionally, a reconsideration of the pure MIMO system-based derivation would exploit the decoupling behaviour of inversion-based control methods. While these adaptations would extend the domain of applicability of the approach, they do not address its robustness issues.

To increase the robustness, more complex linear control design methods could be explored in the derivation of the stabilising feedback loop. The literature proposes a number of combinations of error control solutions with inversion linearisation loops outside the domain of RLVs, which also include robust design methods like $H_\infty$ or similar. Finding a comparable solution for a VTVL RLV would combine the advantages of robust control and inversion-based linearisation: a robust controller valid in a large domain independent of discrete linearisation points.

Smaller adaptations like PCH and variations of the reference signal generation method represent further opportunities to explore the inversion-based design space. It is clear that recent advancements around feedback linearisation methods provide a wide variety of different tools. Therefore, it is probable that the presented design can be evolved to become not just a feasible, but a competitive solution for VTVL RLV control.

# Bibliography


[1] German Aerospace Center (DLR), "Missions and projects: CALLISTO." `https://www.dlr.de/en/irs/research-transfer/missions-and-projects/callisto`, 2026. Accessed: 2026-08-02.

[2] A. Steinert, S. Raab, S. Hafner, F. Holzapfel, and H. Hong, "From fundamentals to applications of incremental nonlinear dynamic inversion: A survey on INDI – part I," *Chinese Journal of Aeronautics*, vol. 38, no. 11, p. 103553, 2025.

[3] J. A. Macés-Hernández, M. Sagliano, A. Heidecker, D. Seelbinder, M. Schlotterer, S. Farì, S. Theil, S. Woicke, and E. Dumont, "Ascent flight control system for reusable launch vehicles: Full order and structured H∞ designs," in *11th International ESA Conference on Guidance, Navigation and Control Systems (ESA GNC 2021)*, 2021.

[4] R. Brockhaus, W. Alles, and R. Luckner, *Flugregelung*. Berlin and Heidelberg: Springer, 3., neu bearb. aufl. ed., 2011.

[5] S. Farì, M. Sagliano, J. A. Macés-Hernández, A. Schneider, A. Heidecker, M. Schlotterer, and S. Woicke, "Physical modeling and simulation of reusable rockets for GNC verification and validation," *Aerospace*, vol. 11, no. 5, p. 337, 2024.

[6] J. A. Macés-Hernández, "Advanced nonlinear modeling and control stability analysis of sloshing dynamics in liquid-propelled reusable launch vehicles," *CEAS Space Journal*, 2026. Published online.

[7] K. Hayward, "The economics of launch vehicles: Towards a new business model," in *Yearbook on Space Policy 2015: Access to Space and the Evolution of Space Activities* (C. Al-Ekabi, B. Baranes, P. Hulsroj, and A. Lahcen, eds.), pp. 247–256, Vienna: Springer Vienna, 2017.

[8] P. Rickmers, E. Dumont, S. Krummen, J. L. Redondo Gutierrez, L. Bussler, S. Kottmeier, G. Wübbels, H. Martens, S. Woicke, M. Sagliano, J. Häseker, L. Witte, M. Sippel, W. Bauer, and H.-J. Peetz, "The CALLISTO and ReFEx flight experiments at DLR – challenges and opportunities of a wholistic approach," *Acta Astronautica*, vol. 225, pp. 417–433, 2024.

[9] A. Marwege, A. Gülhan, J. Klevanski, C. Hantz, S. Karl, M. Laureti, G. De Zaiacomo, J. Vos, M. Jevons, C. Thies, *et al.*, "RETALT: review of technologies and overview of design changes," *CEAS Space Journal*, vol. 14, no. 3, pp. 433–445, 2022.

[10] M. Dumke, G. F. Trigo, M. Sagliano, P. Saranrittichai, and S. Theil, "Design, development, and flight testing of the vertical take-off and landing GNC testbed EAGLE," *CEAS Space Journal*, vol. 12, no. 1, pp. 97–113, 2020.

[11] M. Prampolini, E. Louaas, Y. Prel, S. Kostromin, N. Panichkin, Y. Sumin, M. Osin, D. Iranzo-Greus, M. Rigault, A. Beaurain, and J.-N. Couteau, "Advanced space transportation systems, BARGOUZIN booster," *Acta Astronautica*, vol. 63, no. 1, pp. 435–447, 2008. Selected Proceedings of the 58th International Astronautical Federation Congress, Hyderabad, India, 24–28 September 2007.

[12] S. Krummen, L. E. Briese, E. Dumont, T. Ecker, M. Ertl, S. Giagkozoglou Vincenzino, A. Heidecker, J. Klevanski, A. Krieger, F. Krziwanie, H. Martens, L. Opp, M. Puigserver Rossello, J. Rathke, J. Redondo Gutierrez, B. Reimann, A. Schneider, and J. Windelberg, "CALLISTO: on the design and development of a reusable first stage demonstrator," *CEAS Space Journal*, 2026. Published online.

