Robust nonlinear attitude control of aerospace vehicles

An incremental nonlinear control approach

Acquatella Bustillo, P.J.

DOI
10.4233/uuid:99d82992-080c-4c5d-8d40-4e62e62285c0

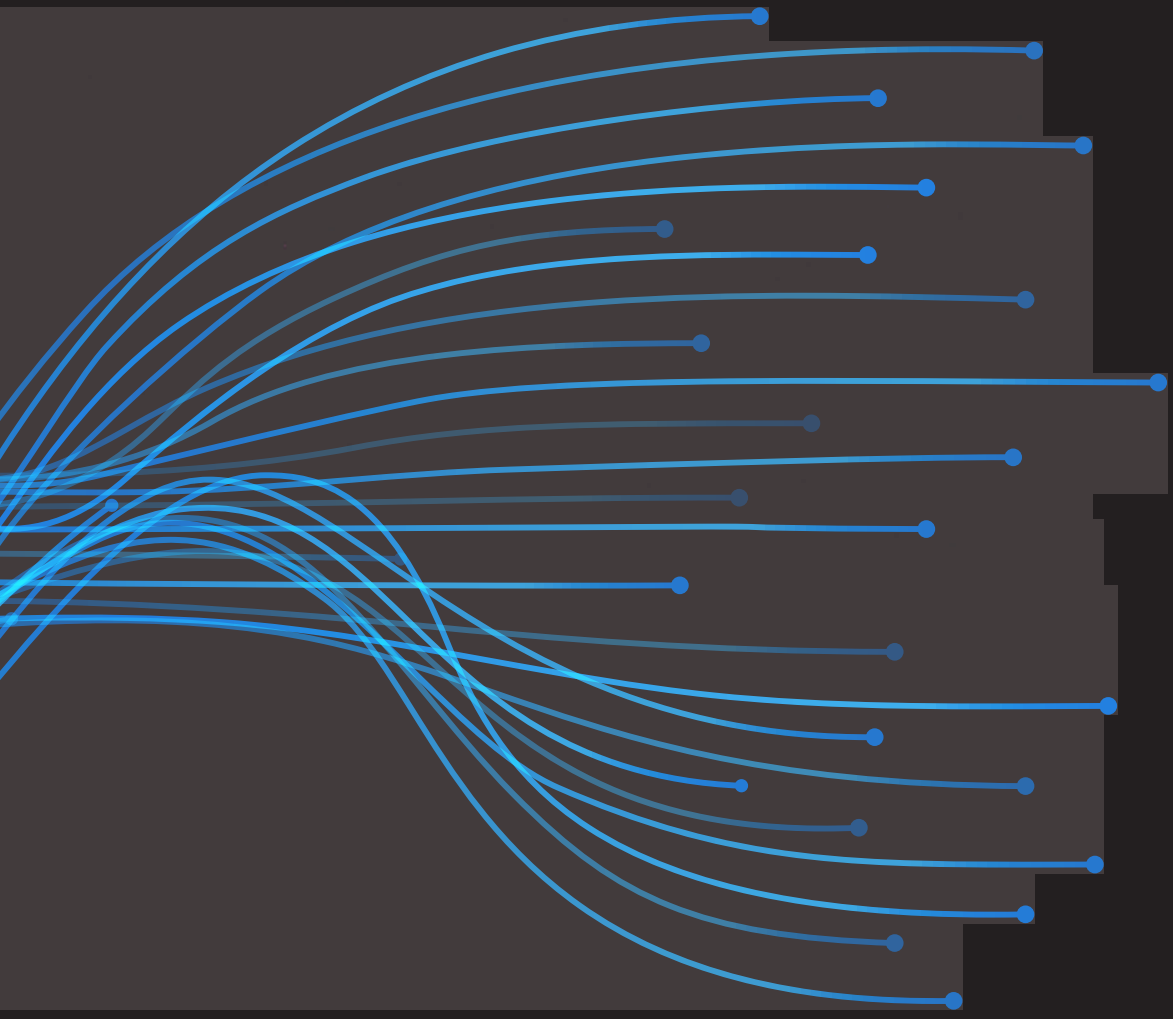
Publication date
2020

Document Version
Final published version

Citation (APA)
Acquatella Bustillo, P. J. (2020). *Robust nonlinear attitude control of aerospace vehicles: An incremental nonlinear control approach*. https://doi.org/10.4233/uuid:99d82992-080c-4c5d-8d40-4e62e62285c0

ROBUST NONLINEAR ATTITUDE CONTROL OF AEROSPACE VEHICLES

an incremental nonlinear control approach



Paul J. Acquatella B.

Robust Nonlinear Attitude Control of Aerospace Vehicles

AN INCREMENTAL NONLINEAR CONTROL APPROACH

ROBUST NONLINEAR ATTITUDE CONTROL OF AEROSPACE VEHICLES

AN INCREMENTAL NONLINEAR CONTROL APPROACH

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus, prof. dr. ir. T. H. J. J van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 30 november 2020 om 15:00 uur

door

Paul José ACQUATELLA BUSTILLO

Ingenieur Lucht- en Ruimtevaart,
Technische Universiteit Delft, Nederland,
geboren te Caracas, Venezuela.

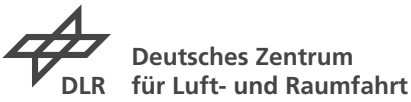
Dit proefschrift is goedgekeurd door de

Promotor: Prof. dr. ir. M. Mulder
Copromotor: Dr. ir. E. van Kampen

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Mulder	Technische Universiteit Delft, promotor
Dr. ir. E. van Kampen	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. E. K. A. Gill	Technische Universiteit Delft
Prof. dr. B. H. K. De Schutter	Technische Universiteit Delft
Prof. dr. A. Tsourdos	Cranfield University, Verenigd Koninkrijk
Prof. dr-Ing. F. Holzapfel	Technische Universität München, Duitsland
Dr. ir. G. H. N. Looye	Deutsches Zentrum für Luft- und Raumfahrt
Prof. dr. G. C.H.E. de Croon	Technische Universiteit Delft, reservelid

Dr. Q. P. Chu heeft in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.



Printed by: Ipskamp Drukkers, Enschede, The Netherlands

Front & Back: Abstract dynamical system response. Graphics: DLR (CC-BY 3.0).

ISBN 978-94-6421-120-7

An electronic version of this dissertation is available at
http://repository.tudelft.nl/

To my beloved family and dearest friends

CONTENTS

List of Figures

SUMMARY

Dynamics modeling, simulation, and control have been studied extensively for many applications in robotics, aeronautics, underwater vehicles, and aerospace vehicles (spacecraft, launchers, re-entry vehicles). In that context, this thesis is motivated from two research directions; namely, space launchers guidance and control (G&C) for preliminary design studies and spacecraft nonlinear and agile attitude control systems.

The research performed in this thesis focuses on two aspects: 1) *attitude motion and control*, which is considered to be one of the classical problems in nonlinear and multivariable control systems; 2) *incremental nonlinear control*, which is a combined model– and sensor–based control approach and has shown promising results in the aerospace community. The high–performance and robustness of incremental nonlinear control comes from the partial dependency removal of an accurate plant model by just requiring a control effectiveness model to estimate the so–called *incremental* dynamics, while relying on angular acceleration and actuator output measurements. This approach, integrated with *nonlinear control* methods, are robust to modeling and parametric uncertainties and allows for aggressive motion control.

The objective of this thesis is to develop concepts and methods for nonlinear flight and attitude control design aspects within a multi-disciplinary modeling and simulation approach. With this approach, attitude dynamics and control can play a more important role in the outcomes of aerospace vehicle design and therefore should be considered more within the preliminary design studies of these vehicles. The research performed in this thesis can be summarized in the following three main parts.

The first part of the thesis is concerned with aerospace vehicle dynamics modeling efforts. The main objective of this part is to investigate *how* an integrated, acausal, and multidisciplinary approach for modeling and simulation *can support* preliminary design studies of space launch vehicles. Such an alternative acausal and multidisciplinary modeling approach, as compared with the methodologies already found in the literature, is implemented with the object–oriented and equation–based modeling language MODELICA which allows to develop subsystems and component models in a *declarative* fashion.

To demonstrate benefits of this approach, a multibody dynamics model was implemented in the context of stage separation dynamics analysis, a critical capability for launch vehicle design studies. Such development of stage separation dynamics in declarative fashion allows performing end–to–end launch vehicle trajectory simulations, by profiting from the mentioned object–oriented and equation–based acausal modeling properties of MODELICA. It is shown that these acausal and declarative modeling features allow for an easy implementation of the *Constraint Force Equation* (CFE) methodology, where the internal joint loads of a multi–stage space launcher can be obtained *automatically* while complying with a set of multi–body constraints: for composite or '*joint*' flight dynamics or during stage separation, respectively.

As example applications, the work developed in this part contributed to studies regarding stage and fairing separation dynamics modeling for generic launchers, and helped to analyse the separation processes and determine possible collision scenarios with the elements involved. The potential of the framework not only spans preliminary design phases, but also relates to activities concerning more detailed system design, software and component verification and validation, and other use cases across a launcher development. However, these efforts outlined had some limitations in their scope and capabilities; for instance, they had not considered in detail the aerodynamics and environment modeling and they were not easily integrable with optimization tools. Those aspects were left to be treated in a separate research aside from the one in this thesis.

The second part of the thesis then focuses on preliminary G&C aspects considering nonlinear design techniques such as *nonlinear dynamic inversion* (NDI) and constrained nonlinear optimization. These techniques are shown to be beneficial for controllability assessments and also for the design of fast slew maneuvers of small satellites. The main objective was to investigate how model–based nonlinear control design and multi–objective optimization could be combined and considered usefully for the study at early design stages of G&C activities in aerospace vehicle dynamics.

A preliminary G&C architecture is presented which combines optimal guidance commands obtained with trajectory optimization together with inner–loop NDI attitude control. This showed that NDI, in combination with trajectory optimization, can be considered for controllability assessments and as a design driver during preliminary launch vehicle design studies. To demonstrate the integrated approach, this G&C architecture was tested on the DLR AURORA reusable launch vehicle concept, where nonlinear flight simulations for the descent phase (including the re–entry) were considered. The results demonstrate the controllability of the launch vehicle as well as the potential to reduce *more than half* the impact on the angular impulse budget for the reaction control system (RCS) by combining it with aerodynamic surface controls during the re–entry phase. This could in turn translate to less propellant mass needed for the RCS, and therefore, better performance of the launcher.

Regarding the extensive topic of optimal reorientation in spacecraft attitude guidance and control, the focus was when discrete–time sampled inputs are required for slewing the continuous–time spacecraft dynamics in agile fashion. This problem was motivated in order to design a high–agility attitude control system for the DLR small satellite BIROS which is actuated in sampled–time by a redundant array of *'High-Torque-Wheels'*. Fast slew maneuvers can be designed for this spacecraft by formulating the problem as a constrained nonlinear optimal control problem. Numerical solutions to this nonlinear optimal control problem can be readily obtained by solving multi–criteria optimization problems using a direct approach and trajectory optimization.

From this second part, it can be concluded that multi–objective optimization techniques, combined with model–based nonlinear control, facilitates early and preliminary G&C studies very efficiently. The limitations regarding the absence of model and parametric uncertainties for improving robustness in the nonlinear attitude control design, and the development of an agile attitude control system that is real–time capable and implementable on board, motivated the next research question.

The third and final part of this thesis brings some contributions to the incremental

nonlinear control body of work. These contributions were motivated on *how to integrate* the incremental nonlinear control approach with backstepping, time-delay control (TDC), and nonlinear PID–control. This part also considers three applications of incremental nonlinear dynamic inversion to robust attitude control of spacecraft, which has not been widely studied in the literature. Incremental nonlinear control requires information of the actuator states and the vehicle's rotational acceleration in order to reduce feedback sensitivities to an inaccurate baseline or airframe model. With such an approach, feedback control dependency on the modeled vehicle dynamics is greatly reduced, overcoming one of the major robustness flaws of conventional model–based flight control systems.

Incremental backstepping was first motivated by combining the design of increments of control action with the recursive step-by-step procedure of the backstepping control design methodology. In this thesis, incremental backstepping is further considered as a methodology for robust nonlinear flight control by tracking outer–loop control variables of such multi–loop nonlinear system *incrementally*, and by accounting for model and parametric uncertainties that may rise during such aggressive maneuvers. This promising methodology for robust nonlinear flight control systems, and its potential, were demonstrated with a longitudinal nonlinear flight control example where good tracking performance was obtained while being subjected to relatively large variations in the vehicle's aerodynamic model parameters.

We present an equivalence of *incremental nonlinear dynamic inversion* (INDI) and *time–delay control* (TDC) when a reformulation of the plant control effectiveness is considered. TDC, more commonly known in the robotics community, is a nonlinear control technique that estimates and compensates for effects of disturbances and system uncertainties by utilizing time–delayed signals of some of the system variables. Moreover, a known relationship and equivalence between discrete formulations of TDC and *proportional-integral-derivative* (PID) for nonlinear plants of second–order controller canonical form, and in the context of a robot motion control application, allowed to find an equivalence between INDI and TDC by considering *sufficiently small* time–delayed signals explicitly, together with a reformulation of the plant control effectiveness and fixed–value gains in the (nonlinear) PID control structure. This brings a new interpretation of INDI that leads to a meaningful and systematic method for tuning of nonlinear PID flight control systems via INDI, as it was previously done for robotics. It is also found that, since incremental nonlinear PIDs are PIDs with state-dependent gains that are implemented in a discrete or sampled–time form, their state-dependent gains might not necessarily be gain–scheduled but rather model–based.

The INDI control approach is shown to be promising for spacecraft attitude control, in particular for agile reorientation maneuvers, since it is robust against model and parametric uncertainty as well as capable to reject external disturbances very effectively. One of the applications considered the attitude tracking and disturbance rejection problem of rigid spacecraft subjected to model and parametric uncertainties, and was initially achieved with a cascaded two–loop control system using as outer-loop control the kinematic inversion of the attitude parameters known as *Modified Rodrigues Parameters* (MRPs). Assuming a time scale separation of the attitude and rate dynamics, the rate control for the inner loop was done using INDI of the plant dynamics. As an improve-

ment versus model–based nonlinear dynamic inversion control, the INDI approach enhances robustness capabilities by reducing feedback control dependency on accurate knowledge of the system dynamics. Simulation results demonstrate the efficient tracking and external disturbances rejection capabilities of the proposed controller under the combined effect of external disturbances, time-delay, and parametric uncertainty.

A recent reformulation of INDI is considered to design a nonlinear and agile spacecraft attitude control system. The improvement over the INDI controller of the previous one is made by designing a full three–axis attitude control for a spacecraft actuated by three reaction wheels, but without the cascaded inner–loop that was based on the assumption of time–scale separation. It is shown that scheduling of the control effectiveness can be done with the Jacobian of the MRP kinematics and is only subject to parametric uncertainty of the spacecraft augmented inertia and its wheelset alignment matrix. A relationship between INDI and nonlinear PID control was found, which demonstrates that (for the class of input–affine nonlinear systems considered) INDI control can be recasted as incremental nonlinear PID control, and vice–versa. The relationship can be useful for closed–loop gain tuning, and for stability and robustness analysis as shown in the literature. Simulation experiments for this particular problem demonstrate that INDI has similar nominal performance as TDC/PID control, but superior robust performance and stability. INDI control for agile spacecraft was again reformulated but in the context of nonlinear sampled–data systems; this was motivated from an explicit consideration of the sampling time via an approximate sampled–data model in normal form that is widely known in the literature. The objective of this reformulation was to bridge the gap between continuous–time and highly sampled INDI formulations (100 – 1000 Hz) and their discrete and lowly sampled counterparts (1 – 10 Hz) in the context of spacecraft attitude control where low sampling rates are common.

Finally, to summarize, in this final part of the thesis we showed that incremental nonlinear control can be integrated with backstepping, time–delay control, and nonlinear PID control; incremental nonlinear control laws can be regarded as both model– and sensor–based, where 'model' refers to the scheduling of the instantaneous control effectiveness; and several applications and scenarios of robust nonlinear attitude control which aim to close the gap in terms of agility, robustness, and performance of future attitude control systems are considered.

The research performed in this thesis is recommended to be continued in three main directions: *i)* consideration of control input constraints and actuator limits; these aspects are very important in particular for agile attitude control systems where exploiting the full capacity of the actuators might be necessary. This raises the question– *what happens during saturation of incremental (nonlinear) control systems?*, *ii)* studying in detail the limits of stability vs. performance, since incremental (nonlinear) control action at high-sampling rates implies or induces a *high–gain* control loop which can compromise stability at high frequencies, and *iii)* doing more efforts in finding relationships between incremental nonlinear control, the early works on TDC pioneered by Hsia and Youcef–Toumi, and the large mathematical body of literature behind nonlinear sampled–data systems established by Monaco and Normand–Cyrot. Most of the literature published on these subjects is found in the fields of robotics, nonlinear control, and applied mathematics; but not so much in the field of aerospace vehicles dynamics and control.

1

INTRODUCTION

This chapter begins with a brief background and motivation and then introduces the approach and the main research questions treated in this thesis. The motivation is driven from preliminary design studies of space launchers and preliminary studies in guidance and control (G&C) for novel aerospace vehicles (space launchers, spacecraft, re-entry vehicles). Further focus is towards dynamics modeling, simulation, and control systems, in particular studying incremental nonlinear control methods for robust and agile attitude control systems design.

1.1. BACKGROUND AND MOTIVATION

THE background and motivation of this thesis comes from two research directions in the context of dynamics modeling, simulation, and control of aerospace vehicles; namely, space launchers guidance and control (G&C) for preliminary design studies and spacecraft nonlinear and agile attitude control systems.

1.1.1. SPACE LAUNCHERS G&C FOR PRELIMINARY DESIGN STUDIES

Reusability of launch vehicles strongly affects the launch servicing market whenever sufficient reliability and low refurbishment costs can be achieved [7]. Keeping up with the rapidly–evolving international launch market is essential for Europe, and with that the need to explore various methods and technologies for reusability [7–10].

Several studies on future launch vehicle configurations and technologies for expendable and reusable launch vehicles have been conducted in the past at the German Aerospace Center (DLR) [8, 10–16]. Currently, partly or fully reusable launch vehicles using different return methods are investigated at DLR within research projects AKIRA, X-TRAS [10, 16–18], and ReFEx (Reusability Flight Experiment) [19–23]. An example is the winged Liquid Fly-back Booster concept LFBB, studied extensively during the early 2000's [24] and more recently based on an LOX/LH2 propellant combination for vertical take-off and vertical landing (VTVL) [17]. A more recent study concerns the delta-winged

horizontal takeoff and horizontal landing (HTHL) concept AURORA [18], based on an LOX/Kerosene propellant combination.

Figure 1.1: DLR SART – SpaceLiner 7. Left: SpaceLiner concept at stage separation with passenger stage in upper position. Right: Artist's impression of satellite payload release from SpaceLiner 7 Orbiter's open payload bay in low-Earth orbit (LEO). Credits: DLR, CC-BY 3.0. [1]

In this context, the field of work and research in space launchers focuses on evaluations of conventional and reusable launch vehicle concepts, in a multidisciplinary collaboration with other DLR institutes. In particular, focus lies on the system dynamics, guidance, and control level (G&C) where contributions in simulation, trajectory optimization, and control design of such space systems have been made.

For instance, after computation of optimal flight trajectories, detailed studies and assessments of the launcher performance during all relevant flight phases can be performed. These can include controllability and stability studies, stage separation studies, and preliminary design of the control systems of the space launcher. Furthermore, these aspects can have effects on the general design of the considered space launcher concept.

The need *for improvement of early systems analysis capability*, which comes with the increased complexity and cost of space launch vehicles [8, 16], is the first motivation of this thesis; that is why the analysis and methods proposed here are aimed only at conceptual and preliminary design phases. These early efforts are important since the vehicle's design and technology decisions have a major impact on its final configuration, feasibility, and on several costs across its development and operations.

In that sense, for the launcher concepts and configurations to consider and optimize at the stages of preliminary design, this thesis focuses on *stability and controllability estimates of these designs as early as possible*.

1.1.2. SPACECRAFT G&C FOR AGILITY AND ROBUSTNESS

Developing technological foundations for new space missions, and especially in the field of *agile Earth Observation* [25–33] for crisis warning and management systems, future small satellite systems have to be more performant: this implies not only fine–pointing capabilities for data acquisition, but also high agility for maneuverability, e. g., high dynamic slewing capability to command the platform for fast and flexible data acquisition.

Advancing the *capability for agility as well as for robustness* in terms of spacecraft attitude control systems is the second motivating aspect of this thesis. Again, we focus on proposing methods at the conceptual and preliminary design phases, which will naturally call for further development when it comes to validation and verification. In that

[1] Source: https://www.dlr.de/irs/en/

sense, this thesis will consider technological (software) experiments in terms of *spacecraft fast-slewing capability as well as robust nonlinear attitude control*, which could demonstrate the feasibility of new approaches for the next generation of optical remote sensing space missions.