[13] P. Simplício, P. Acquatella, and S. Bennani, "Design and analysis of a launcher flight control system based on incremental nonlinear dynamic inversion," *Aerospace*, vol. 12, no. 4, p. 296, 2025.

[14] L. R. Hunt, R. Su, and G. Meyer, "Global transformations of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 24–31, 1983.

[15] A. Isidori, *Nonlinear Control Systems*. Communications and Control Engineering, London: Springer, 3rd ed., 1995.

[16] B. Jakubczyk and W. Respondek, "On linearization of control systems," *Bulletin de l'Académie Polonaise des Sciences, Série des Sciences Mathématiques*, vol. 28, pp. 517–522, 1980.

[17] D. Enns, D. Bugajski, R. Hendrick, and G. Stein, "Dynamic inversion: an evolving methodology for flight control design," *International Journal of Control*, vol. 59, no. 1, pp. 71–91, 1994.

[18] P. Smith, "A simplified approach to nonlinear dynamic inversion based flight control," in *23rd Atmospheric Flight Mechanics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 1998.

[19] B. Bacon and A. Ostroff, "Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2000.

[20] 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, pp. 1732–1742, 2010.

[21] E. J. J. Smeur, Q. P. Chu, and G. C. H. E. de Croon, "Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 450–461, 2016.

[22] D. Milz, M. S. May, S. F. Armanini, and G. Looye, "Stability and robustness of a tandem tilt-wing VTOL with dynamic inversion control laws," 2026.

[23] P. Smith and A. Berry, "Flight test experience of a non-linear dynamic inversion control law on the VAAC Harrier," in *Atmospheric Flight Mechanics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2000.

[24] F. Grondman, G. Looye, R. O. Kuchar, Q. P. Chu, and E.-J. van Kampen, "Design and flight testing of incremental nonlinear dynamic inversion-based control laws for a passenger aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018. AIAA 2018-0385.

[25] H. Xin, Q. Chen, P. Wang, Y. Wang, and Z. Hou, "The control performance analysis of the incremental nonlinear dynamic inverse method and flight test," in *International Conference on Autonomous Unmanned Systems*, pp. 3609–3619, Springer, 2022.

[26] A. Steinert, S. Raab, S. Hafner, F. Holzapfel, and H. Hong, "Advancements in incremental nonlinear dynamic inversion and its components: A survey on INDI – part II," *Chinese Journal of Aeronautics*, vol. 38, no. 11, p. 103591, 2025.

[27] R. R. Da Costa, Q. P. Chu, and J. A. Mulder, "Reentry flight controller design using nonlinear dynamic inversion," *Journal of Spacecraft and Rockets*, vol. 40, no. 1, pp. 64–71, 2003.

[28] P. Acquatella B., L. E. Briese, and K. Schnepper, "Guidance command generation and nonlinear dynamic inversion control for reusable launch vehicles," in *69th International Astronautical Congress (IAC)*, (Bremen, Germany), 2018. IAC–18–C1.5.12.

[29] E. Mooij, "Robust control of a conventional aeroelastic launch vehicle," in *AIAA Scitech 2020 Forum*, (Orlando, FL), American Institute of Aeronautics and Astronautics, 2020. AIAA 2020-1103.

[30] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: Dynamics, controls design, and autonomous systems*. Hoboken, New Jersey: Wiley, 3rd ed., 2015.

[31] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, NJ: Prentice Hall, 3rd ed., 2002.

[32] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[33] S. 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*, American Institute of Aeronautics and Astronautics, 2018. AIAA 2018-3478.

[34] D. Milz, M. S. May, and G. Looye, "Dynamic inversion-based control concept for transformational tilt-wing eVTOLs," in *AIAA SciTech 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024. AIAA 2024-1290.

[35] D. Surmann and S. Myschik, "Gain design for an INDI-based flight control algorithm for a conceptual lift-to-cruise vehicle," 2024. Venue not recorded.

[36] X. Wang, E.-J. van Kampen, Q. P. Chu, and P. Lu, "Stability analysis for incremental nonlinear dynamic inversion control," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 5, pp. 1116–1129, 2019.

[37] P. Acquatella, W. Falkena, E.-J. van Kampen, and Q. P. Chu, "Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion," in *AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, 2012.

[38] Y. Huang, Y. Zhang, D. M. Pool, O. Stroosma, and Q. P. Chu, "Time-delay margin and robustness of incremental nonlinear dynamic inversion control," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 2, pp. 394–404, 2022.

[39] R. J. Adams and S. S. Banda, "Robust flight control design using dynamic inversion and structured singular value synthesis," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 2, pp. 80–92, 1993.

[40] J. Song, X. Song, and Z. Ren, "An integrated sliding model dynamic inversion control method for reusable launch vehicle reentry attitude control," in *2011 International Conference on Electrical and Control Engineering*, pp. 986–991, IEEE, 2011. "sliding model" is the published title; not a transcription error.