One of the applications of this method is oriented towards the recently launched German satellite platform BIROS (*Bispectral InfraRed Optical System*) [27], which is the second technology demonstrator along with the TET-1 (*Technologie Erprobungs Träger* or *Technology Experiment Carrier*) satellite of the DLR 'FireBIRD' (*Fire Bispectral InfraRed Detector*) [28] constellation aiming to provide infrared (IR) remote sensing for early fire detection. Monitoring fires from space involves the detection and measurement of so-called high-temperature events in forests, volcanic activity, gas flares and industrial hotspots. These small satellites are largely based on the flight-proven *Bi-spectral Infra-Red Detection* (BIRD) [34, 35] satellite bus launched in 2001. With the provision of such remote sensing data from FireBIRD, DLR not only supports crisis management activities in the frame of the International Charter "Space and Major Disasters" [2], but also contributes towards scientific goals, being an R&D project of DLR's Aerospace Research and Technology program division.

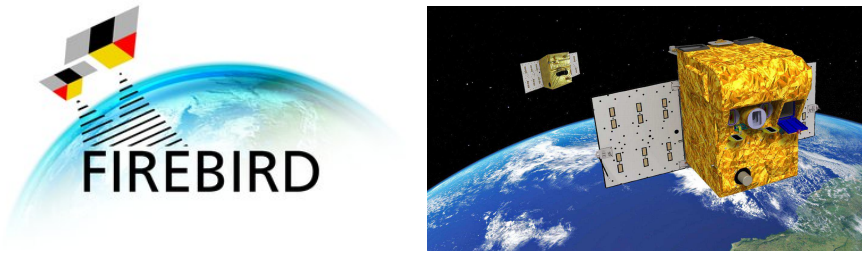


Figure 1.2: FireBIRD – a satellite duo for fire detection. Left: Mission logo. Right: artist impression of BIROS (front) and TET-1 (back). Credits: DLR, CC-BY 3.0. [3]

BIROS was launched on 22 June 2016 at 05:55 CEST into a Sun-synchronous orbit, while TET-1 has been orbiting Earth in a polar orbit since July 2012 and has successfully concluded the first part of its mission as a technology demonstrator. The BIROS satellite bus segment is based on the one developed for TET-1, but is additionally equipped with a propulsion system for active attitude and orbit control. Both satellites are equipped with a multi–spectral camera system as the main payload.

Among several mission goals and scientific experiments, the BIROS platform is also equipped with a redundant set of three *'High-Torque-Wheels' (HTW)* [25, 26] in orthogonal configuration in order to demonstrate a high-agility attitude control system.

One of the main requirements for the HTW experiment is being able to rotate the satellite 30 degrees in 10 seconds around an axis with inertia of about 10 Kg.m^2. For 3-axis agile maneuvers, however, the attitude control system design is more challenging given the current on-board-computer (OBC) requirement of commanding the plant at a sampling rate of only 2 Hz and because of the many nonlinearities involved.

[2] Source: https://activations.zki.dlr.de/en/activations/items/ACT139.html
[3] Source: https://www.dlr.de/dlr/en/

1.2. APPROACH

Addressing these motivating problems requires pursuing the following activities:

1.2.1. SYSTEMS MODELING AND SIMULATION

The emphasis of this activity is the creation of dynamic models and simulations with different levels of detail and fidelity for the task at hand; often in cooperation with other research partners and disciplines. Broadly speaking, modeling of a dynamical system or a *dynamical model* often refers to a simplified reflection of a reality described by its time evolution. In this thesis, a model is generally understood to mean the description of a physical system defined by its fundamental mathematical equations or modeled from *first principles* (physics, mechanics, etc.), together with the changes arising from the interactions with its environment or with its subsystems. Simulations of a model refer to the numerical solution of such mathematical equations, depending on the given initial conditions and boundary conditions; these are usually solved with numerical methods or integration routines.

Physical models can be categorized according to their level of detail in the following categories as delineated in the "Core Process: Systems Modeling" since 2012 at the DLR Institute of System Dynamics and Control:

Level 1 – *Architectural*

Stationary models in which the transient processes are neglected. These are often described by algebraic equations and used for high-level system design, considering power balances for instance.

Level 2 – *Functional*

Models in which transient processes are approximated with some physical quantities and described by *ordinary differential equations* (ODEs) and/or *differential algebraic equations* (DAEs). Usage examples include stability studies or for control systems design.

Level 3 – *Behavioral*

Models in which transient processes are described in detail and are usually described by *hybrid differential algebraic equations* (HDAEs). Example applications include electric network quality investigations and modeling of 'more electric' aircraft.

Level 4 – *Distributed*

Models which are spatially distributed and their transient processes are described in detail by *partial differential equations* (PDEs) with FEM (Finite Element Method), CFD (Computational Fluid Dynamics), or DEM (Discrete Element Method). Example applications include detailed vibration investigations.

In this thesis, only system and control models from Level 1 and Level 2 are considered, whereas more complex behavioral system models (Level 3), three-dimensional mechanical models (Level 3), and FEM Models (Level 4) are not. The multi-disciplinary models and control systems are modeled either with MATLAB&SIMULINK[36], with MODELICA [37–41], or with a combination of both, depending on the task or problem at hand.

MATLAB&SIMULINK is known for its 'block–based' modeling capabilities, meaning that models are usually implemented within a hierarchical structure and according to a signal flow approach where a predefined input/output (I/O) causality is fixed and their relationship is explicit. This is commonly known as a *causal* or *imperative* modeling approach. In contrast to such classical block–based approach for modeling, MODELICA offers the possibility to think of models from a 'component–based' approach where these are rather implemented without a predefined I/O causality, and interact with their environment by means of physical ports, called connectors, that represent some kind of information or energy exchange. For this reason, this paradigm is commonly known as an *acausal* or *declarative* modeling approach [41]. Declarative models are described by their fundamental mathematical equations or first principles. This also means that declarations are given without an explicit order or *how* to compute them, and that is why these models are often called *physics*– or *equation*–based [41].

Models in MODELICA are described using differential, algebraic, and discrete equations which are then mapped into hybrid *differential algebraic equations* (DAEs). DAE systems are generally expressed on their implicit form [40, 41] as

$$\mathbf{F}\big(\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{y}(t), \boldsymbol{\rho}, t\big) = \mathbf{0}, \tag{1.1}$$

where $\dot{\boldsymbol{x}}$ are the state derivatives, $\boldsymbol{x}$ the state variables, $\boldsymbol{u}$ the inputs, $\boldsymbol{y}$ the algebraic variables, $\boldsymbol{\rho}$ the parameters and constants, and t the time variable. Systems are then solved and simulated by a MODELICA simulation environment such as DYMOLA. Moreover, when these systems are represented in this DAE implicit form, they can be solved directly by a DAE solver such as DASSL [42, 43]. Alternatively, by means of specialized algorithms, the system can be sorted according to specific inputs and outputs and mapped into an explicit *ordinary differential equation* (ODE) form by solving for the derivatives and the algebraic variables, and then subsequently solved numerically by typical ODE solvers. The translation process of such declarative models into efficient computer executable code and details of this compilation is, however, beyond the scope of this thesis [44–47].

Another advantage of using MODELICA is that it is *domain neutral*; it has a multidomain modeling capability which means that model components corresponding to physical objects from several different domains can be described and connected efficiently. An example of this domain neutral feature is shown in [48] where an inverse modeling approach for 'more electric' aircraft equipment systems is considered, allowing to analyse power behaviour as a result of given load profiles for electrical, mechanical, hydraulic, and pneumatic equipment systems in a single model. For all those reasons, this thesis considers such modeling approaches in the context of dynamics modeling, simulation, and control of aerospace vehicles.

1.2.2. GUIDANCE AND CONTROL (G&C)

To address the motivation concerning *early stability and controllability aspects*, the emphasis of this process lies on model–based nonlinear methods for *guidance and control*.

Guidance broadly speaking refers to the generation of maneuver commands or a trajectory to achieve a particular vehicle motion. This can be achieved offline or online, depending on the nature of the problem, and uses state information (estimation) from a navigation system. Planning a motion or obtaining a trajectory is usually done in an optimal sense and to achieve a particular set of goals. The optimal trajectory or path reference and their corresponding commands are commonly used in aerospace G&C as references for an inner-loop attitude control subsystem [49–51].

In that sense, in terms of guidance, this thesis focuses on an off-line approach where guidance commands or reference trajectories are generated or obtained by the following sequential methodology or procedure:

1. Developing comprehensive analytical and/or multi-disciplinary models either with MATLAB&SIMULINK, with MODELICA, or with a combination of both, depending on the task or problem at hand;

2. Formulating of the optimization problem (trajectory optimization, guidance commands, time-optimal control, etc.) for the given set of goals;

3. Transcribing the optimization problem formulation into a constrained and *multi-criteria* or *multi-objective* optimization problem considering inequality and equality constraints;

4. Solving the above with a direct approach using a trajectory optimization package, in this case the package '*trajOpt*' [2] of DLR's optimization tool *MOPS* (*Multi-Objective Parameter Synthesis*) [52–54], implemented in MATLAB [36], which solves multi-objective design problems that are mapped to weighted *min-max* optimization problems.

MOPS is an optimization–based tool that allows thorough assessment of control law designs which has been useful in many aeronautics applications [52–59]. This can be done at several layers, for example by finding optimal parameter tuning, performing Monte–Carlo simulations, but also by evaluating robustness via worst–case search.

The *MOPS* synthesis formula [53, 54, 56] starts by the definition of k design objectives to be minimized together with their desired or *demanded* values, denoted mathematically as c_k and d_k, respectively. The original constrained minimization problem is then transcribed into a direct approach resulting in the following *min-max* multi-criteria optimization problem:

$$\min_{\mathscr{T}} \left\{ \max_{k \in \mathscr{S}_m} \left\{ \frac{c_k(\mathscr{T})}{d_k} \right\} \right\}, \tag{1.2a}$$

$$\begin{aligned} \text{subject to } c_k(\mathscr{T}) &= d_k, \quad k \in \mathscr{S}_{\text{eq}}, \\ c_k(\mathscr{T}) &\leq d_k, \quad k \in \mathscr{S}_{\text{ineq}}, \end{aligned}$$

$$\text{with}$$

$$\mathscr{T}_{\min,l} \leq \mathscr{T}_l \leq \mathscr{T}_{\max,l} \text{ over } [0, t_f] \tag{1.2b}$$

Here [53], $\mathscr{S}_m$ is the index set of criteria to be minimised, $\mathscr{S}_{\text{eq}}$ is the index set of equality constraints and $\mathscr{S}_{\text{ineq}}$ is the index set of inequality constraints. Moreover, $\mathscr{T}$ is the tuning parameters vector $\mathscr{T}_l$ to be optimized and that are bounded by $\mathscr{T}_{\min,l}$ and $\mathscr{T}_{\max,l}$, respectively; $c_k\,(k \in \mathscr{S}_m)$ is the $k-$th normalized criterion and $d_k\,(k \in \mathscr{S}_m)$ its corresponding demand value which serves as a criterion weight; lastly, $c_k\,(k \in \mathscr{S}_{\text{eq}}, \mathscr{S}_{\text{ineq}})$ are normalised criteria which are used as equality or inequality constraints, respectively, and $d_k\,(k \in \mathscr{S}_{\text{eq}}, \mathscr{S}_{\text{ineq}})$ their corresponding demand values.

Finally, the multi-criteria optimization problem in Eq. (1.2) containing the objective function together with equality and inequality constraints can be solved using standard *nonlinear programming* (NLP) methods.

With model–based design optimization, various G&C aspects of the overall system dynamics can be considered at early design stages. The optimization then delivers the best possible compromise between existing conflicting goals and provides a reference for subsequent inner–loop attitude control. This combination is inspired by the one already introduced for aeronautical applications in [60], where combining multi–objective optimization, physical models, and nonlinear control showed enormous potential compared to a more integrated model–based approach. Once optimal guidance commands and reference trajectories are obtained, the next step is tracking or approaching the trajectory via nonlinear control.

Nonlinear control does not only address nonlinear plants *directly*, but is in fact designed to cope with the shortcomings of conventional linearization. For example, linearization does not provide an account about the nonlinearities that might occur during operation since a *local* approximation of the nonlinear system over a small domain of interest around an equilibrium is performed.

Many nonlinear control methodologies have been developed in the past decades to overcome these shortcomings on top of the disadvantages from having model or parametric uncertainties that can compromise the closed-loop stability and convergence of the system. Among the most popular ones are *feedback linearization* (FBL) [61–63], *adaptive control* [64], and *backstepping* [65, 66].

Nonlinear dynamic inversion (NDI), which is how feedback linearization is more commonly known in the aerospace literature [60, 67–70], requires an accurate model of the system to cancel its nonlinearities (entirely or partly) by means of state feedback and exact transformations; although meant for a wide class of systems, NDI may only be applied in combination with physical insight [60]. This brings the nonlinear system dynamics into a decoupled linear form for a particular region of interest around an equilibrium. Once *feedback–linearized*, some conventional linear control methods can be sought to the obtained system in order to achieve desired closed–loop dynamics. This is advantageous since it helps to reduce the complex task of linearizing and synthesizing different (linear, robust) controllers for the several operating points obtained as it is done with gain–scheduling. This drawback of gain–scheduling on top of the difficulties of guaranteeing stability and performance in–between operating points was a motivating aspect for the original applications of NDI for flight control systems. NDI have been widely improved and investigated for flight control applications over the years, specially for improvements regarding robustness and performance [69–75], but has also been considered in space applications such as spacecraft control and re-entry vehicles [49–51].

Backstepping (BKS) [65, 66, 76] is aimed to design stabilizing controllers for nonlinear systems thanks to a systematic Lyapunov–based procedure. The origins of the backstepping method are traced to the survey paper by Kokotović and Arcak [66]. In this thesis we focus on strict-feedback and cascaded nonlinear systems but backstepping can be also applied to a broader class of systems. The method starts by considering the scalar equation that is further separated from the control input by the largest number of integrations and 'steps back' recursively towards the input in order to find a stabilizing controller, hence the name 'backstepping' [6]. For each step there is an intermediate stabilizing function that is based on a recursive virtual control law that must be calculated until the end where the final control law is found. An important feature of backstepping is the flexibility of the method. For instance, and in contrast with NDI, if a nonlinearity is helpful for stabilizing (or in some other sense), it can be kept in the formulation of the final control law and in the closed-loop system even when the size of such stabilizing nonlinearity is not known a priori, which makes the controller less reliable on a precise model. This means that dealing with nonlinearities actually becomes a designer's choice [6].

These nonlinear control laws have the disadvantage of requiring an accurate knowledge of the nonlinear system dynamics in order to perform the explicit cancellation (in the case of NDI) and that finding Control Lyapunov functions (CLFs) for higher-order dynamic plants is generally not easy (in the case of backstepping). Moreover, the design and optimization of transient responses is difficult when applying nonlinear control. In that sense, designing for stabilization might not be sufficient. Regarding aerospace applications, in order to apply such model-based nonlinear control methods successfully, this means that *both* the model of the system must match the onboard model and practically all nonlinearities must be known accurately. These assumptions are hardly met in reality and in practice and are the main reasons behind the motivation of further developing this methodology in terms of robustness. This robustness aspect is highly important since there is a dependency of the inner–loop of the control system on the model that is critical, i. e., the stability and performance of the system can be compromised when performing under model and sensor uncertainties. Moreover, reaching the actuator limits can also be problematic in terms of dynamic inversion and nonlinear control. The saturation of the actuators and constraints of the control input variables can potentially degrade the closed–loop system and even compromise the overall system stability.

Several improvements have been made regarding these aforementioned flaws of NDI-based control laws, specially with regards to robustness. Some of these improvements were focused on the robustness of the overall control architecture by applying robust control in the outer loop of the system. A combination of NDI with the structured singular value (μ-analysis) and $\mathscr{H}_\infty$ synthesis for reentry flight clearance was done in [51, 69] where noticeable benefits were obtained in comparison with conventional NDI. However, these improvements came with the introduction of some conservatism since the uncertainties were not considered fully or they were covered by lumped uncertainties. These aspects could be further improved considering an incremental nonlinear control approach, presented next.

1.2.3. THE INCREMENTAL NONLINEAR CONTROL APPROACH

Theoretical development of *increments of nonlinear control action* dates back to the late nineties and started with activities concerning 'implicit dynamic inversion' for DI-based flight control in the works of Smith, Bacon *et al.* [70, 73]. Other designations for these developments found in the literature are 'modified NDI' and 'simplified NDI', but the designation 'incremental' is considered to describe the methodology and nature of these type of control laws better [71, 77–80].

INDI has been elaborated theoretically and applied in the past decade to advanced flight control applications [70, 71, 73–75, 80] as well as in space applications for spacecraft attitude control [4]. More recently, this technique has been applied also to fault–tolerant control of aircraft subjected to sensor and actuator faults [81–83], in practice for quadrotors using adaptive control [84–86], and in real flight tests of small (FASER) unmanned aircraft [87, 88] and a business jet (Cessna Citation II, PH-LAB) aircraft [89, 90]; verifying INDI's performance and robustness properties against aerodynamic model uncertainties and disturbance rejection as studied in detail in [71, 86, 91, 92]. Moreover, the incremental nonlinear control approach has also been considered for motion control in mechanical systems and robotics [93].

To motivate the use of increments of nonlinear control, consider as a starting point and without loss of generality aerospace vehicle dynamics that can be described as n-dimensional multivariable nonlinear systems affine in the m inputs u_i and with m outputs y_i:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{1.3a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{1.3b}$$

where $\boldsymbol{x} \in \mathscr{R}^n$, $\boldsymbol{u} \in \mathscr{R}^m$, and $\boldsymbol{y} \in \mathscr{R}^m$. The functions $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are assumed to be continuously differentiable on $\mathscr{R}^n$ and the functions $\boldsymbol{g}(\boldsymbol{x}) = [\,\boldsymbol{g}_1(\boldsymbol{x}) \quad \dots \quad \boldsymbol{g}_m(\boldsymbol{x})\,]^\top \in \mathscr{R}^{n\times m}$ are assumed to be continuous functions of the state vector $\boldsymbol{x}$. Typical control law designs depart with a Jacobian linearization about a particular *equilibrium* or *operational* point of interest [61, 65, 76, 94] $(\boldsymbol{u}_0, \boldsymbol{x}_0, \boldsymbol{y}_0)$ where $\dot{\boldsymbol{x}}_0 = \boldsymbol{0}$ as:

$$\dot{\boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}}\Big[\boldsymbol{f}(\boldsymbol{x}_0) + \boldsymbol{g}(\boldsymbol{x}_0)\boldsymbol{u}_0\Big](\boldsymbol{x}-\boldsymbol{x}_0) + \boldsymbol{g}(\boldsymbol{x}_0)(\boldsymbol{u}-\boldsymbol{u}_0), \tag{1.4a}$$

$$\boldsymbol{y} - \boldsymbol{y}_0 = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_0)}{\partial \boldsymbol{x}}(\boldsymbol{x}-\boldsymbol{x}_0), \tag{1.4b}$$

where, considering deviations from the equilibrium point of interest of the state variables, output variables, and control inputs as:

$$\tilde{\boldsymbol{u}} := \boldsymbol{u} - \boldsymbol{u}_0, \quad \tilde{\boldsymbol{x}} := \boldsymbol{x} - \boldsymbol{x}_0, \quad \tilde{\boldsymbol{y}} := \boldsymbol{y} - \boldsymbol{y}_0, \quad \dot{\tilde{\boldsymbol{x}}} := \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0,$$

a linear system in state-space form can be obtained as:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\tilde{\boldsymbol{u}}, \tag{1.5a}$$

$$\tilde{\boldsymbol{y}} = \boldsymbol{C}\tilde{\boldsymbol{x}}, \tag{1.5b}$$

where the definitions for the matrices $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{C}$ are:

$$\boldsymbol{A} = \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}\right]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}}, \quad \boldsymbol{B} = \frac{\partial}{\partial \boldsymbol{u}}\left[\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}\right]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}}, \quad \boldsymbol{C} = \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}_0}.$$

Since such Jacobian linearization is only valid *locally* at the equilibrium point of interest $(\boldsymbol{u}_0, \boldsymbol{x}_0, \boldsymbol{y}_0)$, a control law must be then designed for operation only around such a point. Otherwise, performance, and even stability may be compromised. The concept of incremental control action amounts to finding a control law $\boldsymbol{u}$ for systems described in Eqs. (1.3) when they are expressed in the so–called *incremental form* [4, 70, 71, 73–75, 95], which is also an approximation of the nonlinear system by Jacobian linearization or Taylor-series expansions, but now considering deviation variables from operational points that may not necessarily represent a particular equilibrium, i. e., such as:

$$\Delta\boldsymbol{u}(t) := \boldsymbol{u}(t) - \boldsymbol{u}_0(t), \quad \Delta\boldsymbol{x}(t) := \boldsymbol{x}(t) - \boldsymbol{x}_0(t), \quad \Delta\dot{\boldsymbol{x}}(t) := \dot{\boldsymbol{x}}(t) - \dot{\boldsymbol{x}}_0(t),$$

where the current control and state, $\boldsymbol{u}_0$ and $\boldsymbol{x}_0$, respectively, represent –for each time instance– the reference an incremental instance in time before $\boldsymbol{u}$ and $\boldsymbol{x}$ for the construction of the first–order approximation of the system. Notice that as a result, $\dot{\boldsymbol{x}}_0(t)$ is not necessarily zero as in the case of equilibrium points. The time–dependency notation of these deviation variables will be omitted in the remainder of this thesis to simplify notation. Furthermore, in contrast to this incremental nature of the approximation where the partial derivatives are interpreted in a geometric sense [70, 71, 73–75], i. e., with respect to a point $(\boldsymbol{u}_0, \boldsymbol{x}_0)$, in this thesis we also consider explicitly a *sufficiently small* time–delay λ that brings an interpretation of the linearization about the $\lambda-$delayed signals [96]:

$$\boldsymbol{u}_0(t) := \boldsymbol{u}(t-\lambda), \quad \boldsymbol{x}_0(t) := \boldsymbol{x}(t-\lambda), \quad \dot{\boldsymbol{x}}_0(t) := \dot{\boldsymbol{x}}(t-\lambda),$$

of the current control input, state, and state derivative, respectively. This means an approximate linearization about the $\lambda-$delayed signals is performed *incrementally*, and not with respect to a particular equilibrium or operational point of interest. Finding a suitable control law for the newly introduced *incremental control* input $\Delta\boldsymbol{u}$ leads to a control design for the system:

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_0 + \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}\right]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}}(\boldsymbol{x}-\boldsymbol{x}_0) + \frac{\partial \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}}{\partial \boldsymbol{u}}\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}}(\boldsymbol{u}-\boldsymbol{u}_0) + \mathcal{O}(\Delta\boldsymbol{x}^2) \tag{1.6}$$

$$\cong \dot{\boldsymbol{x}}_0 + \boldsymbol{F}_0\Delta\boldsymbol{x} + \boldsymbol{G}_0\Delta\boldsymbol{u}, \tag{1.7}$$

where $\boldsymbol{F}_0 := \boldsymbol{F}(\boldsymbol{x}_0(t), \boldsymbol{u}_0(t))$ and $\boldsymbol{G}_0 := \boldsymbol{G}(\boldsymbol{x}_0(t))$ represent online Jacobian linearizations of the on-board model $\boldsymbol{f}(\boldsymbol{x})$ and the control derivatives $\boldsymbol{g}(\boldsymbol{x})$, respectively. From this point forward, the nature of the control law designs follows from some kind of assumption. The most common assumption in incremental nonlinear control is one that involves high-sampling rates of the closed-loop control system, together with fast control action. Such assumption can be summarized as follows:

Time-scale separation assumption: *For a sufficiently small time-delay λ and for any incremental control input, it is assumed that $\Delta\boldsymbol{x}$ does not vary significantly during λ. In*

other words, the input rate of change is much faster than the state rate of change so it can be ignored:

$$\epsilon_{TSS}(t) \equiv \Delta \boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_0(t) \cong 0, \ \forall \ \Delta \boldsymbol{u}, \tag{1.8}$$

which leads to:

$$\dot{\boldsymbol{x}} \cong \dot{\boldsymbol{x}}_0 + \boldsymbol{A}_0 \underbrace{(\boldsymbol{x} - \boldsymbol{x}_0)}_{\cong 0} + \boldsymbol{G}_0 (\boldsymbol{u} - \boldsymbol{u}_0), \tag{1.9}$$

in other words:

$$\Delta \dot{\boldsymbol{x}} \cong \boldsymbol{G}_0 \cdot \Delta \boldsymbol{u} \tag{1.10}$$

For small time increments and high sampling rates, the nonlinear system dynamics in its incremental form are approximated by the gain matrix $\boldsymbol{G}_0 := \boldsymbol{G}(\boldsymbol{x}_0(t))$, which we will refer to as the *instantaneous control effectiveness (ICE)*, i. e., the control effectiveness evaluated at the current state. Meaning that this model-based term is sampled or scheduled at each incremental instant.

Since this results in a change of state variables, referring back to the original or *absolute* state space variables, the control input $\boldsymbol{u}$, to be yet designed, can be represented in generic form as:

$$\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{u}_0, \boldsymbol{x}, \dot{\boldsymbol{x}}_0) = \boldsymbol{u}_0 + \Delta \boldsymbol{u}(\boldsymbol{x}, \dot{\boldsymbol{x}}_0, \boldsymbol{G}_0) \tag{1.11}$$

It can be concluded that such *incremental–linearization*, and its respective incremental nonlinear control, has a clear dependency not only on the incremental state $\boldsymbol{x}_0$ and inputs $\boldsymbol{u}_0$, but also on the incremental state derivatives $\dot{\boldsymbol{x}}_0$. This approach for control design results in an improvement of the robustness of the closed-loop system compared to conventional nonlinear control (NDI, backstepping) since dependency on the accurate knowledge of the plant dynamics is reduced. The approach is inherently *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed, but result when the feedback loops are closed [73, 74].

However, previous theoretical stability and robustness proofs for INDI controllers have many drawbacks and were not mathematically consistent as pointed out in [97, 98] since they were mostly based on simplifying assumptions, approximated transfer functions, and block diagrams [71, 84, 85]. Recently, the INDI control in the literature has been reformulated [97, 98] for systems with arbitrary relative degree and without recurring to cascaded-control structures, i. e., without using a time–scale separation assumption. The reformulation was also considered to extend further the incremental nonlinear control approach to Sliding Mode Control [99] showing potential in robust fault-tolerant flight control since it can reject a wider range of uncertainties and disturbances. For these new reformulations and extensions, the conditions for stability and robustness analyses have been finally established, and were obtained and analyzed using Lyapunov-based methods [97–99]. This important step in regards to conditions for stability and robustness makes the method more tractable and suitable for future real applications.

In the context of this thesis, we deal with the nonlinearities and uncertainties arising in aerospace attitude control considering this *incremental nonlinear control approach*, which reduces model dependency while making use of actuator output and angular acceleration feedback. This thesis investigates ***incremental nonlinear dynamic inversion***

(INDI) and ***incremental backstepping*** (IBKS) approaches for aerospace applications, as presented in the bold boxes of Figure 1.3.

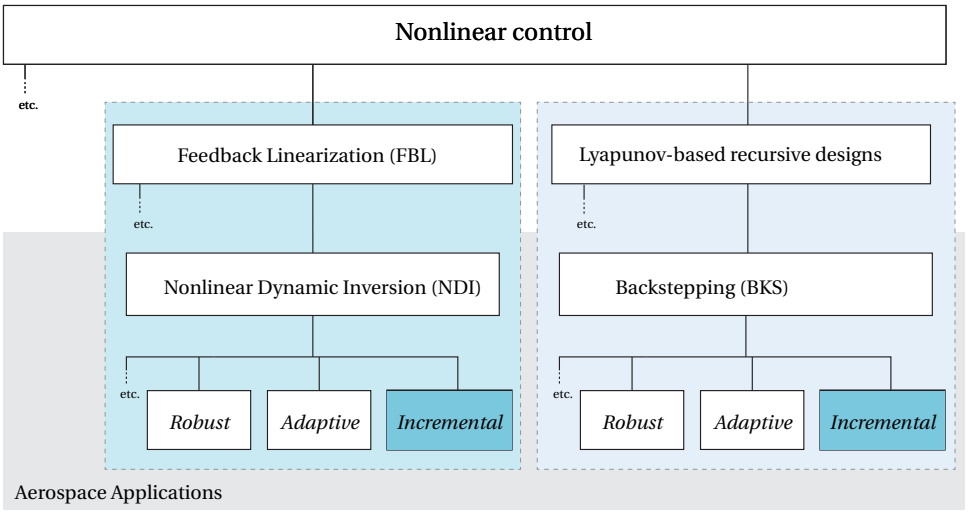


Figure 1.3: The considered nonlinear control framework.

1.2.4. LIMITATIONS AND ASSUMPTIONS

In this thesis the following limitations and assumptions must be kept in mind:

- This thesis only considers rigid bodies and their feedback motion control.
- The scope is limited to aerospace system dynamics that are affine in the control inputs and described in the generic nonlinear form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}$. Other approaches for the robustness of nonlinear control systems in aerospace have been studied in the literature: systems in the forms $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ were treated in Falkena *et al.* [100, 101] and Sun *et al.* [102–104] with a sensor–based backstepping and singular perturbations approach whereas systems in the form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}})$ were considered by Smeur *et al.* [84–86].
- It is assumed that complete and accurate knowledge about the state of the system is available, which implies that the availability of $\dot{\boldsymbol{x}}_0$ may be measured directly or derived from inertial measurement unit (IMU) gyro measurements and filtered accordingly, while the availability of $\boldsymbol{u}_0$ may be computed directly in the algorithm (using continuous or sampled–time delays), provided as a direct measure from the actuator output, or obtained with an accurate model of the actuator dynamics;
- For practical implementations, we may consider first–order and second–order dynamics for each actuator and furthermore, we do not consider these actuator dynamics in the control design process. For that reason, we also assume that these actuators are *sufficiently fast* in the control-bandwidth sense, meaning that the actuator bandwidth is higher than the control system closed–loop bandwidth.

- This thesis expects the incremental approximation $\dot{\boldsymbol{x}} \cong \dot{\boldsymbol{x}}_0 + \boldsymbol{G}_0(\boldsymbol{u} - \boldsymbol{u}_0)$ to hold in the domain of operation. A limitation arises since control design departing from this approximation does not have a mathematically consistent stability proof when applied back to the original system. In fact, the recent reformulated INDI and incremental SMC [97–99] provided sufficient conditions for stability and robustness for the *actual* system $\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_0 + \boldsymbol{G}_0(\boldsymbol{u} - \boldsymbol{u}_0) + \boldsymbol{\delta}(\boldsymbol{x}, t)$, where the corresponding approximation, i.e., $\boldsymbol{\delta}(\boldsymbol{x}, t) \cong 0$, should be made *after* control design.

1.3. RESEARCH QUESTION AND OBJECTIVES

1.3.1. RESEARCH QUESTION

Advancing the capabilities to address the aspects just presented gives rise to the following overarching research question for this thesis:

> *How can the incremental nonlinear control approach be applied to improve agility and robustness of aerospace vehicles' attitude control systems?*

This research question focuses on two aspects: 1) attitude motion and control, which is considered to be one of the classical problems in nonlinear and multi-variable control systems; 2) incremental nonlinear control, which is a combined model– and sensor-based control approach which has shown promising results in the aerospace community. The high-performance and robustness of incremental nonlinear control comes from the partial dependency removal of an accurate plant model by just requiring a control effectiveness model to estimate the so-called *incremental* dynamics, while relying on angular acceleration and actuator output measurements. In that sense, relying on angular accelerations and not explicitly on aerodynamic or 'on-board' models makes the method robust to modeling and parametric uncertainties and allows for aggressive motion control. For that reason, in order to answer the main research question, we consider the full nonlinear motion behind the attitude of aerospace vehicles and focus on the mentioned incremental nonlinear control methodology as line of research.

1.3.2. OBJECTIVES

Having presented the motivation and approach for the current problem considered in this thesis, namely the nonlinear attitude motion and control of aerospace vehicles, the objective of this thesis is to develop concepts and methods for nonlinear flight and attitude control design aspects within a multi-disciplinary modeling and simulation approach. With this approach, attitude dynamics and control can play a more important role in the outcomes of aerospace vehicle design and therefore should be considered more and more within the preliminary design studies of aerospace vehicles. The objectives of this thesis can be encompassed briefly in the following research questions.

Research Question 1

How can an integrated, acausal, and multidisciplinary approach for modeling and simulation support preliminary design studies of space launch vehicles?

This first question relates to dynamics modeling and simulation efforts that might be considered in support of preliminary design studies of aerospace vehicles. Launch vehicle dynamics modeling is quite challenging, mainly because of the highly interconnected disciplines involved: propulsion, aerodynamics, structures, avionics, mechanisms, and GNC among others. Discipline experts perform their respective design often independently and with separate dedicated tools. Consequently, during launcher preliminary design studies, numerous iterations are required in order to keep mission objectives synchronized. In that sense, the development of an integrated and multidisciplinary approach for modeling, analysis, and simulation of space launchers could potentially support their preliminary design efforts. This could allow to reduce the number of iterations and the associated costs, and therefore is a key technology to aim for. Such modeling frameworks were already introduced for aeronautics applications [60, 105], robotics [106, 107], flexible bodies [108], visualization and virtual reality [109], optimization [110], and most recently for satellites [111]. In this thesis the first building blocks are presented towards a framework that could enable physical modeling of conventional and non-conventional launch vehicles and facilitate early developments regarding preliminary vehicle designs.

Early efforts on the subject of launch vehicle dynamics modeling were carried out by NASA during the 60's and 70's to study stage launch vehicle separation dynamics [112–114]. This led to the development of their generalized trajectory simulation, guidance design, and optimization software *Program to Optimize Simulated Trajectories* POST [115] , and its more recent follow-up, POST2 which contains the capability to study separation dynamics in the development of next generation space launchers [1, 116, 117]. However, some of those tools [116, 118–120] have the disadvantage of not being easily integrable in a generic simulation software which eludes the capability of performing end-to-end launch vehicle trajectory simulations. On the European side, early efforts on multibody dynamics for space applications were also carried out by the European Space Agency (ESA) with their *Dynamic and Control Analysis Package* DCAP [121–123]. More recent efforts for developing and consolidating knowledge in launcher dynamics [124, 125], led ESA to develop a launcher multibody dynamics simulator using DCAP as a backbone [126].

Noticing that multidisciplinary dynamics modeling is becoming increasingly important for launch vehicle design and simulation, and that none of these previous dedicated developments fully profits from an object–oriented, equation–based, and acausal modeling language like MODELICA, the objective of this research question is to investigate an alternative approach employing this modeling methodology and to study how such an approach can support early activities in space launcher design, as it is the case for stage separation analyses.

Research Question 2

How can model–based nonlinear control and multi–objective optimization be combined for the study of preliminary guidance and control (G&C) aspects of reusable launch vehicles and spacecraft slew maneuvers?

This second question is related to preliminary considerations in G&C for controllability assessment and agility design from a combined trajectory optimization and nonlinear control perspective.

Preliminary considerations of G&C in aerospace are important since they can identify early on some of the potential impacts on the overall design of the considered aerospace vehicle. For instance, to keep up with the rapidly evolving international launch market, continuous investigation of different G&C methods and technologies (for vertical takeoff and vertical landing (VTVL) or for horizontal takeoff and horizontal landing (HTHL)) to achieve reusability are necessary. After the typical computation of optimal flight trajectories, detailed studies and assessments of the launcher performance during all relevant flight phases should be performed. These include controllability and stability studies, stage separation analyses, and preliminary design of the control systems among many others.

Considering some of the studies on future expendable and reusable launch vehicle configurations conducted at DLR [8–16, 24], the main challenge arises from the fact that these studies most often do not consider controllability aspects in detail. This is because they are usually based on 3–DOF models where attitude dynamics are not considered. For 6–DOF models, general purpose flight control architectures can be designed to track the reference trajectory using attitude control while based on a time–scale separation assumption [49–51]. Some applications for nonlinear flight control and more advanced methods involving robustness were already discussed in *Section 1.2*. These general purpose G&C architectures are quite effective, however the approach in this thesis is mostly based on [60] where model–based nonlinear control is combined with optimization and therefore leads to a more integrated approach.

Another important G&C subject of consideration is the extensive topic of optimal spacecraft reorientation [127–135]. More specifically, the challenge of designing time-optimal slew maneuvers which are, in general, not of the *Euler-axis* rotation type [131, 136, 137]. Some results from the literature for imaging satellites have even been experimentally validated in–orbit [138]. However, most of the work reported in literature does not consider the challenge of designing time–optimal control solutions for a spacecraft equipped with reaction wheels that are commanded by sampled-time control inputs, as is the case for the BIROS satellite introduced in *Section 1.1*. Such time–optimal maneuvers can be mathematically formulated as an optimization problem and therefore be solved numerically with direct methods.

This motivates the objective to find a methodology to combine model–based nonlinear control and multi–objective optimization for the study of preliminary guidance and control (G&C) aspects in aerospace. To realize this objective, the two main applications (space launchers and spacecraft) treated in this thesis are considered to design a

general–purpose G&C architecture. The architecture should allow to study the controllability of space launch vehicles during their preliminary design studies and to design fast slew maneuvers for agile satellites. This can be achieved by considering the sequential methodology or procedure proposed in *Section 1.2*.

Research Question 3

How can incremental nonlinear controls be integrated with, e. g., backstepping, time–delay control (TDC), or nonlinear PID–control? And how can these incremental nonlinear control methods be applied for agile and robust nonlinear spacecraft attitude control?

The focus of this last question is towards improvements and applications of the *incremental nonlinear control approach*, introduced in *Section 1.2* for agile and robust nonlinear attitude control of aerospace vehicles. For future missions, agility (e. g., for Earth observation) and tight (robust) maneuverability (e. g., for hypersonic to subsonic aerodynamic flight control) are desired and expected.

Incremental Backstepping (IBKS) was introduced in [139] with the motivation to combine the design of increments of control action with the recursive procedure of the backstepping control design methodology. This helped to stabilize or track outer-loop control variables of cascaded nonlinear system *incrementally*, accounting for model and parametric uncertainties besides undesired factors such as external perturbations and time delays. However, the first application of incremental backstepping was for robust nonlinear attitude control of rigid spacecraft; this means that the only parametric uncertainty considered was in terms of the moments of inertia. This motivates to deal with large model and parametric uncertainties that arise in flight control systems, mainly because of unmodeled dynamics and aerodynamic uncertainties. In that sense, this thesis proposes the incremental backstepping methodology also as an approach for robust nonlinear flight control.

Time–delay–control (TDC) [140–142], more commonly known in the motion control and robotics community, is a nonlinear control technique that estimates and compensates disturbances and system uncertainties (model and parametric) by utilizing time–delayed signals of some of the system variables. In [142] it has been shown that TDC can be rendered equivalent to a discrete PID–control under some assumptions and some discrete sampling considerations. Since TDC relies on what is called a *time-delay estimation* (TDE), which in turn also depends on some time–delayed signals (as those discussed briefly in *Section 1.2*), this motivates the study of how incremental nonlinear controls are also related to TDC and (nonlinear) PID–control.

Lastly, and as mentioned in *Section 1.2*, incremental nonlinear dynamic inversion (INDI) has been elaborated and applied theoretically in the past decade for advanced flight control applications [70, 71, 73–75, 80, 97, 98], for fault-tolerant control of aircraft subjected to sensor and actuator faults [81–83], and more recently for adaptive control of quadrotors [84–86, 143] and real flight campaigns [87–90]. However, very few applica-

tions for space applications have been considered. The application of incremental nonlinear control to the attitude tracking and disturbance rejection problem of rigid spacecraft in the presence of model and parametric uncertainties therefore can close this gap. Furthermore, the reformulated INDI [97, 98] can be considered to this attitude control problem since it does not rely on a time–scale separation assumption of the closed–loop system.

1.4. CONTRIBUTIONS

The contributions performed during this thesis have resulted in a number of publications in international conference proceedings and journal submissions. In this section the contributions are listed by their appearance order in the thesis.

- **P. Acquatella B.**, *Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies.* In: Proceedings of ICATT 2016, 6th ESA International Conference on Astrodynamics Tools and Techniques. Darmstadt, Germany (2016).
- **P. Acquatella B.**, M. J. Reiner, *Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations.* In: Proceedings of the 10th International Modelica Conference. Lund, Sweden (2014).
- **P. Acquatella B.**, L. Briese, K. Schnepper, *Guidance Command Generation and Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles.* In: **Acta Astronautica**, Vol. 174, pp. 334–346 (2020); presented at *IAC 2018, 69th International Astronautical Congress.* Bremen, Germany (2018).
- **P. Acquatella B.**, *Fast Slew Maneuvers for the High-Torque-Wheels BIROS Satellite.* In: **Transactions of the Japan Society of Aeronautical and Space Sciences**, Vol. 61, No. 2, pp. 79–86 (2018); presented at *ISSFD 2017, 26th International Symposium on Space Flight Dynamics.* Matsuyama-Ehime, Japan (2017).
- **P. Acquatella B.**, E. van Kampen, Q. P. Chu, *Incremental Backstepping for Robust Nonlinear Flight Control.* In: Proceedings of EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control. Delft, The Netherlands (2013).
- **P. Acquatella B.**, W. van Ekeren, Q. P. Chu, *PI(D) tuning for Flight Control Systems via Incremental Nonlinear Dynamic Inversion.* In: **IFAC-PapersOnLine**, Vol. 50, No. 1, pp. 8175–8180 (2017); presented at *IFAC-WC 2017, 20th World Congress of the International Federation of Automatic Control.* Toulouse, France (2017).
- **P. Acquatella B.**, W. Falkena, E. van Kampen, Q. P. Chu, *Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion.* In: Proceedings of the AIAA Guidance, Navigation and Control Conference. Minneapolis, MN, USA (2012).
- **P. Acquatella B.,** Q. P. Chu, *Agile Spacecraft Attitude Control: an Incremental Nonlinear Dynamic Inversion Approach.* In: **IFAC-PapersOnLine**, Vol. 53, No. 2, (2020); presented at *IFAC-WC 2020, 21st World Congress of the International Federation of Automatic Control.* Berlin, Germany (2020).