[41] J. Reiner, G. J. Balas, and W. L. Garrard, "Flight control design using robust dynamic inversion and time-scale separation," *Automatica*, vol. 32, no. 11, pp. 1493–1504, 1996.

[42] J. Zhang and F. Holzapfel, "Saturation protection with pseudo control hedging: A control allocation perspective," in *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021. AIAA 2021-0370.

[43] R. Steffensen, A. Steinert, Z. Mbikayi, S. Raab, J. Angelov, and F. Holzapfel, "Filter and sensor delay synchronization in incremental flight control laws," *Aerospace Systems*, vol. 6, pp. 285–304, 2023. 2.

[44] G. M. Siouris, *Missile guidance and control systems*. New York: Springer, 2004.

[45] A. Marwege, J. Riehmer, J. Klevanski, A. Gülhan, T. Ecker, B. Reimann, and E. Dumont, "First wind tunnel data of CALLISTO – reusable VTVL launcher first stage demonstrator," in *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, (Madrid, Spain), 2019.

[46] J. Klevanski, "Aerodynamic studies in preparation for CALLISTO – reusable VTVL launcher first stage demonstrator," 2018. Venue not recorded.

[47] J. A. Macés-Hernández, D. Seelbinder, and S. Theil, "Design of flight control systems for RLVs with structural flexibility: Application to the CALLISTO vehicle," in *IAF Astrodynamics Symposium, 75th International Astronautical Congress (IAC 2024)*, (Milan, Italy), pp. 195–205, International Astronautical Federation (IAF), 2024.

[48] P. Lu, E.-J. van Kampen, C. de Visser, and Q. P. Chu, "Aircraft fault-tolerant trajectory control using incremental nonlinear dynamic inversion," *Control Engineering Practice*, vol. 57, pp. 126–141, 2016.

[49] M. Sagliano, T. Tsukamoto, J. A. Macés-Hernández, D. Seelbinder, S. Ishimoto, and E. Dumont, "Guidance and control strategy for the CALLISTO flight experiment," in *8th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, 2019.

[50] K. Ogata, *Modern control engineering*. Upper Saddle River, NJ: Prentice Hall, 5th ed., 2010.

[51] P. Acquatella and Q. P. Chu, "Agile spacecraft attitude control: an incremental nonlinear dynamic inversion approach," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5709–5716, 2020.

# Appendix A
# Mathematical Concepts

## A.1 Lie Derivatives and Lie Brackets

The presented definitions of the mathematical operations *Lie Derivative* and *Lie Bracket* are paraphrased from the referenced works [32][31]. For a detailed description going beyond the scope of this limited version, the reader is referred to the original sources.

### Lie Derivative

The Lie Derivative is an operation that returns the directional derivative of a smooth scalar function $h(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ in the direction of a smooth vector field $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$. It is formally defined as

$$\mathcal{L}_{\boldsymbol{f}} h = (\nabla h)^{\mathrm{T}} \boldsymbol{f} \tag{A.1}$$

When applied in succession, Lie Derivatives can be defined recursively as

$$\begin{aligned} \mathcal{L}^0_{\boldsymbol{f}} h &= h \\ \mathcal{L}^i_{\boldsymbol{f}} h &= \mathcal{L}_{\boldsymbol{f}} (\mathcal{L}^{i-1}_{\boldsymbol{f}} h) \qquad\qquad \forall i \in 1, 2, ... \end{aligned} \tag{A.2}$$

The repeated application is not restricted to only one individual vector field. When the Lie Derivative is formed in the direction of another smooth vector field $\boldsymbol{g}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$, the resulting scalar function is defined as

$$\mathcal{L}_{\boldsymbol{g}} \mathcal{L}_{\boldsymbol{f}} h = (\nabla (\mathcal{L}_{\boldsymbol{f}} h))^{\mathrm{T}} \boldsymbol{g} \tag{A.3}$$

### Lie Bracket

The Lie Bracket operator returns a vector field from two smooth vector fields $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{g}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$ according to the following expression.

$$ad_{\boldsymbol{f}} \boldsymbol{g} = \nabla \boldsymbol{g} \boldsymbol{f} - \nabla \boldsymbol{f} \boldsymbol{g} \tag{A.4}$$

Other notations of the Lie Bracket include $[\boldsymbol{f}, \boldsymbol{g}]$ and $\mathcal{L}_{\boldsymbol{f}} \boldsymbol{g}$, though the formulation given in A.4 shall be used within this work.