- **P. Acquatella B.,** E. van Kampen, Q. P. Chu, *A Sampled–Data Form of Incremental Nonlinear Dynamic Inversion for Spacecraft Attitude Control*, *to be submitted.*

1.5. THESIS OUTLINE

Following the objectives just presented, the structure of this thesis is illustrated in Figure 1.4. The first noticeable aspect of the thesis structure is the two main applications considered: *space launchers* and *satellites*. The chapters at the left-hand side of Figure 1.4 are related to modeling, simulation, and control of space launchers, while chapters at the right-hand side are similar in scope but related to spacecraft (satellites) instead. In addition, the thesis is divided into three parts, each related to the three research questions, and labeled from I to III:

PART I of the thesis, consisting of **Chapters 2 – 3**, describes aerospace vehicle dynamics modeling efforts in support of preliminary design studies of space launchers. The contribution of this part is mainly providing building blocks towards a multibody dynamics and equation–based object–oriented modeling approach for these kind of systems. In that sense, this part is related to Research Question 1.

PART II, consisting of **Chapters 4 – 5**, then focuses on preliminary guidance and control (G&C) aspects considering nonlinear design techniques such as nonlinear dynamic inversion and constrained nonlinear optimization. These techniques are shown beneficial for controllability assessments and also for the design of fast slew maneuvers of small satellites. This part is related to Research Question 2.

PART III of the thesis, consisting in the final **Chapters 6 – 10**, brings some contributions to the incremental nonlinear control body of work. These include the assessment of *incremental backstepping* for robust nonlinear flight control and the relation between *incremental nonlinear dynamic inversion*, time-delay control (TDC), and PID–control. This part also applies incremental nonlinear dynamic inversion to robust attitude control of satellites, which has not been widely studied in the literature. Finally, this part is related to Research Question 3.

Conclusions and recommendations are presented in **Chapter 11**.

Each chapter, excluding the introduction and conclusions, begins with a one page '*header*' that includes an abstract, a short list of the key contributions, and a citation referencing where the chapter has been published [4]. Since most of the chapters have been presented at international conferences, or have been published (or submitted for publication) in scientific journals, these chapters can be read separately.

The outline of the thesis, including a brief description of the scope of each chapter is as follows:

Part I – Dynamics Modeling for Preliminary Design Studies

Chapter 2 presents an object-oriented, equation-based, and acausal modeling methodology for launch vehicles using the MODELICA modeling language. This framework enables physics-based modeling of subsystems and components related to most

[4] Some minor modifications have been considered with respect to the presented or published versions; mostly aesthetical and/or typographic.

key analyses of a launcher system. This chapter contributes the first building blocks leading to a multidisciplinary tool for launcher preliminary design studies.

Chapter 3 demonstrates the benefits of the approach presented in the previous chapter in the context of simulation of launch vehicles' stage separation dynamics. Since stage separation dynamics modeling is a critical capability of future launchers preparatory studies, this chapter contributes with a multibody and acausal modeling approach to simulate the physics behind launcher separation events.

Part II – Aerospace Guidance and Control (G&C)

Chapter 4 investigates a general purpose guidance and control (G&C) architecture for preliminary studies of space launchers. The architecture combines physical models with trajectory optimization for guidance command generation and nonlinear dynamic inversion control for the subsequent trajectory tracking. This architecture has benefits for analyzing early stability and controllability aspects since these in turn can have a gross impact to the overall design of the vehicle.

Chapter 5 investigates a high–agility attitude control system for spacecraft actuated by reaction–wheels. Formulating the problem as a constrained nonlinear optimal control problem allows to design time–optimal slew maneuvers in open–loop.

Part III – Robust Nonlinear Attitude Control

Chapter 6 proposes the *incremental backstepping* approach for robust nonlinear flight control. The advantage of the combination of incremental nonlinear control with the backstepping design methodology is showcased by the tracking capability under aerodynamic uncertainty for a simple longitudinal nonlinear flight control example, overcoming some difficulties of conventional adaptive and model–based flight control strategies.

Chapter 7 follows the flight control context of the previous chapter and shows the relationship between incremental nonlinear dynamic inversion, discrete time–delay control (TDC), and discrete proportional–integral–derivative control (PID). The original result, relating PID with TDC comes from the robotics and TDC literature; while here, the relation with INDI is established bringing a new interpretation of the method.

Chapter 8 presents a robust nonlinear spacecraft attitude control system for tracking and disturbance rejection of a rigid spacecraft subjected to model and parametric uncertainties. This is achieved with a cascaded two-loop control system using an outer-loop control in terms of the *Modified Rodrigues Parameters* (MRP) attitude parameterization and using INDI in the inner-loop.

Chapter 9 presents an agile and robust spacecraft attitude tracking controller using the recently reformulated incremental nonlinear dynamic inversion control. The reformulated INDI allows a non-cascaded dynamic inversion control in terms of Modified Rodrigues Parameters (MRPs) where scheduling of the time-varying control effectiveness is done analytically. This way, the controller is only sensitive to parametric uncertainty of the augmented spacecraft inertia and its wheelset alignment. Moreover, we draw some parallels to time-delay control (TDC) which have been shown

to be equivalent to the incremental formulation of proportional-integral-derivative (PID) control for second order nonlinear systems in controller canonical form.

Chapter 10 presents a sampled–data form of the recently reformulated incremental nonlinear dynamic inversion (INDI) applied for robust spacecraft attitude control. The contribution is aimed to bridge the gap between continuous–time and highly sampled INDI formulations and their discrete and lowly sampled counterparts in the context of spacecraft attitude control where low sampling rates are common. Neglecting the sampling time and its effect in the controller derivations can lead to stability and performance issues of the resulting closed–loop nonlinear system. The sampled–data reformulation allows explicit consideration of the sampling time via an approximate sampled–data model in normal form widely known in the literature.

Chapter 11 provides the conclusions of this thesis and some recommendations for further research.

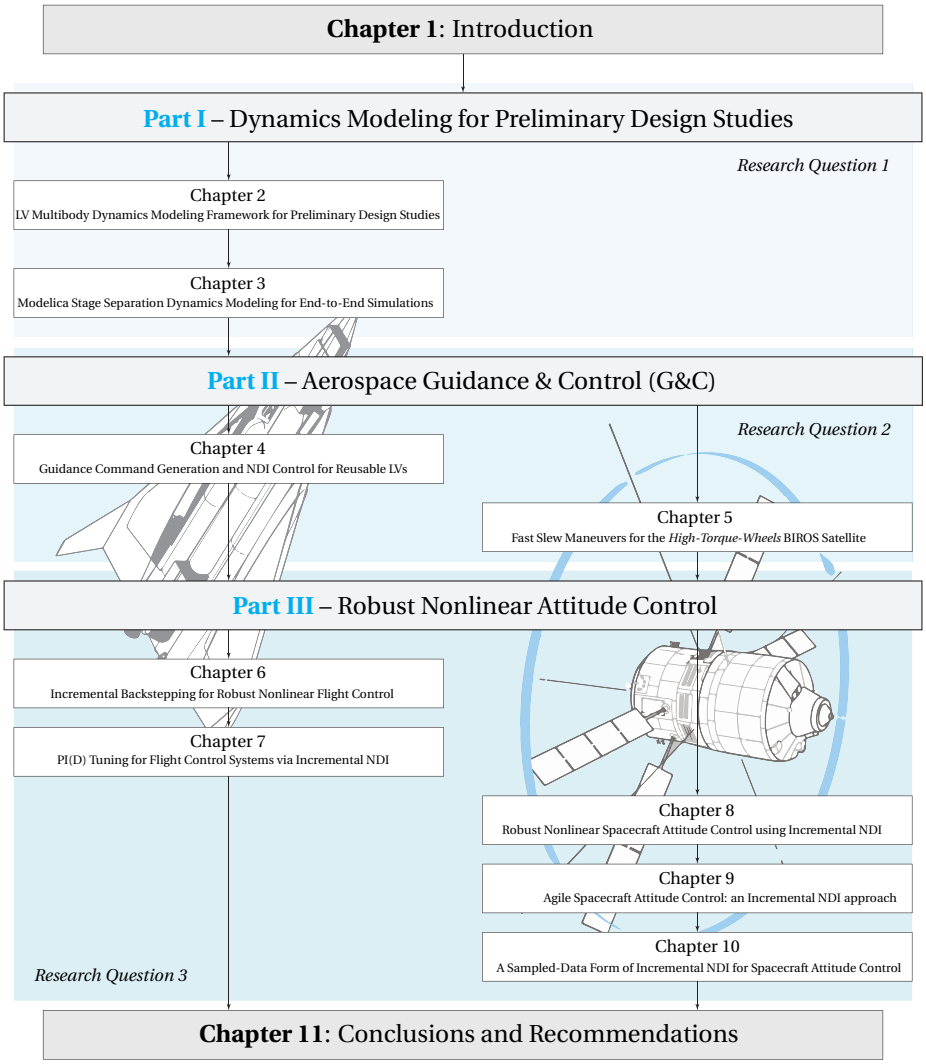


Figure 1.4: Thesis outline. Image credits: Aurora (left) DLR/A. Kopp., ATV (right) ESA/I. Baroncini.

I

DYNAMICS MODELING FOR PRELIMINARY DESIGN STUDIES

2

Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies

Abstract

Launch vehicle dynamics modeling is quite challenging mainly because of the highly interconnected disciplines involved. Discipline experts perform their respective design often independently and with separate dedicated tools. Dedicated developments of multidisciplinary modeling tools for launch vehicle multibody dynamics have been presented in the relevant literature. However, none fully profits from an object-oriented, equation-based, and acausal modeling language like MODELICA. *As yet, such an approach is still missing. It is therefore the objective of this paper to introduce such an alternative approach employing this modeling framework enabling object-oriented and physics-based modeling of subsystems and components related to most key analyses of a launcher system. The paper gives an overview on the first building blocks leading to an integrated and multidisciplinary tool for launcher preliminary design studies. Particularly, its easiness of implementation is demonstrated along with the benefits of this approach.*

Publication

Paul Acquatella B.: *Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies.* In: Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques (ICATT 2016), March 14-17, 2016. Darmstadt, Germany.

2.1. INTRODUCTION

FOR the several architectures and configurations to consider and optimize at preliminary design studies, several launch vehicle models with varying levels of scope and complexity are necessary.

In that sense, launch vehicle dynamics modeling is quite challenging mainly because of the highly interconnected disciplines involved: propulsion, aerodynamics, structures, mechanisms, and GNC among others. Discipline experts perform their respective design often independently and with separate dedicated tools. Consequently, during launcher preliminary design studies, numerous iterations are required in order to keep mission objectives synchronized.

Preliminary design efforts could potentially be reduced by using a multidisciplinary launch vehicle model integrated in one single tool. Because this allows to reduce the number of iterations and the associated costs, a launch vehicle multibody dynamics modeling framework is a key technology to aim for.

Early efforts on the subject of launch vehicle dynamics modeling were carried out by NASA during the 60's and 70's given the importance to study stage launch vehicle separation [112–114]. This led to the development of their generalized trajectory simulation, guidance design, and optimization software *Program to Optimize Simulated Trajectories* POST [115], and its more recent follow-up, POST2. For multibody dynamics, TREETOPS [144, 145] was conceived based on Kane's equations, and followed by the more recent CLVTOPS, both featuring capabilities for multiple flexible body dynamic simulation, separation analysis, and liftoff clearance analysis [146].

On the European side, early efforts on multibody dynamics for space applications were also carried out for over 30 years by the European Space Agency (ESA) with their *Dynamic and Control Analysis Package* DCAP [121–123]. It provides capabilities to model, simulate, and analyze the dynamics and control performances of coupled rigid and flexible structural systems subject to structural and space environmental loads. More recent efforts for developing and consolidating knowledge in launcher dynamics [124, 125], led ESA to develop a launcher multibody dynamics simulator using DCAP as a backbone [126]. This tool has been adapted to meet typical requirements of the ESA Concurrent Design Facility (CDF) environment.

Many other proprietary and commercial tools, like ASTOS developed by Astos Solutions GmbH, are relevant to the launcher modeling and simulation literature, but the extensive list of tools and solutions is not covered here. Noticing that multidisciplinary modeling is becoming increasingly important for launch vehicle design and simulation, and that none of these previous dedicated developments fully profits from an object-oriented, equation-based, and acausal modeling language like MODELICA; the objective of this chapter is to introduce an alternative approach employing this modeling methodology. This approach comes with the first building blocks leading to an integrated and multidisciplinary launcher vehicle dynamics modeling tool.

A brief description of MODELICA as a modeling methodology is given; then an object-oriented and physics-based modeling framework is introduced; followed by a basic mathematical description of a launcher multibody dynamics model; and finally an application example is presented, outlining the key benefits of this approach.

2.2. MODELING METHODOLOGY

MODELICA [37–41] is a modern object-oriented, equation based ***modeling language*** well suited to model complex physical systems containing, e.g., mechanical, electrical, power, hydraulic, thermal, control, or process-oriented subsystems and components.

Models in MODELICA are described using differential, algebraic, and discrete equations which are then mapped into a mathematical description form called hybrid DAE (Differential Algebraic Equations). A DAE system on its implicit form is generally expressed as

$$\mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t), \mathbf{y}(t), \boldsymbol{\rho}, t\right) = 0 \tag{2.1}$$

where $\dot{\mathbf{x}}$ are the state derivatives, $\mathbf{x}$ the state variables, $\mathbf{u}$ the inputs, $\mathbf{y}$ the algebraic variables, $\boldsymbol{\rho}$ the parameters and constants, t the time variable, and the dimension $dim(\mathbf{F}) = dim(\mathbf{x}) + dim(\mathbf{y})$. Systems are then solved and simulated by MODELICA simulation environments. When these systems are represented in the DAE implicit form, they can be solved directly by a DAE solver such as DASSL. Alternatively, the system can be sorted out according to specific inputs and outputs and mapped into an explicit ODE (Ordinary Differential Equation) form by solving for the derivatives and the algebraic variables, and then subsequently solved numerically by an ODE solver. The process and details of MODELICA's code compilation is out of the scope of this chapter.

MODELICA MAIN FEATURES

In contrast to imperative languages, in which statements and algorithms are assigned in explicit steps, MODELICA is ***declarative***, meaning that declarations are given through equations [41]. These declarations most often describe model's first-principles at their lowest levels without explicit orders or *how* to compute them, hence why MODELICA is said to be ***equation based***. By means of specialized algorithms, these declarative models are translated into efficient computer executable code. This allows ***acausal*** modeling capabilities that give better reuse of classes since equations do not specify a certain data flow direction. This is therefore one of the most important features of the language.

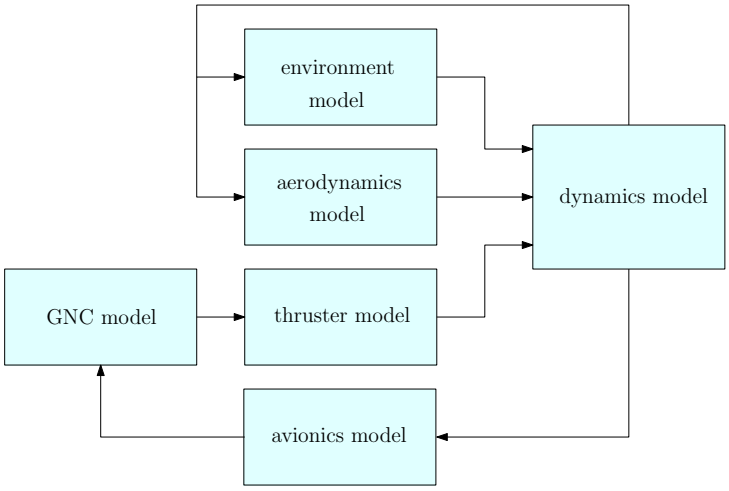
MODELICA is *domain neutral*. In other words, it has ***multidomain*** modeling capability, meaning that model components corresponding to physical objects from several different domains can be described and connected. This interaction between components is defined by means of physical ports, called connectors, and the interconnection is given accordingly to their physical meaning. This meaning is typically represented by flow variables, which describe quantities whose values add up to zero in a node connection (Kirchhoff's first rule); and by non-flow (or potential) variables, which in contrast remain equal (Kirchhoff's second rule).

MODELICA is an ***object-oriented*** language. This helps to model systems and their physical meaning within an object-oriented structure, facilitating the reuse of component models and the evolution of the structure itself. Thus, object-orientation is primarily used as a ***structuring*** concept which exploits the declarative feature of the language, as well as the re-usability of models.

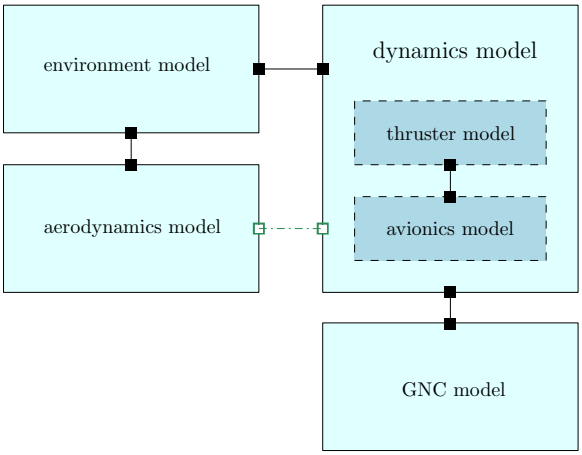
MODELICA has a strong software component model with constructs for creating and connecting components in a ***modular*** fashion. Systems' individual components are defined separately as objects, and their interconnection is given accordingly to their phys-

ical meaning. Thus the language is ideally suited as an ***architectural*** description language for complex physical systems.

2.3. MODELING FRAMEWORK



(a) Classical input-output representation.



(b) Acausal approach, or energy exchange representation.

Figure 2.1: Classic approach vs. acausal approach.

A framework for the physical modeling of conventional and non-conventional launch vehicles is presented here. In contrast to the classical signal-based approach, where systems are mainly considered and modeled as signal processors with a fixed causality, this approach employs an acausal approach where systems exchange energy, see Figure 2.1. In there, the connectors in the acausal approach represent a physical interaction where

an energy balance is applied.

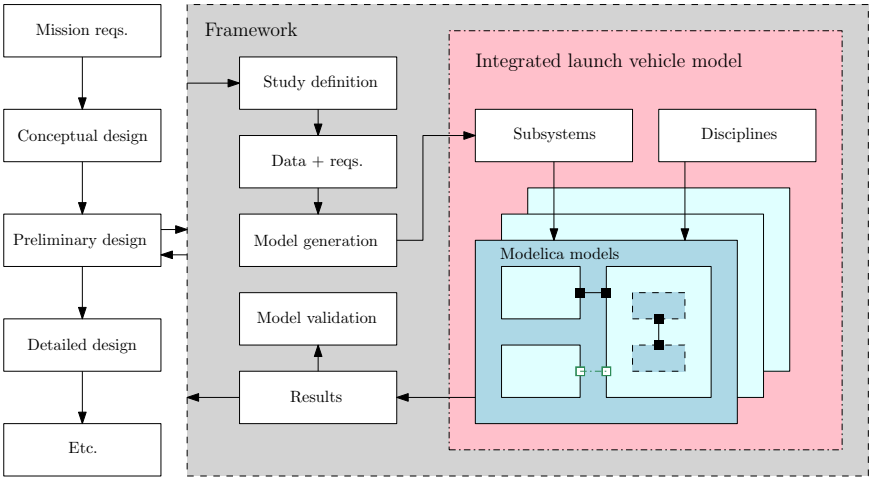


Figure 2.2: Overall picture of the framework.

MAIN FEATURES

The framework consists of a structured and *object-oriented architecture* which enable combinations of several sets of system and subsystem models, themselves built and composed into components and interfaces corresponding to different physical domains (mechanical, electrical, structural, control, etc.) and therefore described from their first principles with the MODELICA language.

Referring to Figure 2.2, given a particular study definition (3-DOF/6-DOF performance, stability and controllability, optimization, etc.) of a preliminary design phase, the first step of the framework is to obtain all necessary data and specific requirements of the study in order to properly generate a particular launch vehicle model. Once the key subsystems and disciplines interacting are properly identified, a multidisciplinary launch vehicle model integrated in one single tool is used to generate study results. For this reason, this tool is quite *versatile*.

In this sense, subsystems of a launch vehicle, as well as the launch vehicle system itself can be modeled within a single simulation environment, and without necessarily implementing coupling interfaces to other specialized tools. This allows the capability of performing end-to-end launch vehicle trajectory simulations as it will be shown in the application example.

To provide application-specific capabilities, the generic functionality of the framework can be tailored and extended by additional user-specific code. For instance, the framework may include databases, pre-processing and post-processing scripts, several MODELICA libraries, interfaces to commercial software like MATLAB&SIMULINK (available for instance in DYMOLA), combination of multibody and FEM [108], and application programming interfaces (APIs) to other tools.

The framework implementation is based upon the extension of the *DLR Space Sys-*

tems Library, introduced in [111], in order to enable object-oriented and physics-based modeling of subsystems and components related to launch vehicle system dynamics.

The main feature of the library is the **World** component. It defines basis coordinate systems such as the Earth Centered Inertial (ECI) and the Earth Centered Earth Fixed (ECEF) coordinate systems, and manages calendar and Julian times. Most notably, it provides capabilities to instantiate multiple gravity models of different kinds of complexity, up to the most precise EGM96 gravity model [147]. Moon and sun perturbation terms to the gravity models are also available. The library also contains state-of-the-art space environment models like the NRLMSISE-00 atmospheric density model [148].

This library builds upon the *Modelica Standard Library* [149, 150], the *Modelica Multi-Body Library* [107], the *DLR Flight Dynamics Library* [105], the *DLR Flexible Bodies Library* [108], the *DLR Visualization Library* [109] and the *DLR Optimization Library* [110].

2.4. MULTIBODY DYNAMICS MODEL

Typically, a multibody system is described by a collection of bodies and their interactions.

The interactions, representing physical coupling of the bodies, can be described as rigid connections between frames (Section 2.4.1); joints representing motion constraints (Section 2.4.2), useful for meaningful physical joint models (prismatic joints featuring, e.g., spring-damper actuators); or even special elements describing more complex dynamic behavior like joint motion and separation dynamics (Section 2.4.3).

Bodies are represented by their physical properties (mass, moments of inertia, etc.) and a collection of frames located at special points of interest (center of mass, joint locations, reference points, etc.). Their translational and rotational dynamics are described depending on the physical nature of the system and their components, for instance, Newton-Euler equations of motion in the case for rigid body models. Here, variable mass systems are described by Kane's equation as obtained by Eke [151] (Section 2.4.4).

2.4.1. FRAMES

Recalling the concept of acausal connectors of Figure 2.1-(b), a *frame* connector from MODELICA's *Multibody Standard Library* [107] is a coordinate system fixed to a model component with a cut-force and a cut-torque as flow variables, and with a position and an orientation object as non-flow variables. Subsequently, mechanical components can be interconnected together rigidly at this frame.

The dynamics of a frame A is completely described by its *generalized position* $\hat{\mathbf{r}}_A$, *velocity* $\hat{\mathbf{v}}_A$, *acceleration* $\hat{\mathbf{a}}_A$, and *force* $\hat{\mathbf{f}}_A$, respectively

$$\hat{\mathbf{r}}_A = \begin{bmatrix} \mathbf{r}_A \\ \mathbf{R}_A \end{bmatrix}, \qquad \hat{\mathbf{v}}_A = \begin{bmatrix} \mathbf{v}_A \\ \boldsymbol{\omega}_A \end{bmatrix},$$
$$\hat{\mathbf{a}}_A = \begin{bmatrix} \mathbf{a}_A \\ \boldsymbol{\alpha}_A \end{bmatrix}, \qquad \hat{\mathbf{f}}_A = \begin{bmatrix} \mathbf{f}_A \\ \boldsymbol{\tau}_A \end{bmatrix},$$

where $\mathbf{r}_A$, $\mathbf{v}_A$, and $\mathbf{a}_A$ are the absolute position, velocity, and acceleration of the frame

A with respect to an inertial frame; $\mathbf{R}_A$, $\boldsymbol{\omega}_A$, and $\boldsymbol{\alpha}_A$ the attitude direction cosine matrix, absolute angular velocity, and angular acceleration of the frame A with respect to an inertial frame; and $\mathbf{f}_A$, $\boldsymbol{\tau}_A$ the resulting forces and torques at frame A [106].

For rigidly interconnected frame connectors, say frames A and B, and as mentioned in the modeling methodology section, the kinematic quantities related to the non-flow variables $\hat{\mathbf{v}}_A$ and $\hat{\mathbf{v}}_B$ are equal to each other, whereas the flow variables, cut-forces and cut-torques $\hat{\mathbf{f}}_A$ and $\hat{\mathbf{f}}_B$ in this case, sum up to zero [106, 107]. This is due to a power P balance constraint considering that no energy is stored:

$$\sum P = \mathbf{0} = \hat{\mathbf{f}}_A^\top \hat{\mathbf{v}}_A + \hat{\mathbf{f}}_B^\top \hat{\mathbf{v}}_B \tag{2.2}$$

2.4.2. JOINTS

Specific joint interconnections in multibody dynamics are very useful to interconnect mechanical systems featuring a non-rigid and physically-meaningful joint motion.

For that, consider a *generalized joint coordinate* $\mathbf{q}$ allowing certain motions between two frames A and B, and its associated *generalized joint force* $\boldsymbol{\lambda}$. Because of the newly allowed motion, additional relationships between the connected frames are necessary. These are given as functions of $\mathbf{q}$ (and possibly $\dot{\mathbf{q}}$) and in terms of the relative quantities between the frames [106].

The corresponding description between the connected frames A and B can be determined similarly as before from a power balance constraint because no energy is stored in such an ideal joint

$$\sum P_i = \mathbf{0} = \hat{\mathbf{f}}_A^\top \hat{\mathbf{v}}_A + \hat{\mathbf{f}}_B^\top \hat{\mathbf{v}}_B + \boldsymbol{\lambda}_A^\top \dot{\mathbf{q}} \tag{2.3}$$

In that sense, the dynamics of a the joint is also completely described by its related generalized quantities. Since the elements of $\dot{\mathbf{q}}$ are independent from each other, the last expression leads to a constraint equivalent to d'Alembert's principle, see [106].

2.4.3. AUTOMATIC JOINT LOADS COMPUTATION

For launch vehicle staging and separation dynamics, joint models for both physical connection and separation between bodies are required.

This can be done with MODELICA by automatic joint loads computation [152], which is applied to each of the connected bodies prior to their physical separation and released for their subsequent and independent motion. This is the principle behind the *Constraint Force Equation* (CFE) methodology, developed by NASA for similar kinds of studies [1, 117, 153].

The CFE methodology is a highly intuitive method consisting in the computation of joint loads, namely internal forces and torques, caused by joint constraints; along with their application as external forces and torques on each body independently. In consequence, the CFE joint model simply augments the external loads of the system [117] as shown in Figure 3.1. The constrained equations of motion of two rigid bodies (A and B) connected by a single joint (point $\overline{A}$ in body A and point $\overline{B}$ in body B) [1, 117] are

$$m_A \, \ddot{\mathbf{r}}_A = \mathbf{f}_A^{ext} + \mathbf{f}_A^{con}, \tag{2.4a}$$

$$\boldsymbol{I}_A \, \dot{\boldsymbol{\omega}}_A + \boldsymbol{\omega}_A \times \boldsymbol{I}_A \, \boldsymbol{\omega}_A = \boldsymbol{\tau}_A^{ext} + \rho_A \mathbf{f}_A^{con} + \boldsymbol{\tau}_A^{con}, \tag{2.4b}$$

where ρ_A is the position vector from the mass center of A to point $\overline{A}$, the point at which the constraint force is applied. The similar equation applies for body B, giving so far 12 equations out the 24 unknowns. Another set of six equations can be obtained as

$$\mathbf{f}_A^{con} + \mathbf{f}_B^{con} = \mathbf{0} \tag{2.5a}$$

$$\boldsymbol{\tau}_A^{(con)} + \boldsymbol{\tau}_B^{(con)} + (\mathbf{r}_{\bar{B}} - \mathbf{r}_{\bar{A}}) \times \mathbf{f}_B^{con} = \mathbf{0} \tag{2.5b}$$

where $\mathbf{r}_{\bar{A}} = \mathbf{r}_A + \rho_A$ and $\mathbf{r}_{\bar{B}} = \mathbf{r}_B + \rho_B$. For relative translation and rotation constraints and $\mathbf{e}$ being unit-vectors of the corresponding (A or B) body-frame, it is required that:

$$(\mathbf{r}_{\bar{A}} - \mathbf{r}_{\bar{B}}) \cdot \mathbf{e}_A = 0 \tag{2.6a}$$

$$\mathbf{e}_A \cdot \mathbf{e}_B = 0 \tag{2.6b}$$

To couple Eqs. (3.7) with the equations of motion, these must be differentiated twice with respect to time so that the resulting relationships involve the unknown linear and angular accelerations. In other words, the six missing equations are given by the following *generalized constraint* equations of the joint, $\ddot{\mathbf{g}} = \mathbf{0}$, where $\mathbf{g}$ represents the non-differentiated constraints in Eqs. (3.7).

To improve the accuracy of the joint loads solution, which is sensitive to computational error and initial joint misalignment, the generalized constraint equations are augmented with the *Baumgarte stabilization* [1, 117, 154, 155] as:

$$\ddot{\mathbf{g}} + 2\eta\dot{\mathbf{g}} + \eta^2\mathbf{g} = \mathbf{0},\ \eta > 0 \tag{2.7}$$

As demonstrated in [152], the manual differentiation of Eqs. (3.7) and their coupling with the equations of motion can be avoided altogether in MODELICA since this is done automatically by the declarative feature of the language.

2.4.4. DYNAMICS OF VARIABLE MASS SYSTEMS

Launch vehicles are systems involving considerable changes in motion as well as in mass (and therefore inertia). The extra loads due to the variable mass effects must be included in the formulation of the dynamic equations of motion.

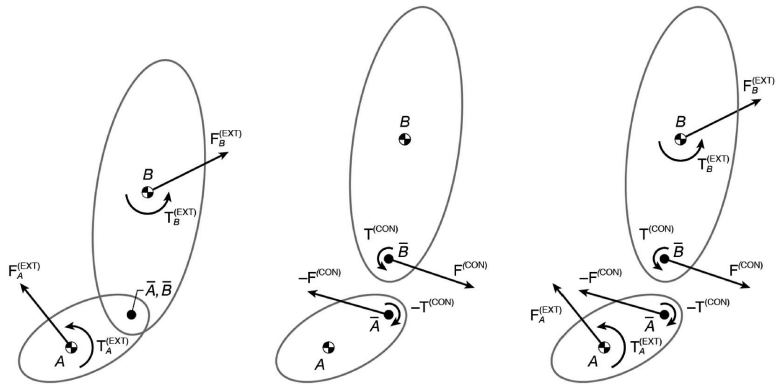


Figure 2.3: CFE diagram. Illustration credits: [1].

Consider for instance a solid rocket motor, a system that loses mass while subject to dynamical motion, and which at any given instant of time is a mixture of both a solid rigid part (R) and a fluid part (F) due to products of combustion. These are delimited by the boundary $\mathscr{B}$.

The dynamic equations of motion for these kind of systems as obtained by Eke [151], and established with Kanes's formalism, are summarized here. In [151], it is claimed that these are identical to those obtained by other authors using a Newton-Euler formulation.

The translational equations of motion are given by

$$m\mathbf{a} = \mathbf{f}^C + \mathbf{f}^L + \mathbf{f}^{thr} + \mathbf{f}^{ext} \tag{2.8}$$

with

$$\begin{aligned}
\mathbf{f}^C &= -2\int_{\mathscr{B}} \rho(\boldsymbol{\omega} \times \mathbf{v}_r)\mathrm{d}V, \\
\mathbf{f}^L &= -\frac{^{(R)}\mathrm{d}}{\mathrm{d}t}\int_{\mathscr{B}} \rho\mathbf{v}_r\mathrm{d}V, \\
\mathbf{f}^{thr} &= -\int_{\mathscr{S}} \rho\mathbf{v}_r(\mathbf{v}_r\cdot\mathbf{n})\mathrm{d}S,
\end{aligned}$$

where $\mathbf{f}^C$ is the Coriolis force, $\mathbf{f}^L$ the system's linear momentum decrease rate relative to the closed surface $\mathscr{B}$, $\mathbf{f}^{thr}$ the thrust vector force, and $\mathbf{f}^{ext}$ the sum of all external forces about the current center of mass of the system, respectively. The left superscript on time derivatives indicates that the derivative is to be taken while the reference frame is kept fixed.

Concerning the thrust vector force, whenever $\mathbf{v}_r \cdot \mathbf{n}$ can be approximated relatively well at the nozzle exit plane, the surface integral can be evaluated in closed form [151]. Using the effective exhaust velocity $v_e = I_{SP}g_0$, a model of the thrust force considering atmospheric losses is given by

$$\mathbf{f}^{thr} = \dot{m}I_{SP}g_0 - S_x P_z(h) \tag{2.9}$$

The attitude equations of motion are given by

$$\mathbf{I}\,\boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{I}\,\boldsymbol{\omega} + \left(\frac{^{R}\mathrm{d}\boldsymbol{I}}{\mathrm{d}t}\right)\boldsymbol{\omega} = \boldsymbol{\tau}^{C_1} + \boldsymbol{\tau}^{C_2} + \boldsymbol{\tau}^{H} + \boldsymbol{\tau}^{thr} + \boldsymbol{\tau}^{ext} \tag{2.10}$$

where

$$\begin{aligned}
\boldsymbol{\tau}^{C_1} &= -\int_B \rho[\mathbf{r}_p \times (\boldsymbol{\omega} \times \mathbf{r}_p)](\mathbf{v}_r \cdot \mathbf{n})\mathrm{d}S \\
\boldsymbol{\tau}^{C_2} &= -\int_B \rho\left[\boldsymbol{\omega} \times (\mathbf{r}_p \times \mathbf{v}_r)\right]\mathrm{d}V \\
\boldsymbol{\tau}^{H} &= -\frac{^{R}\mathrm{d}}{\mathrm{d}t}\int_B \rho(\mathbf{r}_p \times \mathbf{v}_r)\mathrm{d}V \\
\boldsymbol{\tau}^{thr} &= \int_S \rho(\mathbf{r}_p \times \mathbf{v}_r)(\mathbf{v}_r \cdot \mathbf{n})\mathrm{d}S
\end{aligned}$$

$\boldsymbol{\tau}^{C_1}$ is the so-called jet damping, $\boldsymbol{\tau}^{C_2}$ is due to the Coriolis effect and can be neglected for axisymmetric motion as well as for negligible internal flow, $\boldsymbol{\tau}^{H}$ represents the rate of decrease of the system's angular momentum inside $\mathscr{B}$, $\boldsymbol{\tau}^{thr}$ the moment of the thrust vector about the mass center, and $\boldsymbol{\tau}^{ext}$ the sum of all external moments about the current center of mass of the system.

Notice that if $\mathbf{v}_r$ is zero everywhere, then the Newton-Euler equations of motion for a rigid body are recovered. In general, depending on the nature of the propulsion system and its corresponding shape or assumed burn profiles, these terms can be further simplified and further evaluated in closed form, see [151]. In this way, these loads can be included explicitly in the formulation of the dynamic equations of motion of the corresponding element of the vehicle so that their effect can be included in dynamic analyses.

To conclude the main mathematical formulations, aerodynamic forces and moments can be generally expressed in the body-axis frame as

$$\mathbf{f}^{aero} = -q\, S_r C_i(h, \mathbf{v}, \alpha, \beta, ...), \tag{2.11}$$

$$\boldsymbol{\tau}^{aero} = q\, S_r l C_j(h, \mathbf{v}, \alpha, \beta, ...), \tag{2.12}$$

where C_i (for $i = C, Y$, and L) and C_j (for $j = l, m$, and n) are the aerodynamic drag, side force, and lift coefficients, respectively. Finally, the expressions for the dynamic pressure, Mach number, and relative speed are given:

$$\begin{aligned} q &= \tfrac{1}{2}\rho\, \mathbf{v}^2 = \tfrac{1}{2}\gamma P_z(h) M^2, \\ M &= |\mathbf{v}_{rel}| / \mathbf{v}_s(h), \\ \mathbf{v}_{rel} &= \mathbf{v} - \boldsymbol{\omega}_e \times \mathbf{r} \end{aligned}$$

2.5. APPLICATION EXAMPLE

An application example for a 3-DOF open-loop point-mass launcher model featuring stage separation dynamics is presented here.

Separation dynamics is simulated with the automatically obtained joint loads satisfying the CFE constraints. The release device is simulated with a linear cutting charge model, and the separation mechanism with the use of retro-thrusters. Properties for this launcher model are taken from the VEGA launcher users' manual as shown in Table 2.1. Parameters not available were assumed with best guesses.

At $t = 106.8$ s, the first burn is completed and the first stage is separated at $t = 108$ s. Then after a few seconds, at $t = 112$ s, giving enough time for clearance aspects, retro-thrusters are actuated to further separate the first stage from the remaining composite. The sequence is similar for the second stage, where the retro-thrusters are commanded at $t = 190$ s, a few seconds after the second stage separation.

Figure 2.4 presents the stages' altitude (normalized), relative velocity (normalized), and acceleration during their connected motion as well as during their subsequent separate flight motion.

Results shows that the automatically obtained joint loads satisfying the CFE methodology constraints successfully models the launcher system during its connected flight motion. This demonstrate the capabilities as well as the ease of use and implementation under the proposed framework by taking advantage of MODELICA's modeling methodology.

Table 2.1: VEGA User's Manual Data (2006)

Property	Stage 1	Stage 2	Stage 3
Length [m]	11.2	8.39	4.12
Diameter [m]	3	1.9	1.9
Gross mass [kg]	95 796	25 751	10 948
Propellant mass [kg]	88 365	23 906	10 115
Thrust (S/L) [kN]	2261	1196	225
Isp (Vac) [s]	280	289	295
Burn time [s]	106.8	71.7	109.6
Ignition time [s]	0	115	195
Separation command [s]	108	188	-

2.6. SUMMARY AND OUTLOOK

The objective of this chapter was to present an object-oriented and equation-based acausal modeling approach as the first building blocks leading to an integrated and multidisciplinary tool for launcher vehicle dynamics modeling with MODELICA.

Based on MODELICA language as the modeling methodology, we provide a framework which enable object-oriented and physics-based modeling of subsystems and components related to most key analyses of launch vehicle system dynamics. To demonstrate its benefits, a launch vehicle multibody dynamics model is described and implemented within this framework as described with introductory mathematical formulations. Its easiness of implementation is done with an application example.

Future work will be dedicated upon extension of this framework by adding more capabilities, featuring for instance the interconnection of flexible bodies, dedicated algorithms for GNC sizing and design, and most importantly, for optimization studies concerning trajectory, stage sizing, and performance among others.

Moreover, this launch vehicle modeling and simulation framework could in fact support a vast number of use cases across a launcher program life cycle. These may include not only preliminary design phases, but also activities concerning detailed system design, software and component verification and validation, etc.

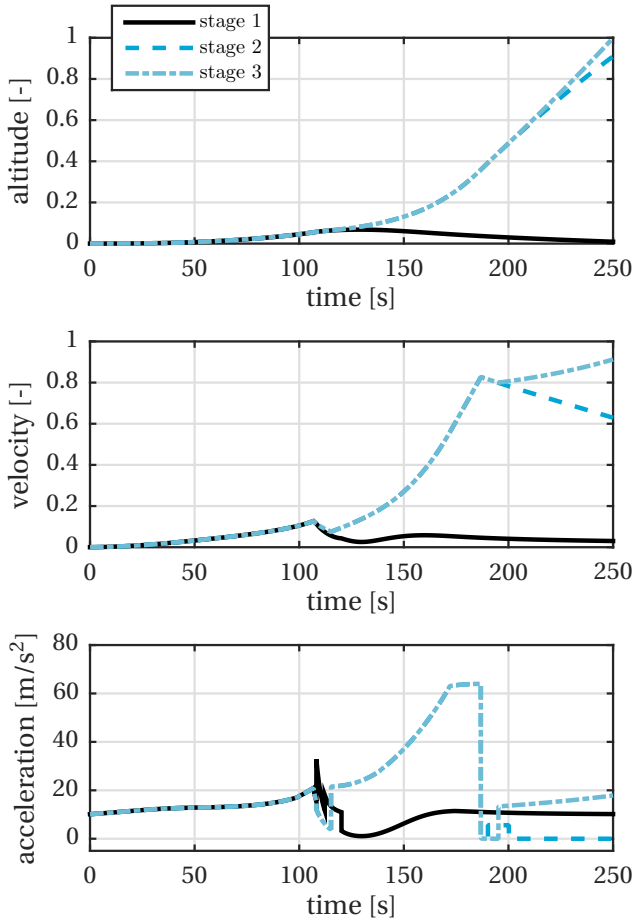


Figure 2.4: Application example results.