As is the case for the Lie Derivative, successive application of the Lie Bracket can be defined recursively as

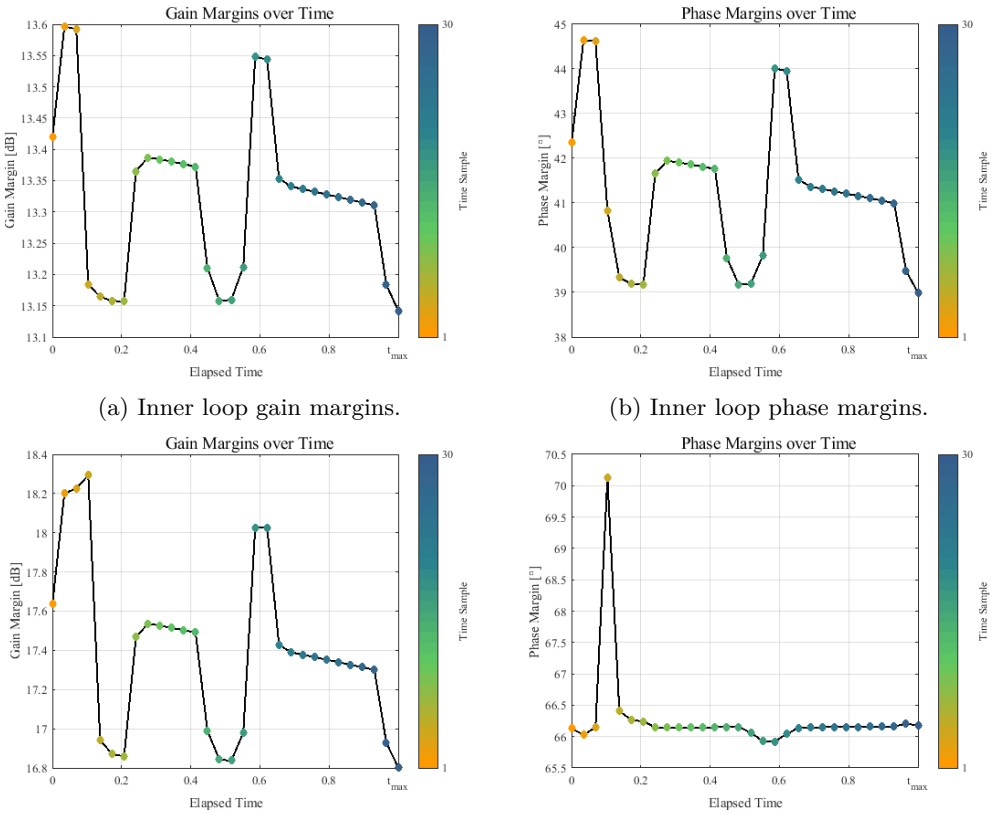
$$\begin{aligned} \mathcal{L}^0_{\boldsymbol{f}} \boldsymbol{g} &= \boldsymbol{g} \\ \mathcal{L}^i_{\boldsymbol{f}} \boldsymbol{g} &= \mathcal{L}_{\boldsymbol{f}} (\mathcal{L}^{i-1}_{\boldsymbol{f}} \boldsymbol{g}) \qquad\qquad \forall i \in 1, 2, ... \end{aligned} \tag{A.5}$$

Further algebraic properties of the Lie Bracket can be found in the sources mentioned at the beginning of this appendix.

# Appendix B
# Additional Results

This appendix presents some additional results which add to the performance analysis of the NDI-based control architecture in the linear verification framework. They are referenced in the main body of this document in the respective context.

## B.1 Margin Evolution of the Cascaded Architecture




(a) Inner loop gain margins.

(b) Inner loop phase margins.

(c) Outer loop gain margins.

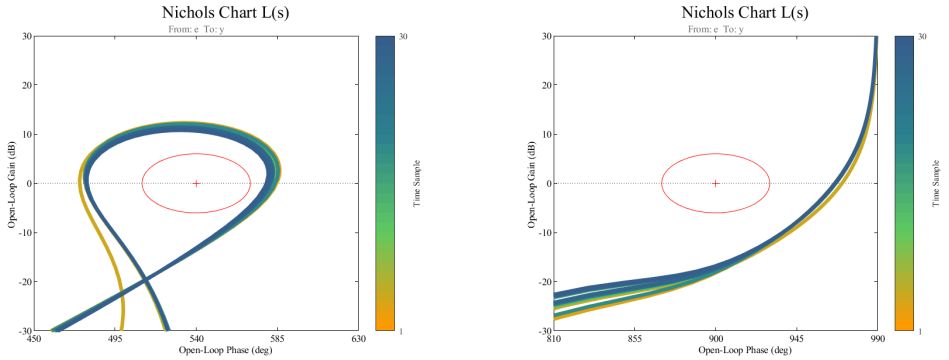
(d) Outer loop phase margins.

Figure B.1: Gain and phase margin evolution for the planar cascaded control architecture on the loop functions of the outer $L_{\text{trans}}(s)$ and inner loops $L_{\text{att}}(s)$.

Figure B.1 shows the gain and phase margin evolution for the outer and inner loop transfer functions $L_{\text{trans}}(s)$ and $L_{\text{att}}(s)$. All except the phase margins of the outer loop can be seen to be dominated by the thrust magnitude, which follows a similar profile. Further, the recurring outlier already identified in chapter 6 reappears in the outer loop phase and gain margins. It can be seen to affect the phase margins of the outer loop most prominently. Previous analysis has shown the outer loop phase margin to follow an inverse dependence on the thrust magnitude.

## B.2 Linearisation Loop Based on the Non-linear Model

When the NDI-based control architecture in the linear verification framework is derived from the non-linear system equations as presented in chapter 4, the results change only marginally. This appendix briefly presents these results and discusses some of the most important observations when compared to the NDI-based implementation derived from the linear reference model in chapter 5. Figure B.2 depicts the Nichols diagrams of the outer and inner loops in the cascaded controller when constructed from the non-linear plant model. As before, thirty samples are extracted from the trajectory to form the analysed LTIs.




(a) Attitude. (b) Translation.

Figure B.2: Nichols diagrams of the open-loop functions $L_{\text{att}}(s)$ and $L_{\text{trans}}(s)$ of the cascaded controller based on the non-linear plant model for thirty discrete time samples along the trajectory.

The results are very similar to the implementation using the linear reference plant. This can be explained when the construction of the LTIs in the linear verification framework is considered again. Both the linearisation loops and the plant are built with static variables from the reference trajectory, depicting the frozen dynamic behaviour at that specific sample in time. As long as this reference trajectory is used to build the LTIs, which itself follows the dynamic behaviour of the vehicle, the results between the two linearisation loop variants will remain very similar.

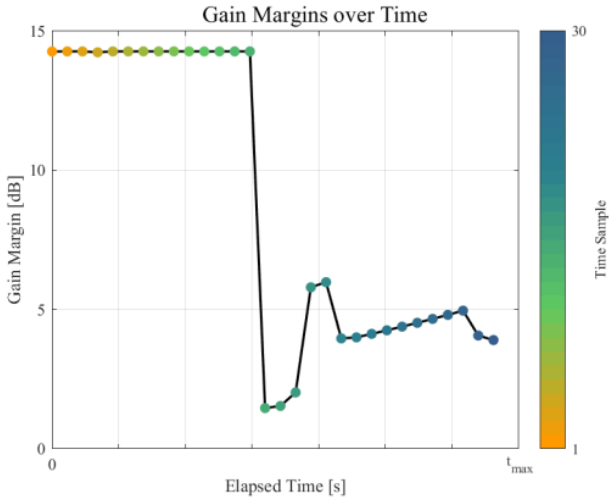



Figure B.3: Gain margin evolution for the isolated pitch plane attitude loop transfer function $L_{\text{att}}(s)$ for thirty discrete time samples along the trajectory. The inversion law is derived from the non-linear system equations.

Where a difference between the two implementations can be seen more prominently, however, is in the isolated analysis of the attitude control loop. Figure B.3 depicts the evolution of the gain margins of the isolated inner loop. It can be seen that the margins drastically diminish from

the descent phase onwards when compared to the gain margin evolution of the linear model-based implementation in figure 6.8a. This is connected to the modelling error contained within the linear reference model and gives an indication that the error is contained within the attitude dynamics. The gain margin evolution shows that the non-linear system equation-based implementation does not contain this error and, as a consequence, the linearisation loop is only accurate in the first half of the flight, where the plant dynamics behave as expected. Beyond that, the double integrator approximation can no longer be formed properly and the margins dwindle. This exposes a bias in the approach of deriving the linearisation law together with the linear plant model. When both contain the same error, the inversion law will make it difficult to observe the error's effect, as merely a double integrator remains after applying the linearisation loop.

## B.3 Second Test Flight Scenario Monte-Carlo Analysis

This section presents the results obtained during a Monte-Carlo campaign carried out on the second test flight scenario of the CALLISTO vehicle described in section 6.3. Fifty-three randomised uncertain cases are contained within the campaign.