3

Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations

Abstract

Stage separation dynamics modeling is a critical capability of future launchers preparatory studies. The development of stage separation frameworks integrable in end-to-end launch vehicle trajectory simulations have been presented in the relevant literature but none profiting from the object-oriented and equation-based acausal modeling properties of MODELICA. *The objective of this paper is therefore to present such an approach to this problematic. Based on the Constraint Force Equation (CFE) methodology, two case studies to evaluate the proposed approach are considered. Results demonstrate that the approach corresponds very well with the physics behind separation. In addition, we found easiness of implementation of the method within a single environment such as* DYMOLA, *demonstrating the benefits of an integrated approach.*

Publication

Paul Acquatella B., Matthias J. Reiner: *Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations.* In: Proceedings of the 10th International Modelica Conference, March 10-12, 2014. Lund, Sweden.

3.1. INTRODUCTION

STAGE separation dynamics modeling is a very challenging task and a critical capability that must be considered in the preparatory studies and development of next generation launchers [1, 116, 117]. The integration of such stage separation modeling into a single environment capable of end-to-end launch vehicle trajectory simulation is also a key technology to aim for.

The importance of such capability arises from the fact that after separation, the integrity of each stage must be kept in order to guarantee overall success of the space mission pursued. In this sense, the development of an integrated framework for analysis and simulation of stage separation is desired.

Early efforts on the subject of multi stage launch vehicle separation from the 60's and 70's are mainly from NASA studies [112–114] and their *Program to Optimize Simulated Trajectories* (*POST*) as a generalized trajectory simulation and optimization software [115], developed in partnership with the (then) Martin Marietta Corporation. Renewed interest in the subject in the 2000's led NASA's development of a stage separation conceptual separation tool, *ConSep* [116, 118–120]; which is a MATLAB-based wrapper to the commercially available ADAMS solver, as its predecessor *SepSim*. However, being *SepSim* and *ConSep* dependent on the commercial software ADAMS, they have the disadvantage of not being easily integrable in a generic trajectory simulation software. This in turn eludes the capability of performing efficient end-to-end launch vehicle trajectory simulations. As a result, a generalized approach to stage separation problems of launch vehicles was developed [1]. The approach, coined as the *Constraint Force Equation* (CFE) methodology, was implemented into the *Program to Optimize Simulated Trajectories II* (*POST2*), the *POST* follow-up. Separation studies applied to real platforms such as the Hyper-X or the Space Shuttle can be found in [153, 156]. The thesis [157] studies launcher separation analysis with OPENMODELICA but results in a tool (*OMSep*) which is only capable of input-output analyses *at* separation time, and not for generic launch vehicle trajectories.

As yet, an object-oriented and equation-based acausal modeling approach to stage separation dynamics integrable in end-to-end launch vehicle trajectory simulations is still missing. Such approach could potentially facilitate the integration of this and other capabilities within a single multi-physics environment such as DYMOLA.

The objective of this chapter is therefore to present such an alternate approach to stage separation dynamics based on the CFE methodology using MODELICA [37, 38]. We do this by means of the following sub-objectives: We study first the modeling challenges of multi-stage launcher separation dynamics; then we present an approach based on CFE implemented in MODELICA; following, we provide two case studies for which we apply the method; and finally we present some results and discussion, outlining benefits and disadvantages.

3.2. MODELING

For the simulation of launch vehicle stage separation dynamics, it is necessary being able to model two bodies connected together according to properly-selected constraints prior to their physical separation; and at the release command of such constraints, their

subsequent and independent flight motion must continue. This section presents the separation dynamics and the separation mechanisms modeling aspects.

3.2.1. SEPARATION DYNAMICS

We refer to separation dynamics in this chapter the study of the effects of forces and torques of a two-body system during their physical separation.

Such separation dynamics modeling clearly exhibits discontinuities similar to those described by other phenomena such as switching, limiting, friction, etc. Modeling must deal with these problems in special ways since this kind of behavior is sensitive to numerical solution errors, initial condition calculation/propagation, and integration in general.

MODELICA offers the possibility to implement a.o. several methods for such phenomena:

- *Stop and restart:* The complete system is simulated as a single body until separation time. Then the system is splitted into two bodies with independent states, and initial conditions are propagated accordingly. This solution however requires the split of two (or more) events.
- *Regularization:* This methodology consists on applying the constraint between the two bodies during their connected motion with a smooth but very stiff spring-damper system. This avoids the use of strict discrete or event behaviors. Such methodology is commonly used for simulation of friction, stiction, and other similar nonlinear behavior.
- *Hybrid:* This methodology consists on treating the simulation as a hybrid state machine where continuous and discontinuous behaviors are conditioned with data flows and proper transitions. This hybrid state machine framework is however complex to integrate in generic form for launch vehicle trajectory simulations.
- *Constraint Force Equation (CFE) Methodology:* The CFE methodology [1, 117, 153] consists on computing internal constraint forces and moments on two bodies during their connected motion and their application as external forces and torques to each of them separately. On separation command, these internal forces are set to zero, and then each body carries their own flight motion separately.

Of these methods, particular interest due to its applicability and easiness of implementation is given to the CFE methodology, which is selected as the primary method for the follow up of this study.

CONSTRAINT FORCE EQUATION METHODOLOGY

The *Constraint Force Equation* (CFE) methodology [1, 117, 153] is a highly intuitive method consisting in the computation of joint loads, namely internal forces and torques, caused by joint constraints; along with their application as external forces and torques on each body independently, see Figure 3.1.

The joint loads which constrain one body's motion relative to the other are dependent upon the external forces acting on each body as well as the type of joint. The

net forces and torques on each body are therefore the sum of the usual external forces and torques *plus* the joint loads applied to each body as additional external forces and torques. In consequence, the CFE joint model simply augments the external loads of the system [117]. Quoting step by step [1, 117], the equations of constrained motion of two

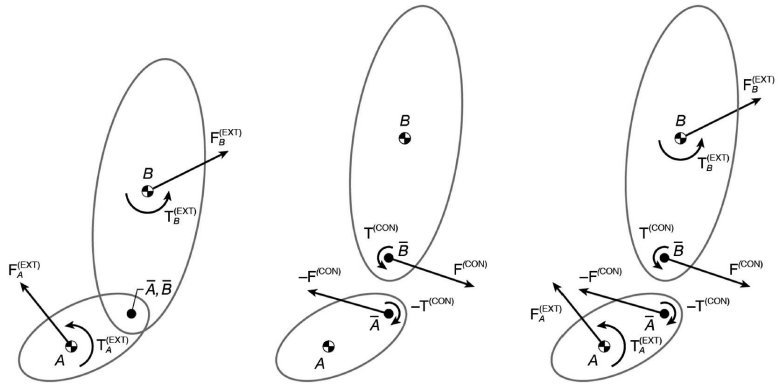


Figure 3.1: CFE diagram. Illustration credits: [1].

rigid bodies (A and B) connected by a single joint (point $\overline{A}$ in body A and point $\overline{B}$ in body B) are as follows:

$$\mathbf{F}_A^{(ext)} + \mathbf{F}_A^{(con)} = m_A \ddot{\mathbf{x}}_A, \tag{3.1}$$

$$\mathbf{T}_A^{(ext)} + \rho_A \mathbf{F}_A^{(con)} + \mathbf{T}_A^{(con)} = \boldsymbol{I}_A \dot{\boldsymbol{\omega}}_A + \boldsymbol{\omega}_A \times \boldsymbol{I}_A \boldsymbol{\omega}_A \tag{3.2}$$

where ρ_A is the position vector from the mass center of A to point $\overline{A}$ of A at which the constraint force is applied. Similarly for B:

$$\mathbf{F}_B^{(ext)} + \mathbf{F}_B^{(con)} = m_B \ddot{\mathbf{x}}_B, \tag{3.3}$$

$$\mathbf{T}_B^{(ext)} + \rho_B \mathbf{F}_B^{(con)} + \mathbf{T}_B^{(con)} = \boldsymbol{I}_B \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times \boldsymbol{I}_B \boldsymbol{\omega}_B. \tag{3.4}$$

There are so far 24 unknowns and 12 equations. Another set of six equations can be obtained from the law of action and reaction:

$$\mathbf{F}_A^{(con)} + \mathbf{F}_B^{(con)} = \mathbf{0} \tag{3.5}$$

$$\mathbf{T}_A^{(con)} + \mathbf{T}_B^{(con)} + (\mathbf{r}_B - \mathbf{r}_A) \times \mathbf{F}_B^{(con)} = \mathbf{0} \tag{3.6}$$

where $\mathbf{r}_A = \mathbf{x}_A + \rho_A$ and $\mathbf{r}_B = \mathbf{x}_B + \rho_B$.

Six equations are missing. Worth noticing at this point, we only consider a *single joint* which constrain all six remaining degrees of freedom between the two bodies. This is because our focus is towards trajectory simulations and having multiple connections is not necessary unless when considering actuator sizing, sensitivity analyses, etc. In general, the CFE methodology allows to consider any type of joint which allows or not any specific relative motion between bodies; and redundancy of joints when necessary.

In this sense, for relative translation constraints and **e** being unit-vectors of the corresponding (A or B) body-frame, it is required that:

$$(\mathbf{r}_B - \mathbf{r}_A) \cdot \mathbf{e}_A = \mathbf{0} \tag{3.7}$$

meaning that the distance between the two points of a particular direction remain fixed. And finally, for relative rotations constraints, it is required that:

$$\mathbf{e}_A \cdot \mathbf{e}_B = \mathbf{0} \tag{3.8}$$

meaning that three properly selected two-unit-vector sets must remain perpendicular.

Eqs. (3.7)-(3.8) would have to be differentiated twice with respect to time so that the resulting relationships involve the unknown accelerations and angular accelerations, thus finally being able to couple them with the equations of motion. In other words, the six missing equations are given by the following generalized constraint equations of the joint:

$$\ddot{\boldsymbol{g}} = \mathbf{0} \tag{3.9}$$

where g represents either of the nondifferentiated constraints in Eqs. (3.7) and (3.8). As it will be demonstrated in the next section, the manual differentiation of Eqs. (3.7)-(3.8) and their coupling with the equations of motion can be avoided altogether by the MODELICA implementation since this is done automatically.

The last important aspect of the CFE methodology relevant to this work is the accuracy of the joint loads solution, which is sensitive to computational error and initial joint missalignment [117]. To handle such concern, the CFE algorithm could feature a.o. a stabilization technique known as Baumgarte stabilization [1, 154, 155]. This particular stabilization technique consists on replacing the ODE given by Eq. (3.9) which allows perturbations to grow linearly with time, by the following asymptotically stable ODE ($\eta > 0$) involving terms of the once differentiated and nondifferentiated forms of g:

$$\ddot{\boldsymbol{g}} + 2\eta\dot{\boldsymbol{g}} + \eta^2 \boldsymbol{g} = \mathbf{0} \tag{3.10}$$

however at the expense of more computational effort. Many other stabilization techniques [155] could be implemented; these other methods, and a guidance for selecting η are however out of the scope of this chapter.

3.2.2. PHYSICAL MODELING OF MULTI-STAGE SEPARATION MECHANISMS

Separation mechanism refers in this proposal to a mechanical model (or device) that makes separation possible in simulation (or reality). Physical modeling refers in this context on the capability to model separation behaviour by considering first principles (kinematics, dynamics, mechanics, physics, etc.); and being able to get realistic insight from such models for other purposes such as actuator sizing, sensitivity analyses, control, optimization, etc.

Based on our internal *DLR Space Systems Library*, separation mechanism physical models of different complexity levels can be studied. Simplified models for preliminary and conceptual studies; and more detailed ones for engineering validation aspects. These varying degrees of complexity would be helpful in order to perform separation mechanics analyses and to assess the performance of the overall separation.

Configuration details of the separation mechanisms as well as their physical specifications must be provided to achieve more detailed and realistic models. Concerning the simple models, four variants have been studied:

- *Linear charge (release device):* The linear charge model performs ideal or benchmark separation between two bodies. This mechanism "cuts" the two-body system on command. It simulates (ideal) explosive release devices, clamps, diaphragms, or point-release devices such as explosive bolts.
- *Bushing (separation impulse device):* This model performs an impulsive reaction due to the release of a smooth but very stiff spring-damper system which keeps the two body system connected until separation command.
- *Kick-off spring (separation impulse device):* Same as before, the impulsive reaction due to the release of a spring-damper system simulates the proper transmission of forces and moments of the two-body system during separation. This model is implemented with the *Constraint Force Equation* (CFE) methodology. This element is combined with a release device to simulate a realistic kick-off spring mechanism.
- *Generic (auxiliary devices):* Other generic devices can be modeled in combination with the previous models, or with any other physical model from the library.

3.3. MODELICA IMPLEMENTATION

In this section, the MODELICA implementation of separation mechanism models is presented. The challenges of this implementation strongly depends on the method selected as outlined in Section 2. Since the separation models in this work relies on a proper combination of the CFE methodology with physically-relevant elements, the implementation is not a straightforward application of existing MODELICA libraries; other aspects such as proper setup of initial conditions, state selection, modularity, and extendability are also challenging.

The baseline for the development of separation dynamics and separation mechanisms is the following partial mechanism model:

```
partial model PartialMechanism
  "Partial separation mechanism model"
  Interfaces.Frame_a frame_a
    "Joint frame a";
  Interfaces.Frame_b frame_b
    "Joint frame b";
  Interfaces.BooleanInput u;
end PartialMechanism;
```

As shown in the code, the partial mechanism interface model consists of two frames to connect a two-body system, and a boolean input for the ignition or separation command. Such interface allows the use of several separation models depending on the desired level of complexity by using repleaceable instances. The approach here is bottom-up design, where the basis of separation dynamics simulation comes first from a single instance of a 'release device' mechanism.

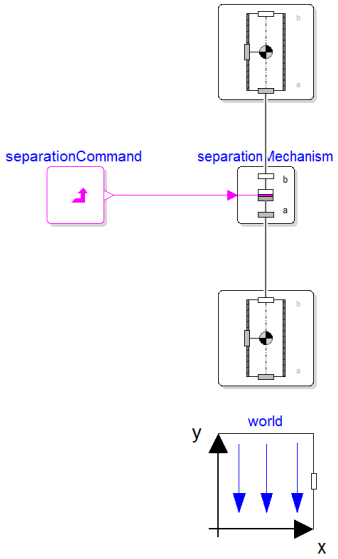


Figure 3.2: DYMOLA simulation layout consisting on a world model, two instances of rigid bodies, the separation mechanism model, and a boolean input for the separation command.

```
model SeparationMechanism
  "Separation mechanism model"
  Interfaces.Frame_a frame_a
    "Mechanism frame a";
  Interfaces.Frame_b frame_b
    "Mechanism frame b";
  Interfaces.BooleanInput u;
  replaceable Interfaces.PartialMechanism
end SeparationMechanism;
```

In this work, a release mechanism model is implemented to simulate both a linear charge device commonly used in launcher stage separation, where the forces and moments at separation are zero; and as a base model for the next level of complexity. In other words, for the implementation of a separation impulsive device, an instance of a release device providing the capabilities of joint motion until separation is required on top of another physical model providing the corresponding impulsive forces or moments at the time of separation. Therefore, increasing the functionality to the separation model will consist on adding impulsive devices or simply improving the physics behind the device in question.

The implementation of the CFE procedure in MODELICA is as follows. The generalized constraint equations of the joint (3.9) have to be differentiated twice as explained before. Translational and rotational constraints at the joint are hence implemented as:

```
equation
  // generalized constraints
  g_con = frame_a.r_0 - frame_b.r_0;
  G_con = Frames.relativeRotation(frame_a.R,frame_b.R);

  // generalized velocity constraints
  g_con_dot = der(g_con);
  G_con_dot = Frames.angularVelocity2(G_con);

  // generalized acceleration constraints
  g_con_ddot = der(g_con_dot);
  G_con_ddot = der(G_con_dot);

  // CFE generalized joint constraints
  g_con_ddot = {0,0,0};
  G_con_ddot = {0,0,0};
```



```
equation
...
  // CFE generalized joint constraints with Baumgarte stabilization
    g_con_ddot + 2*eta*g_con_dot + eta*eta*g_con = {0,0,0};
    G_con_ddot + 2*eta*G_con_dot + Frames.Orientation.equalityConstraint(frame_a.R,
     frame_b.R) = {0,0,0};
```

In short, we present briefly two of the main models developed in this work:

- *Linear charge (separation release device):* A release device is modeled by an instance of the *SeparationMechanism* model, called for instance **linearCharge**, which contains the partial interface outlined before, plus a switching mechanism between the CFE methodology and free body motion.

- *Kick-off spring (separation impulse device):* An impulsive device is modeled by an instance of the *SeparationMechanism* model, called for instance **kickOffSpring**, which contains a **linearCharge** instance, plus a replaceable **separationMechanism** instance simulating the physics behind the impulsive device, such as a spring-damper system.

For a practical scenario to study, consider the trajectory phase of a generic launcher where the payload (Body B - the satellite to be placed in orbit) is to be separated from the remaining launcher upper stage (Body A - assuming a multi stage launcher). In this case, the problem consists of two bodies flying together under the effect of gravity in joint motion (the composite) up until separation is commanded. The separation command is usually given immediately after the shut down of the upper stage main engine. In this study however, we provide the separation command at any specified time. Figure 3.2 shows the DYMOLA simulation layout while Figure 3.3 shows a simulation of the physical setup of the case studies.

Initial conditions with respect to Earth-Centered-Inertial (ECI) frame of the compos-

ite are given to Body A as follows:

$$\mathbf{x}_A(t=0) = \begin{bmatrix} 1.1378 \times 10^7 \\ 0 \\ 0 \end{bmatrix} \text{m},$$

$$\mathbf{v}_A(t=0) = \begin{bmatrix} 0 \\ 5.9188 \times 10^3 \\ 0 \end{bmatrix} \text{m/s}$$

and their translational and rotational dynamics are obtained from the rigid body model of the *Modelica Multibody Library* [107]. In the following section, we will study the separation dynamics implementation in MODELICA by means of two case studies: the first one considers the upper stage and payload (the composite) joint motion, while the second study considers the separation phase. For both cases, the forces due to gravity acceleration are obtained from the EGM96 model implemented in our internal *DLR Space Systems Library*.

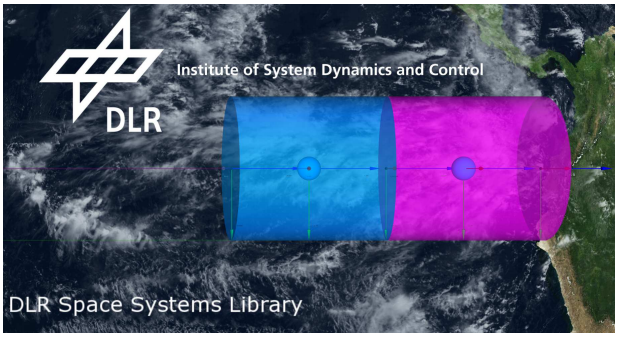


Figure 3.3: Simulation of the physical setup of the case studies.

Table 3.1: Mechanical properties of the two-body system.

Property	Body A	Body B	Units
Mass	6000	1000	Kg
I_{11}	23000	800	Kg·m^2
I_{22}	23000	800	Kg·m^2
I_{33}	18000	600	Kg·m^2
$I_{21} = I_{31} = I_{32}$	0	0	Kg·m^2

Both case studies are implemented in DYMOLA and the solution is computed using the DASSL solver with a tolerance of $1e-7$. A smaller tolerance of this solver would increase significantly the resulting chattering when Baumgarte stabilization is used.

3.3.1. CASE STUDY I: UPPER STAGE AND PAYLOAD (COMPOSITE) JOINT MOTION

The joint motion of the composite (Bodies A and B, the upper stage and the payload respectively) is simulated for a total time of 2000 s. During such motion, the MODELICA implementation of the CFE methodology is expected to derive automatically the joint constraint forces and torques such that the two-body system stays properly connected, with relative zero displacement. This case study therefore accounts for the validity of such implementation.

3.3.2. CASE STUDY II: UPPER STAGE PAYLOAD SEPARATION DYNAMICS

The upper stage payload separation is simulated in a practical scenario setup. It consists of a simulation of 20 s, half of which is in connected or joint motion, and then at $t = 10$ s, the ignition command for separation is given. At this point, a kick-off spring separation mechanism model is in charge of the dynamical separation between the bodies. The subsequent independent motion of each body is then expected. This case study therefore accounts for the applicability of the physical models of separation mechanisms implemented.

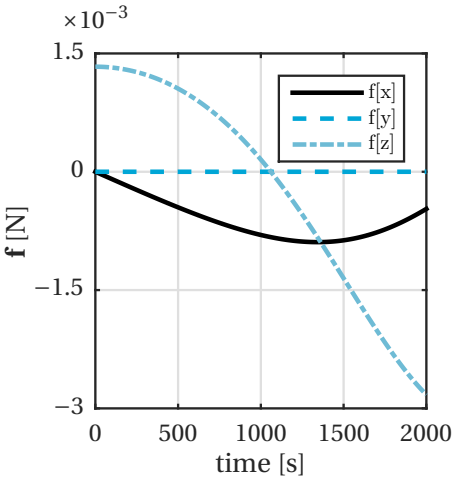
3.4. RESULTS AND DISCUSSION

As outlined in the last section, **Case Study I** accounts for the study of internal forces and torques of the composite joint motion during a given portion of its trajectory by means of the *Constraint Force Methodology* implemented in MODELICA. During such joint motion, an important metric to assess the proposed method is the relative joint displacement between the two bodies when they are supposed to stay connected, as proposed and suggested by [117].

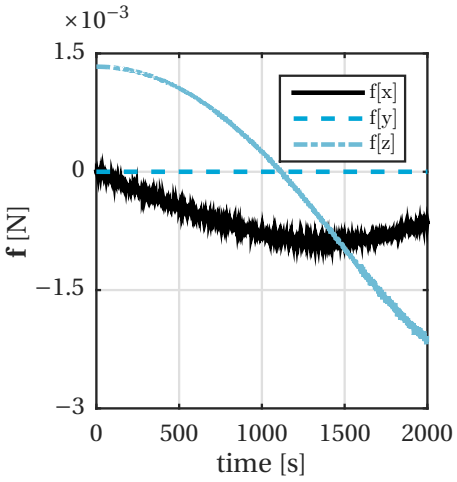
In this respect, Figure 3.4 presents the resulting constraint forces $\mathbf{f}[i]$ and torques $\mathbf{tau}[i]$ at the joint during the connected motion, in all ECI directions $i = x, y, z$, respectively; while Figure 3.5 presents the resulting *relative* joint position $\mathbf{rrel}[i]$ and the *relative* joint velocity $\mathbf{vrel}[i]$, in all ECI directions $i = x, y, z$, respectively.

Results shows that the corresponding joint constraint forces and torques, obtained automatically by MODELICA in order to satisfy the CFE methodology constraints successfully keeps the bodies properly connected (hence, the composite) during their connected flight motion. Such result is evidenced looking at the relative joint position and relative joint velocity between the two bodies, which are supposed to be zero during the connected flight. A clear disadvantage for long simulation periods of joint composite motion is the necessity to keep the drift within physical boundaries, hence requiring a stabilization method. Stability and accuracy of the solution, especially for large simulation times, are improved with the addition of the *Baumgarte stabilization*. Nevertheless at the expense of chattering as shown in Figures 3.4-(b), 3.4-(d), 3.5-(b), 3.5-(d), meaning more computational time and effort.

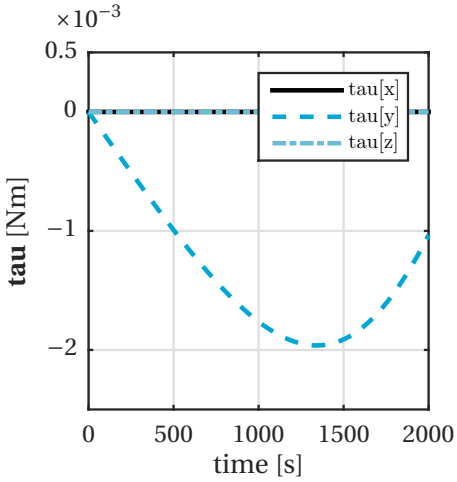
Case Study II, as outlined in the last section, accounts for the study of absolute− and relative− position, velocity, and acceleration, respectively, between the two bodies from a multi-stage separation dynamics practical scenario. In here, the 'release device' simulated by a linear charge model has been augmented with an 'impulsive device' in parallel



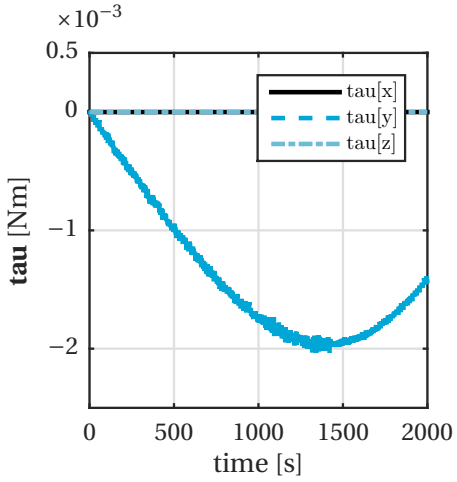
(a) Constraint forces at joint during connected motion with CFE methodology, in all ECI directions $i = x, y, z$.



(b) Constraint forces at joint during connected motion with CFE methodology plus Baumgarte stabilization with $\eta = 2$, in all ECI directions $i = x, y, z$.



(c) Constraint torques at joint during connected motion with CFE methodology, in all ECI directions $i = x, y, z$.

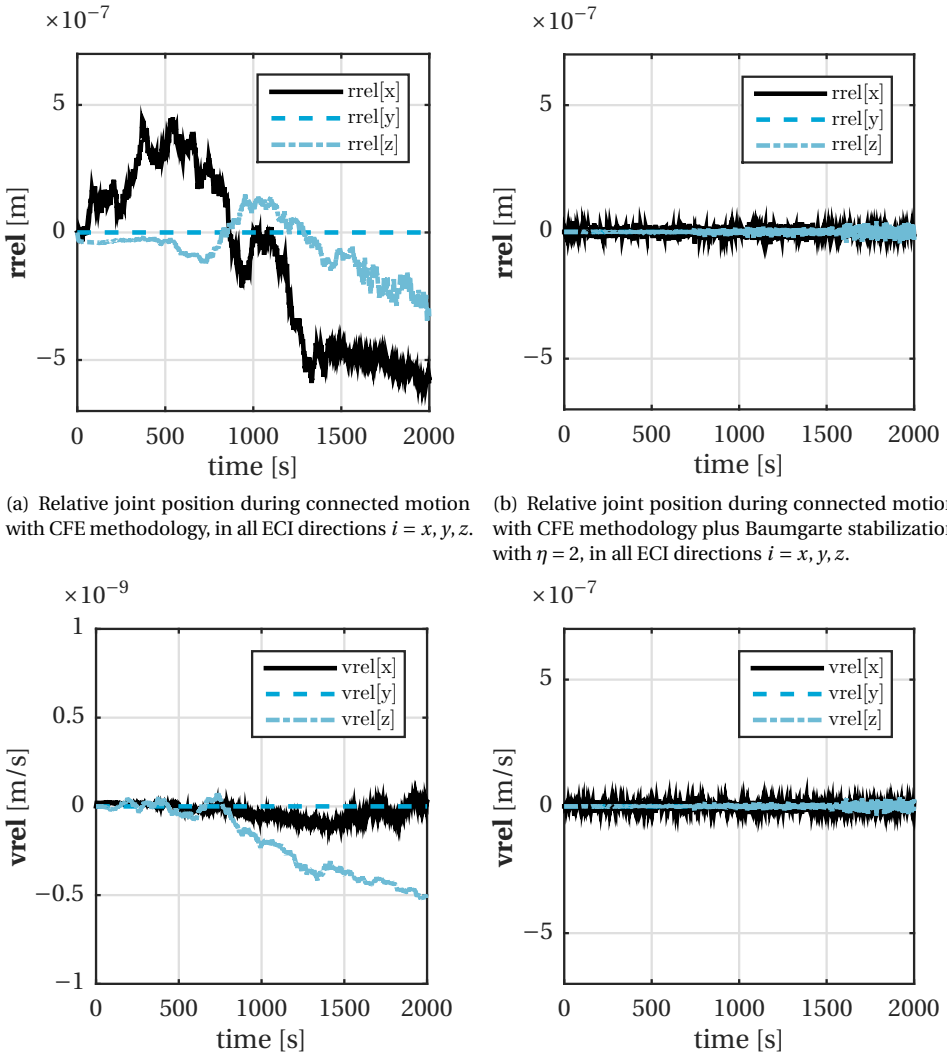


(d) Constraint torques at joint during connected motion with CFE methodology plus Baumgarte stabilization with $\eta = 2$, in all ECI directions $i = x, y, z$.

Figure 3.4: Case Study A results: constraint forces and torques at joint during connected motion.

simulated by a kick-off spring model in order to simulate such a separation mechanism between the two bodies at their time of release from each other.

In this respect, Figure 3.6 presents the bodies' *relative* position **rrel**$[i]$, velocity **vrel**$[i]$, and acceleration **arel**$[i]$ along the ECI orbital flight direction $i = y$ (which is valid only for such a very small time frame) during the connected motion (first 10 seconds), and during their subsequent separation (last 10 seconds). Figure 3.6 also presents a zoom of



(a) Relative joint position during connected motion with CFE methodology, in all ECI directions $i = x, y, z$.

(b) Relative joint position during connected motion with CFE methodology plus Baumgarte stabilization with $\eta = 2$, in all ECI directions $i = x, y, z$.

(c) Relative joint velocity during connected motion with CFE methodology, in all ECI directions $i = x, y, z$.

(d) Relative joint velocity during connected motion with CFE methodology plus Baumgarte stabilization with $\eta = 2$, in all ECI directions $i = x, y, z$.

Figure 3.5: Case Study A results: relative joint position and velocity during connected motion, in all ECI directions $i = x, y, z$.

the small time window just around the separation command.

Results of this separation scenario shows the corresponding relative states of the composite up until separation command and then their subsequent independent flight. Once again, the benefit and ease of use of the MODELICA implementation of the CFE methodology is evidenced during the connected flight of the composite, since constraint forces and torques are automatically computed and applied to the system. At separation,

the relative states suggest an impulsive behaviour due to the kick-off spring separation mechanism model. This model releases a pre-compressed force stored in a replaceable spring-damper model, evidencing good correspondence with the physics behind separation. Such devices result in impulsive forces applied to the two-body system. This in turn causes a change in relative velocity and therefore, a successful physical separation of the system.

3.5. CONCLUSIONS

The objective of this chapter was to present an object-oriented and equation-based acausal modeling approach to launch vehicle stage separation dynamics with MODELICA. The aim is to develop an integrated approach for end-to-end launch vehicle trajectory simulation within a single environment.

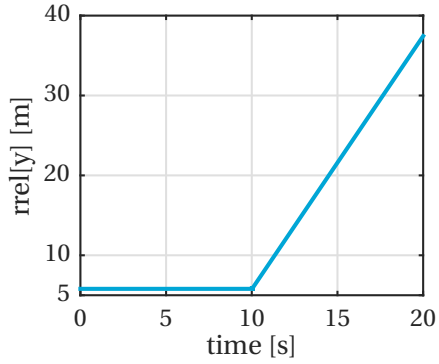
Based on the *Constraint Force Equation* (CFE) methodology, two case studies to evaluate the proposed approach were considered. The scenario under study consisted of two bodies –representing a generic launcher stage and its payload– prior, during, and after their separation in orbital flight motion.

Results demonstrated that the approach, mainly thanks to the acausal and equation-based modeling features of the MODELICA language, corresponds very well with the physics behind separation while providing easiness of implementation within a single environment such as DYMOLA. The method computes and applies constraint loads automatically during joint motion and removes them accordingly at separation time, all in consistency with the CFE methodology.

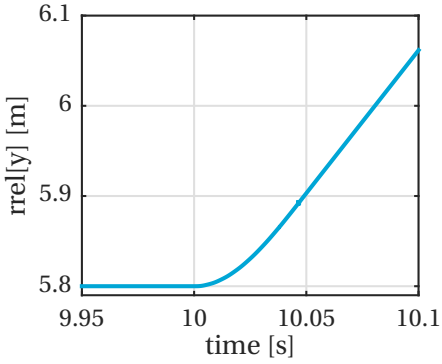
A disadvantage for long simulation periods of joint body motion is the necessity to keep the drift within physical boundaries, hence requiring a stabilization method. This in turn increases chattering and computational time and effort, thus resulting in a trade-off to consider for the task at hand. Validation studies are left to future work.

ACKNOWLEDGEMENTS

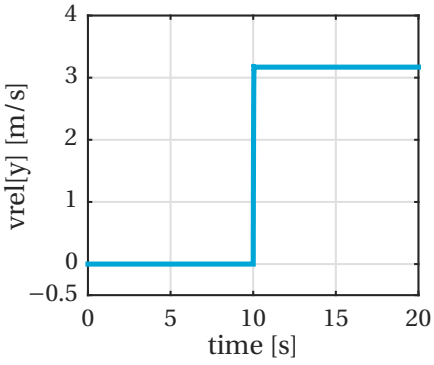
This work corresponds to DLR Institute of System Dynamics and Control activities of the ESA study **Upper Stage Attitude Control Design Framework** (USACDF), led by Astrium GmbH as part of Europe's Future Launchers Preparatory Program (FLPP).



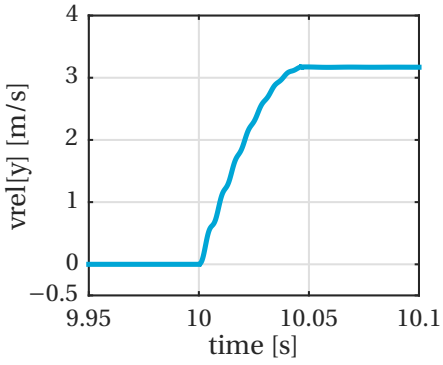
(a) Relative position between bodies A and B. The initial relative position (5.8 m) corresponds to the fixed distance between the bodies center of masses during joint motion.



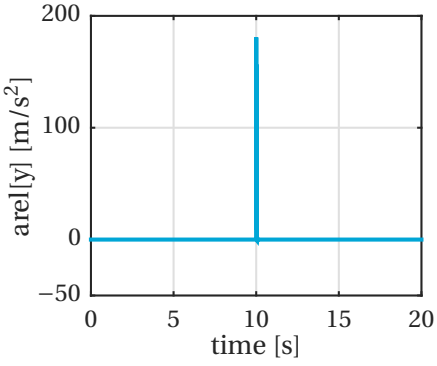
(b) Same as (a) with a close view around time of separation.



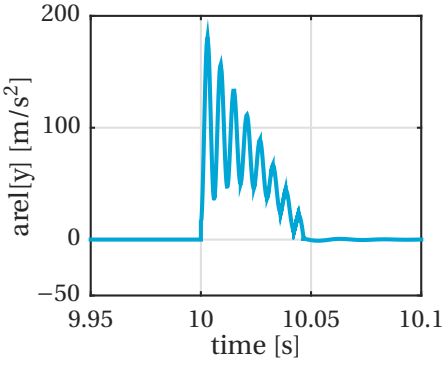
(c) Relative velocity between bodies A and B.



(d) Same as (c) with a close view around time of separation.



(e) Relative acceleration between bodies A and B.



(f) Same as (e) with a close view around time of separation.

Figure 3.6: Case Study B results: Relative position, velocity, and acceleration from a kick-off separation scenario along orbital flight direction $i = 2$. Ignition / separation command at $t = 10$ s.

II

Aerospace Guidance and Control (G&C)

4

Guidance Command Generation and Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles

Abstract

Future launch vehicle concepts and technologies for expendable and reusable launch vehicles are currently investigated by the DLR research projects AKIRA *and* X-TRAS*. In particular, the winged Liquid Fly-back Booster concept* LFBB *based on an LOX/LH2 propellant combination for vertical takeoff and vertical landing (VTVL), as well as the delta-winged horizontal takeoff and horizontal landing (HTHL) concept* AURORA *based on an LOX/Kerosene propellant combination are considered in these projects. Because of the complexity and risks involved in on-line trajectory optimization, off-line reference trajectories are still considered important for tracking purposes. In that sense, the goal of this paper is to investigate an off-line and general-purpose guidance and control (G&C) architecture for preliminary studies of reusable launch vehicles. This is done by using trajectory optimization combined with* MODELICA *models for the generation of optimal guidance commands, and then trajectory tracking is performed by means of inner-loop feedback controls in terms of nonlinear dynamic inversion with prescribed desired dynamics. We showcase the advantages of this baseline G&C architecture in terms of early stability and controllability aspects during the preliminary design studies of an example configuration of a reusable launch vehicle investigated in the context of the research projects above mentioned.*

Publication

Paul Acquatella B., Lâle E. Briese, and Klaus Schnepper: *Guidance Command Generation and Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles.* In: **Acta Astronautica**, Vol. 174, pp. 334–346 (2020); presented at *IAC 2018, 69th International Astronautical Congress*, October 1-5, 2018. Bremen, Germany.

4.1. INTRODUCTION

SEVERAL studies on future launch vehicle configurations and technologies for expendable and reusable launch vehicles have been extensively conducted in the past at DLR [8, 10–16]. Currently, partly or fully reusable launch vehicles using different return methods are investigated at DLR in the context of the research projects AKIRA and X-TRAS [5, 17, 158].

Reusability of launch vehicles strongly impacts the launch servicing market whenever sufficient reliability and low refurbishment costs can be achieved. Thus, keeping up with such rapidly evolving international launcher market is essential for Europe, and therefore the need for continuous investigation of different methods and technologies for reusability [8–10].

In particular, the winged *Liquid Fly-back Booster* concept LFBB, studied extensively during the early 2000's [24] and more recently in [17], based on an LOX/LH2 propellant combination for vertical takeoff and vertical landing (VTVL), as well as the more recent study of the delta-winged horizontal takeoff and horizontal landing (HTHL) concept AURORA [5, 158] based on an LOX/Kerosene propellant combination have been considered.

For the launcher concepts and configurations to consider and optimize at preliminary design studies, early stability and controllability aspects are necessary. This leads to the following motivation for this chapter.

4.1.1. MOTIVATION

This chapter focuses on early stability and controllability aspects during the preliminary design studies of launcher conceptual designs. Identifying the impact of such aspects on performance, reaction control system (RCS) design, and actuator sizing (RCS, aerodynamic control surfaces, thrust vector control), among many others, is of great importance. In particular, for each reusable launcher design study we ask ourselves these questions:

- *What is the optimal reference trajectory according to the mission constraints and requirements?*
- *Is this configuration controllable?*
- *What is the impact of the controllability on the design (impulse budget, reaction control system sizing, aerodynamic control surfaces, etc.)?*

Because of the complexity and risks involved in on-line trajectory optimization, off-line reference trajectories are still considered important for tracking purposes. In that sense, to answer these questions, we focus on a guidance and control (G&C) architecture by using an optimal trajectory generator to find an off-line reference trajectory, and then trajectory tracking is performed by means of inner-loop feedback controls using Nonlinear Dynamic Inversion (NDI) and linear control (LC).

4.1.2. PREVIOUS WORK

Nonlinear Dynamic Inversion, based on *feedback linearization* [61–63], is very common in the aerospace field; some applications of flight control include [60, 67–70]. More ad-

vanced methods involving robustness and improvements of the method in NDI-based flight control applications are considered, among many others, in [69–75].

NDI methods also found their application for the control of spacecraft and re-entry vehicles, see for example [4, 49–51] and the references therein. Early works on NDI for space applications include [49], where a nonlinear flight control system for a winged re-entry vehicle was designed that accurately tracks attitude commands while being subject to significant aerodynamic uncertainties, and [50], where a general purpose two-loop flight control architecture for attitude control was designed based on time-scale separation for a lifting body re-entry vehicle using nonlinear dynamic inversion.

The work here presented is largely based on these last references [4, 49–51], however more oriented towards an integrated approach as in [60] combining trajectory optimization, nonlinear models implemented in the acausal modeling language MODELICA, and NDI control; which leads to the following objectives.

4.1.3. OBJECTIVES

The goal of this chapter is therefore to investigate a baseline and general-purpose G&C architecture for reusable launch vehicles involving the combination of trajectory optimization and MODELICA models for nonlinear control. We do this by combining the following three separate methods:

1. **Trajectory Optimization**. An off-line reference trajectory can be generated by transcribing the trajectory optimization problem into a multi-criteria optimization problem. Solutions are found with a direct approach using the trajectory optimization package '*trajOpt*' of DLR-SR's optimization tool MOPS (*Multi-Objective Parameter Synthesis*).

2. **Guidance Command Generation**. Guidance commands are generated via combination of *trajOpt* with nonlinear models implemented with the object-oriented, equation-based, multi-physical, and acausal modeling language MODELICA. These consists on the optimal flight path reference and its corresponding commands (aerodynamic angles) for the inner-loop attitude control.

3. **Nonlinear Dynamic Inversion Control**. Lastly, inner-loop attitude control is based on nonlinear dynamic inversion (NDI). NDI cancels out nonlinearities in the system via state feedback, and then desired dynamics can be prescribed to track the optimal reference trajectory accordingly. The nominal performance is therefore considered as a benchmark for the controllability analysis of the launch vehicle along the reference trajectory.

To demonstrate the feasibility of using this integrated approach, we showcase the advantages of this baseline G&C architecture in terms of early stability and controllability aspects during the preliminary design studies of an example configuration of a reusable launch vehicle.