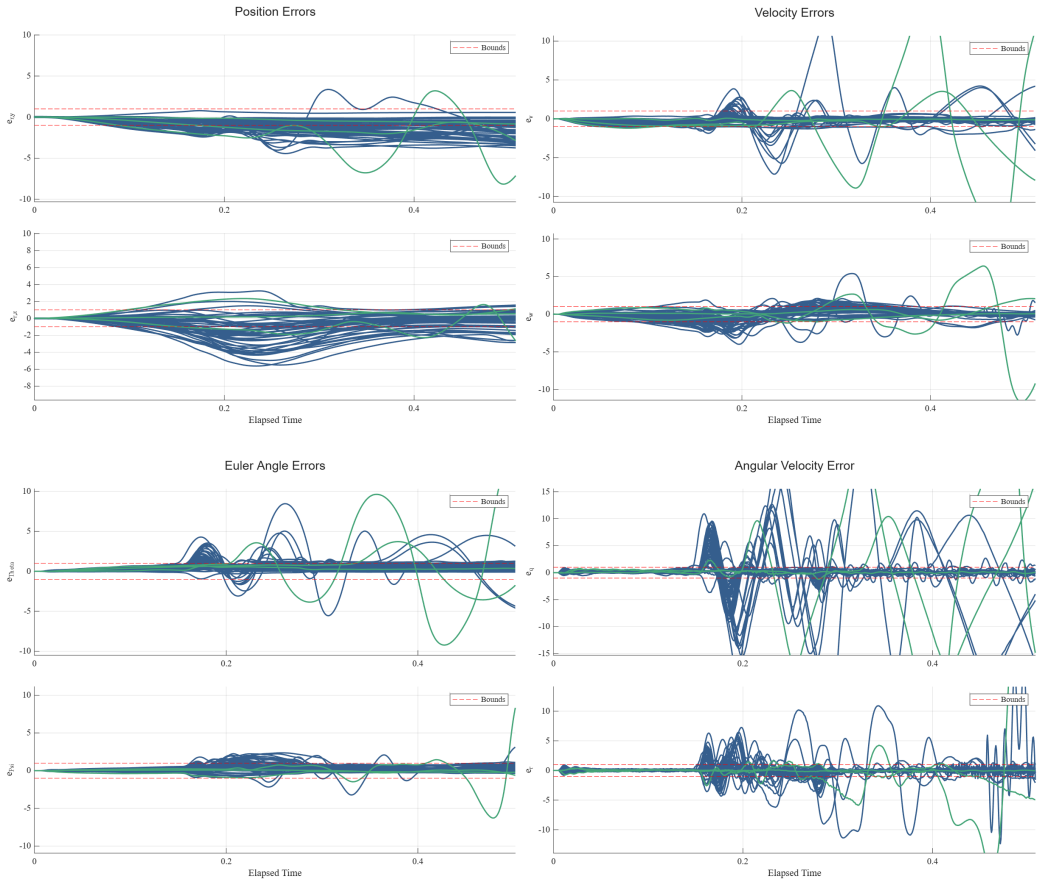



Figure B.4: Normalised errors in position, velocity, angles and angular velocities in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified cases (green) for the second demonstration scenario of CALLISTO. Only ascent and hover flight phases are displayed.

Figure B.4 displays the normalised errors in position, velocity, Euler angles and angular velocities. It should be kept in mind that the NDI-based control architecture only controls the system during the first half of the flight, approximately until 52% of the flight time has passed. It immediately becomes clear that under the more dynamic conditions of the second demonstration scenario, certain sets of uncertainties cause highly oscillatory or even unstable behaviour. These uncertain samples pose a threat to the confidence in the control solution, as clearly a large set of uncertain

variable combinations exists, which would possibly lead to system failure. Beyond the unstable MC runs, similar trends as during the first test flight can be observed. The lateral drift error $y_e$ and the pitch angle errors $\Theta_e$ contain the same wind-based bias and the time of maximum dynamic pressure causes a peak across all errors. It is somewhat expected that the performance under the more dynamic conditions of the second test flight results in this partly unstable behaviour. It is likely that the linearisation quality of the inversion structure deteriorates with the more non-linear behaviour of the plant during these more extreme conditions. Without an accompanying linear analysis to isolate the origin of the instabilities however, it is difficult to properly assess the control solution's robustness in this scenario. A future investigation could aim to explain the behaviour under these more dynamic conditions in further detail.

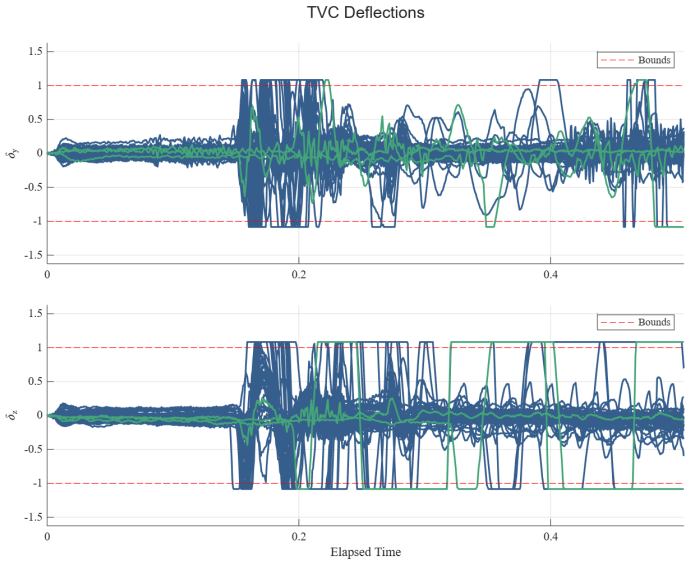



Figure B.5: Normalised TVC deflection angles in both the pitch and yaw planes during the non-linear simulation of 50 random sets of uncertainties (blue) and three pre-identified cases (green) for the second demonstration of CALLISTO. Only ascent and hover flight phases are displayed.

Figure B.5 depicts the normalised TVC control action for the same MC campaign and again reflects the performance deterioration for specific combinations of uncertain sets of variables. From the time of maximum dynamic pressure, many of the MC runs saturate along the trajectory, reflecting the attempted control of the highly oscillatory behaviour seen in the errors. Outside of the high velocity domain during ascent where saturation is most prevalent, the control action seems to remain within similar bounds as for the first test flight scenario with a maximum deflection of about 50%.

In this second scenario, because the performance is seen to deteriorate greatly for a number of sets of uncertain variables even within this small number of samples, the conclusion can be drawn that the dependence of the NDI-based control architecture on the accuracy of the underlying mathematical model poses a real threat to its applicability in uncertain conditions. It underlines the need to investigate extensions of the method like INDI, which reduce the dependence on the mathematical model. They promise to produce more satisfactory performance when model mismatches occur, as is the case in uncertain conditions.

# Appendix C
# Project Management

The project management concept was devised at the beginning of the study and adapted according to changing conditions over the course of the project. It encompasses a Work Breakdown Structure (WBS) and a Gantt chart, of which the WBS is presented in the following.

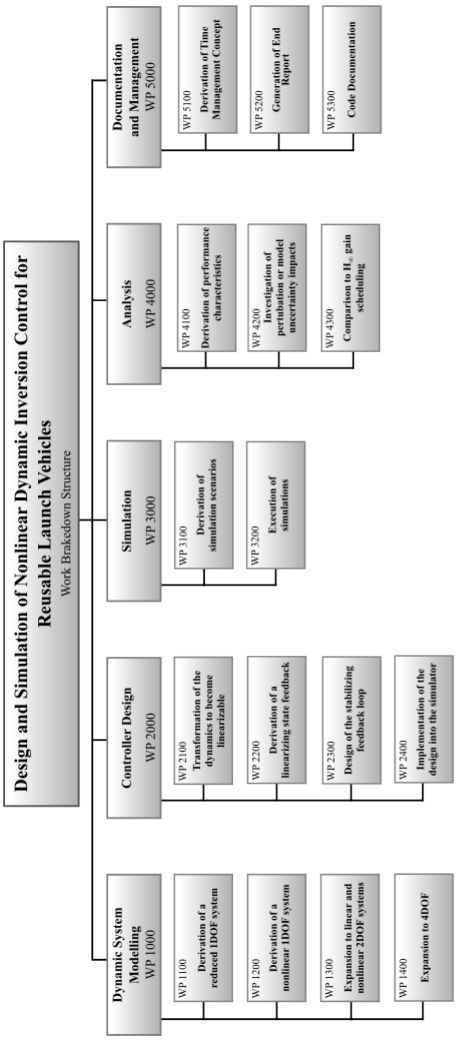



Figure C.1: Project Work Breakdown Structure.

Universität
Bremen

## Hinweise zu den offiziellen Erklärungen

1. Die folgende Seite mit den offiziellen Erklärungen

   **A)** Eigenständigkeitserklärung,

   **B)** Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten und

   **C)** Einverständniserklärung über die Bereitstellung und Nutzung der Bachelorarbeit / Masterarbeit in elektronischer Form zur Überprüfung durch eine Plagiatssoftware

   ist unverändert im Wortlaut in die Bachelor- oder Masterarbeit zu übernehmen.

   **Bitte achten Sie darauf die Erklärungen zu unterschreiben. Ein Scan der Unterschrift ist ausreichend.**

**zu B)**
Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

**zu C)**
Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig.
Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.
Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatssoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

---

## Notes on the official declarations

1. The following pages with the official declarations
   A) Declaration of Authorship
   B) Declaration on the Publication of Bachelor's or Master's Thesis
   C) Declaration of Consent for the Provision and Use of the Bachelor's Thesis / Master's Thesis in Electronic Form for Review by Plagiarism Software

   is to be adopted unchanged in the wording to the bachelor's or master's thesis.

   **Please be sure to sign each declaration. A scan is sufficient.**

**Regarding B)**
The consent can be revoked at any time with future effect by notifying the University of Bremen.

**Regarding C)**
Consent for the permanent storage of the text is voluntary.The consent can be revoked at any time with future effect by notifying the University of Bremen.

Further information on the checking of written work using plagiarism software can be found in the data protection and usage concept. This can be found on the University of Bremen website.

Universität Bremen

# Eigenständigkeits- und Einverständniserklärung zur Überprüfung mit Plagiatssoftware sowie die Erklärung zur Veröffentlichung bei Bachelor- und Masterarbeiten

# Declarations of Authorship and Consent for Checking with Plagiarism Software and the Declaration of Publication for Bachelor's and Master's Thesis

Studierenden-Angaben / Student Information:

| | |
|---|---|
| Matrikelnr./ Student ID | **6347144** |
| Nachname / Surname | **Niebuhr** |
| Vorname / First Name | **Christian** |

Titel der Arbeit / Title of Thesis

**Design and Simulation of Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles**

## A) Eigenständigkeitserklärung / Declaration of Authorship

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge.
Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht.

I hereby affirm that I have written the present work independently and have used no sources or aids other than those indicated. All parts of my work that have been taken from other works, either verbatim or in terms of meaning, have been marked as such, indicating the source. The same applies to drawings, sketches, pictorial representations and sources from the Internet, including AI-based applications or tools. The work has not yet been submitted in the same or a similar form as a final examination paper.

☑ Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI basierte Anwendungen" dokumentiert.

I have used AI-based applications and/or tools and documented them in the appendix "Use of AI-based applications".

**B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten**
Declaration regarding the publication of bachelor's or master's thesis

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten. Archiviert werden:

Two years after graduation, the thesis is offered to the archive of the University of Bremen for permanent archiving. The following are archived:

1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Masterarbeiten
Master's theses with a local or regional focus, as well as per subject and academic year 10 % of all Master's thesis

2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.
Bachelor's thesis for the first and last bachelor's degrees per subject and year.

☑ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I agree that my thesis may be viewed by third parties in the university archive for academic purposes.

☐ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I agree that my thesis may be viewed by third parties for academic purposes in the university archive after 30 years in accordance with para. 2 BremArchivG).

☐ Ich bin **nicht** damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I do not consent to my thesis being made available in the university archive for third parties to view for academic purposes.

## C) Einverständniserklärung zur elektronischen Überprüfung der Arbeit auf Plagiate
## Declaration of consent for electronic checking of the work for plagiarism

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatsvorwürfe untersucht werden.
Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatssoftware.

Submitted papers can be checked for plagiarism using qualified software in accordance with § 18 of the General Section of the Bachelor's or Master's Degree Examination Regulations of the University of Bremen. For the purpose of checking for plagiarism, the upload to the server is done using the plagiarism software currently used by the University of Bremen.

☑ Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

I agree that the work I have submitted and written will be stored permanently on the external server of the plagiarism software currently used by the University of Bremen, in a library belonging to the institution accessed only by the University of Bremen), for the above-mentioned purpose.

☐ Ich bin **nicht** damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

I do not consent to the work I submitted and wrote being permanently stored on the external server of the plagiarism software currently used by the University of Bremen, in a library belonging to the institution accessed only by the University of Bremen), for the above-mentioned purpose.

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig. Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden. Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Consent to the permanent storage of the text is voluntary. Consent can be withdrawn at any time by making a declaration to this effect to the University of Bremen, with effect for the future. Further information on the checking of written work using plagiarism software can be found in the data protection and usage concept. This can be found on the University of Bremen website.

Mit meiner Unterschrift versichere ich, dass ich die obenstehenden Erklärungen gelesen und verstanden habe und bestätige die Richtigkeit der gemachten Angaben.

With my signature, I confirm that I have read and understood the above explanations and confirm the accuracy of the information provided.

17.10.2026

Datum / Date

Unterschrift als Scan/ Signature

Table D.1: Documentation of AI tool usage.

| | **AI-based Tool** | **Purpose** | **Aspect of the Work Affected** | **Example Prompt (if applicable)** | **Comment** |
|---|---|---|---|---|---|
| 1 | Claude Opus 5 | Literature Database Assistance | Literature Management | Please find mentions of (*topic*) in my literature database. Strictly adhere to the database and merely highlight insufficiencies instead of filling them with sources from the internet. Remain brutally honest. | The described prompt is an example. The setup further consists of two agentic skills. One to make sure only the provided literature is scanned and one to ensure brutal honesty in the answers. |
| 2 | Claude Sonnet 5 | Image Prompt Assistance | Title Page | I would like to generate a cover image for my masters thesis and require a prompt for an image generation model. The first image I gave you is approximately what I would like to have in terms of composition. Please adjust the number engines according to the CALLISTO vehicle. It has only one engine and landing legs. The second image depicts the CALLISTO vehicle and its subsystems. Please have the legs be deployed as in the image. | The described prompt came with two context images which are referenced on the title page of this work. |
| 3 | Gemini Nano Banana 2 | Cover Image Generation | Title Page | *No single prompt was used to generate the image. Over the course of several prompts the image was iteratively adjusted to arrive at the final product.* | See prompt column. |
| 4 | Claude Opus 5 | Code Generation, Review and Debugging | Implementation, Analysis | How can I implement [*example feature*] in [*coding language*]. Please ask questions if something is not clear and provide a stand-alone solution preferably in form of a function or object. | In certain cases context in form of example images of plots or code snippets were provided as context and a continuous chat was used to have the model reference previous solutions and adhere to previously defined preferences. |