The remainder of the chapter is organized as follows. Section 4.2 presents trajectory optimization problem formulation and its solution. In Sections 4.3 the optimal guidance commands that are obtained with the trajectory optimization in combination with MODELICA models is explained. Section 4.4 briefly explains the control design method

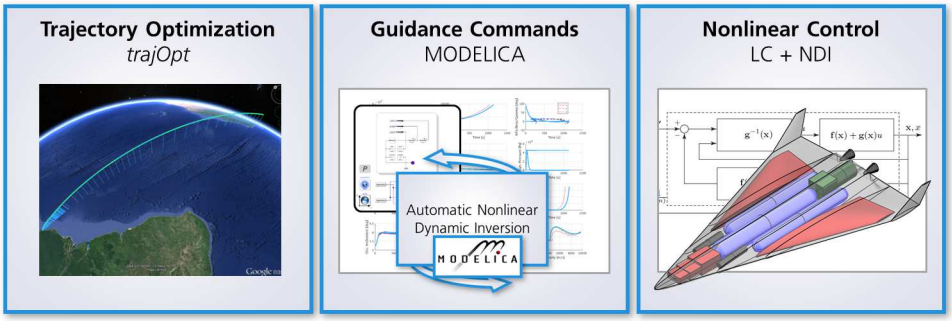


Figure 4.1: Workflow of the proposed G&C design architecture.

behind the nominal trajectory tracking, and Section 4.5 presents the simulations of these controllers in a particular reusable launcher configuration. Conclusions are discussed in Section 4.6.

4.2. OPTIMAL TRAJECTORY GENERATION

In this section, the general trajectory optimization problem that can be treated with the *Trajectory Optimization Package 'trajOpt'* [2] is specified. Following that, the transcription to a problem handled by MOPS *(Multi-Objective Parameter Synthesis)* [54] is shown. MOPS solves the transcribed multi-objective design problems by mapping them to weighted *min-max* optimization problems. The *trajOpt* structure and its classes supports this transcription process by its implementation as an object-oriented MATLAB [36] package within MOPS.

4.2.1. OPTIMAL TRAJECTORY OPTIMIZATION PROBLEM FORMULATION

The description of trajectory optimization problems follows the notation used in MOPS. In particular constraints and optimization criteria are defined by just one category of functions: MOPS criteria. Mathematically the trajectory optimization problems covered can be described as:

Given m phases with possibly optimizable phase times

$$t^j \in \{ t^0 < t^1 < \ldots < t^m \} \tag{4.1}$$

the states $x^j(t)$ for each phase j obey initial value problems of the form:

$$\dot{x}^j = f^j(t, x^j, u^j, p^j), \quad x^j(t^{j-1}) = s^j, \quad j \in 1 \ldots m. \tag{4.2}$$

Here $u^j(t)$ are (optimizable) control functions in phase j and p^j are constant scalar modeling parameters (design parameters). The differential equations for each of the multiple phases can differ completely. A well known example is the ascent optimization for multistage rockets, where each stage configuration defines a phase of the problem.

For each phase there can be criteria specified at the phase's final time (right side)

$$\begin{aligned} &\min {}_r\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) && j, k_j \in S_m \\ &{}_r\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) \le 1 && j, k_j \in S_i \\ &{}_r\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) = 1 && j, k_j \in S_e \\ &\text{for all} \quad \begin{cases} j \in 1 \dots m \\ k_j \in 1 \dots n^r_j \end{cases} \end{aligned} \tag{4.3}$$

and the phase's initial time (left side)

$$\begin{aligned} &\min {}_l\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) && j, k_j \in S_m \\ &{}_l\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) \le 1 && j, k_j \in S_i \\ &{}_l\Psi^j_{k_j}(t^j, x^j(t^j), u^j(t^j), p^j) = 1 && j, k_j \in S_e \\ &\text{for all} \quad \begin{cases} j \in 1 \dots m \\ k_j \in 1 \dots n^l_j \end{cases} \end{aligned} \tag{4.4}$$

4

In this notation, for phase j, we have n^r_j final and/or n^l_j initial criteria. The initial values for state differential equations in phase j are $x^j(t^{j-1}) = s^j$. Optionally there can be additional path criteria evaluated at specified discrete times in the phase

$$\begin{aligned} &\min g^j_{o_j}(t_{k_j}, x^j(t_{k_j}), u^j(t_{k_j}), p^j) && j, k_j \in S_m \\ &g^j_{o_j}(t_{k_j}, x^j(t_{k_j}), u^j(t_{k_j}), p^j) \le 1 && j, k_j \in S_i \\ &g^j_{o_j}(t_{k_j}, x^j(t_{k_j}), u^j(t_{k_j}), p^j) = 1 && j, k_j \in S_e \\ &\text{for all} \quad \begin{cases} j \in 1 \dots m \\ o_j \in 1 \dots n_j \\ t_{k_j} \in [t_{j-1}, t_j] \end{cases} \end{aligned} \tag{4.5}$$

and phase connect constraints of the form

$$\begin{bmatrix} x^{j+1}(t^j) \\ u^{j+1}(t^j) \\ p^{j+1} \end{bmatrix} = h^{j+1}(t^j, x^j(t^j), u^j(t^j), p^j) \tag{4.6}$$

$$\text{for all} \quad j \in 1, \dots, m-1.$$

Here, S_m denotes the set of criteria to be minimized, and S_e and S_i are the sets of equality and inequality criteria from Equations (4.3), (4.4), and (4.5), and the equality criteria defined by Equation (4.6).

A graphical representation of this general problem is shown in Figure 4.2 including the control approximation and path criteria formulation.

4.2.2. TRANSCRIPTION INTO A DIRECT APPROACH

The trajectory optimization problem as posed in the previous section is an optimal control problem in function space for the control functions u^j. In order to solve trajectory

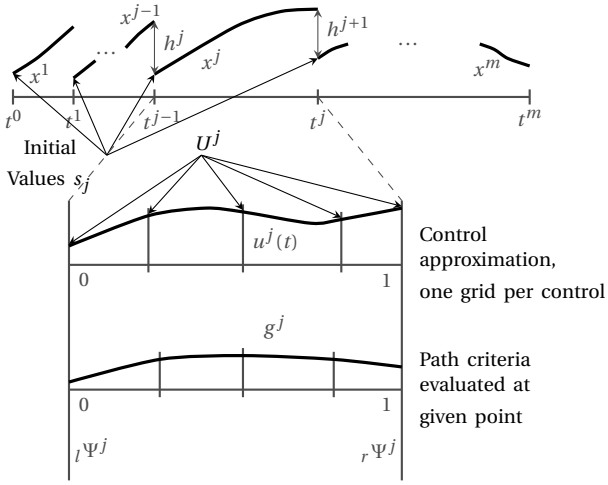


Figure 4.2: Multi-phase trajectory optimization problem with control discretization.

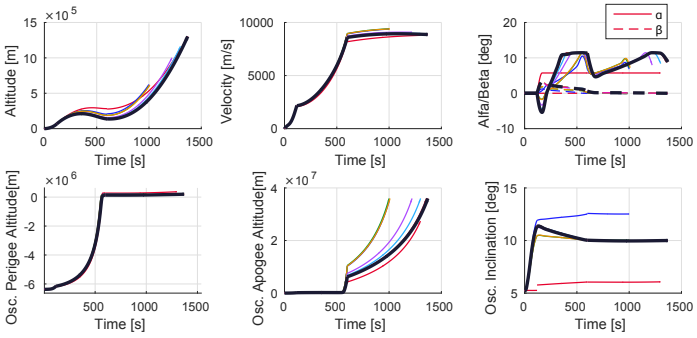


Figure 4.3: *trajOpt* optimization progress example for a classical expendable launch vehicle [2].

optimization problems from Equations (4.1) to (4.6) the control functions are discretized by approximation functions $u^j(t) = u^j(U^j, t)$, like piecewise polynomial functions with discretization parameters $U^j, j \in 1,\ldots,m$. These discretization parameters are added to the initial values s_j for the state equations, modeling parameters p^j, and the phase times t^j to form the optimization parameters (and tuners) of the rewritten optimization problem.

This transcription of the original trajectory optimization problem results in defining k design objectives as positive criteria c_k to be minimized against demanded values d_k by considering the following *min-max* constrained multi-criteria optimization problem (see MOPS [54])

$$\min_{\mathcal{T}} \left\{ \max_{k \in S_m} \left\{ \frac{c_k(\mathcal{T})}{d_k} \right\} \right\}, \tag{4.7a}$$

$$\text{subject to } c_k(\mathcal{T}) = d_k, \quad k \in S_e,$$
$$c_k(\mathcal{T}) \leq d_k, \quad k \in S_i,$$

with:

$$\mathcal{T}_{min} \leq \mathcal{T} \leq \mathcal{T}_{max}. \tag{4.7b}$$

Here, $\mathcal{T}$ is a vector containing the tuning parameters to be optimized, which is constrained by upper and lower bounds $\mathcal{T}_{min}$ and $\mathcal{T}_{max}$. $c_k \in S_m$ is the $k-$th normalized criterion and d_k its corresponding demand value which serves as a criterion weight; lastly, $c_k \in S_e, S_i$ are normalized criteria which are used as equality and inequality constraints. This multi-criteria optimization problem can then be solved using standard nonlinear programming (NLP) methods contained in MOPS [54].

As already mentioned, to support this transcription process, MOPS was augmented by the object-oriented MATLAB package *trajOpt* [2]. *trajOpt* defines base classes for specifying the ODE right-hand sides from Equation (4.2) and the criteria functions from Equations (4.3) to (4.6). These base classes handle much of the detail of criteria definition and evaluation handling within MOPS. A user needs to derive classes from these base classes for specifying only the actual criteria functions. This can be particularly easy when using Funtional Mockup Units (FMUs) as models for the ODE and criteria functions where this can reduce to a purely declarative process.

In addition *trajOpt* defines classes for handling the simulation of the actual model within the different phases and the correct evaluation of criteria functions. In particular, classes exist that hide the intricacies of using FMUs as models within the trajectory optimization framework. Additionally, using different FMU units in different phases is supported along with the ability to use MOPS and *trajOpt* in MATLAB parallel computation environments.

For a classical expendable launch vehicle Figure 4.3 shows the optimization progress when solving such a trajectory optimization problem.

4.3. GUIDANCE COMMAND GENERATION

In this section we focus on a nominal off-line guidance method to generate an optimal reference trajectory which keeps the launch vehicle's mission and physical constraints within its optimal values. These guidance reference commands are generated via combination of the trajectory optimization package *trajOpt* with nonlinear models implemented with the object-oriented, equation-based, multi-physical, and acausal modeling language MODELICA, which is briefly introduced in the next subsection. The MODELICA models used in this study regarding trajectory optimization (3-DOF) and trajectory tracking with nonlinear control (6-DOF), together with their implementation using an advanced launch vehicle modeling framework are presented in more detail in [3, 159].

4.3.1. MODELICA

MODELICA [37–41] is a modern object-oriented, equation-based *modeling language* well suited to model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, power or process–oriented subsystems and components.

Models in MODELICA are described using differential, algebraic, and discrete equations which are then mapped into a mathematical description form called hybrid *Differential Algebraic Equations* (DAEs). A DAE system on its implicit form is generally expressed as

$$\mathbf{F}\big(\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{y}(t), \boldsymbol{\rho}, t\big) = \mathbf{0}, \tag{4.8}$$

where $\dot{\boldsymbol{x}}$ are the state derivatives, $\boldsymbol{x}$ the state variables, $\boldsymbol{u}$ the inputs, $\boldsymbol{y}$ the algebraic variables, $\boldsymbol{\rho}$ the parameters and constants, t the time variable, and the dimension $dim(\mathbf{F}) = dim(\boldsymbol{x}) + dim(\boldsymbol{y})$. Systems are then solved and simulated by MODELICA simulation environments. When these systems are represented in the DAE implicit form, they can be solved directly by a DAE solver such as DASSL. Alternatively, the system can be sorted out according to specific inputs and outputs and mapped into an explicit ODE (Ordinary Differential Equation) form by solving for the derivatives and the algebraic variables, and then subsequently solved numerically by an ODE solver. The process and details of MODELICA's code compilation is out of the scope of this chapter.

MAIN FEATURES [41]

In contrast to imperative languages, in which statements and algorithms are assigned in explicit steps, MODELICA is *declarative*, meaning that declarations are given through equations. These declarations most often describe model's first-principles at their lowest levels without explicit orders or *how* to compute them, hence why MODELICA is said to be *equation based*. By means of specialized algorithms, these declarative models are translated into efficient computer executable code. This allows *acausal* modeling capabilities that give better reuse of classes since equations do not specify a certain data flow direction. This is therefore one of the most important features of the language.

MODELICA is *domain neutral*. In other words, it has *multi-domain* modeling capability, meaning that model components corresponding to physical objects from several different domains can be described and connected. This interaction between components is defined by means of physical ports, called connectors, and the interconnection is given accordingly to their physical meaning. This meaning is typically represented by flow variables, which describe quantities whose values add up to zero in a node connection (Kirchhoff's first rule); and by non-flow (or potential) variables, which in contrast remain equal (Kirchhoff's second rule).

MODELICA is an *object-oriented* language. This helps to model systems and their physical meaning within an object-oriented structure, facilitating the reuse of component models and the evolution of the structure itself. Thus, object-orientation is primarily used as a *structuring* concept which exploits the declarative feature of the language, as well as the re-usability of models.

MODELICA has a strong software component model with constructs for creating and connecting components in a *modular* fashion. Systems' individual components are defined separately as objects, and their interconnection is given accordingly to their physical meaning. Thus the language is ideally suited as an *architectural* description language for complex physical systems.

4.3.2. FLIGHT PATH GUIDANCE

Position and flight path control loops are readily obtained from the trajectory optimization package *trajOpt* depending on problem-specific optimization goals, requirements, and constraints. These can be such as maximizing the payload to a desired orbit and maximizing downrange for the descent vehicle while minimizing accelerations and dynamic pressure, and thus mechanical and thermal loads, for instance.

In this sense, the reference trajectory in terms of position and flight path provides the guidance commands that have to be tracked by the attitude control subsystem, which in turn commands the launch vehicle in terms of moments that are actuated by the aerodynamic surface deflections, the thrust vector control (TVC), or by the reaction control system (RCS) thrusters, depending on the configuration and the phase considered.

To that end, MODELICA-based 3-DOF launch vehicle models with phase-dependent configuration parameters are exported as *Functional Mock-up Units* benefiting from MODELICA's object-oriented structure. Subsequently, these FMUs are imported separately for each phase into *trajOpt*. Depending on the chosen configuration and flight phase of the launch vehicle, multiple control input variables like the aerodynamic angle of attack α, the aerodynamic sideslip angle β, the aerodynamic bank angle μ, as well as a throttle factor c_s can be active during the trajectory optimization. As a result, the optimal reference trajectory can be obtained for several quantities such as positions, velocities, transformation matrices, forces, or even some corresponding atmospheric parameters. This approach, as well as the launch vehicle modeling framework as shown in Figure 4.4, are explained in more detail in [3].

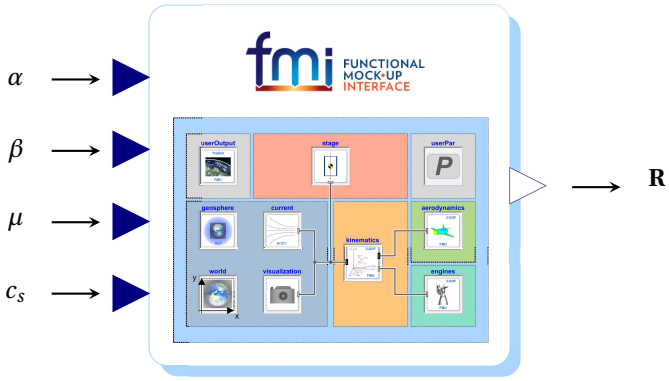


Figure 4.4: Input-Output Structure of an FMU containing the Launch Vehicle Modeling Framework [3].

The guidance command generation for this loop in this sense consists on the resulting flight path reference commands that are given in terms of the reference flight path parameters

$$\begin{bmatrix} V \\ \gamma \\ \chi \end{bmatrix}_{\text{ref}}$$

and its corresponding control commands for the aerodynamic angles

$$\begin{bmatrix} \mu \\ \alpha \\ \beta \end{bmatrix}_{\text{cmd}}$$

respectively. The additional throttle factor c_s is taken from the reference trajectory and used as a feedforward command in the attitude control.

One major advantage of the trajectory optimization and guidance command generation approach as discussed in detail in [3] is, that by considering multi-phase trajectory optimization, the computation of each trajectory phase with their respective objectives and constraints can be parallelized. This is useful when the ascent and upper stage phases have different objectives in contrast to the descent phase, although the overall trajectory must fulfill the overall mission objectives. This allows the rapid prototyping and analyses of different concepts and mission profiles.

Having found the off-line reference trajectory providing the nominal guidance commands, the final step for the baseline G&C architecture of this work is the attitude control subsystem (ACS). The ACS is designed to track this reference trajectory within prescribed desired dynamics together with nonlinear dynamic inversion control, which are presented next.

4.4. NONLINEAR DYNAMIC INVERSION CONTROL

Without loss of generality, consider a general multiple-input and multiple-output (MIMO) system whose number of inputs are equal to the number of outputs in order to avoid control allocation problems. Let's also assume momentarily that the nonlinear system can be described affine in the inputs as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{4.9a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{4.9b}$$

where $\boldsymbol{x} \in \mathscr{R}^n$ is the state vector, $\boldsymbol{u} \in \mathscr{R}^m$ is the control input vector, and $\boldsymbol{y} \in \mathscr{R}^m$ is the system output vector, the functions $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are assumed to be smooth vector fields on $\mathscr{R}^n$, and $\boldsymbol{g}(\boldsymbol{x}) \in \mathscr{R}^{n\times m}$ is a matrix whose columns are also assumed as smooth vector fields $\boldsymbol{g}_j$. Moreover, we consider $\boldsymbol{y} = \boldsymbol{x}$ so that the relative degree of each of the outputs $y_i,\ i = \{1,\ldots,m\}$ is one.

The idea of Nonlinear Dynamic Inversion (NDI) consists on canceling the nonlinearities in such nonlinear system so that the closed-loop dynamics is in a linear form. In other words, the nonlinear system is inverted by means of state feedback into a linear structure, and hence conventional linear controllers can be applied. A fundamental assumption is that the model of the system is exactly known, which gives NDI a great disadvantage from the point of view of uncertainties. Moreover, we also assume to have complete and accurate knowledge about the state of the system, which is hard to achieve in practice. NDI consists on the application of the following input transformation [61, 76, 78, 79]

$$\boldsymbol{u}_{\text{cmd}} = \boldsymbol{g}^{-1}(\boldsymbol{x})\left(\boldsymbol{v} - \boldsymbol{f}(\boldsymbol{x})\right) \tag{4.10}$$

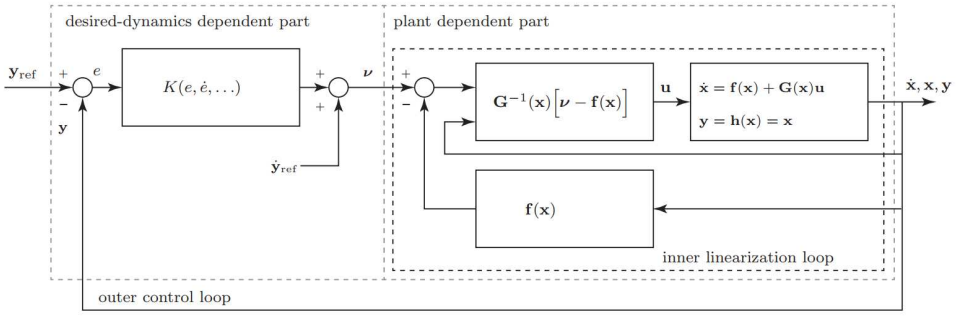


Figure 4.5: Nonlinear dynamic inversion tracking control for a nonlinear MIMO system (here, ref = cmd) [4].

which cancels all nonlinearities in closed-loop, and a simple linear input-output relationship between the new virtual control input $\boldsymbol{\nu}$ and the output $\boldsymbol{y}$ is obtained

$$\dot{\boldsymbol{y}} = \boldsymbol{\nu} \tag{4.11}$$

Apart from being linear, an interesting result from this relationship is that it is also decoupled since the input ν_i only affects the output y_i. From this fact, the input transformation (4.10) is called a *decoupling control law*, and the resulting linear system (4.11) is called the *single-integrator* form. This single-integrator form (4.11) can be rendered exponentially stable with

$$\boldsymbol{\nu} = \dot{\boldsymbol{y}}_{\text{des}} = \dot{\boldsymbol{y}}_{\text{cmd}} + \boldsymbol{K}_P\, \boldsymbol{e} \tag{4.12}$$

where $\dot{\boldsymbol{y}}_{\text{des}}$ defines the desired dynamics for the output vector or control variables, $\dot{\boldsymbol{y}}_{\text{cmd}}$ is the feedforward term for tracking, $\boldsymbol{e} = \boldsymbol{y}_{\text{cmd}} - \boldsymbol{y}$ is the error vector, $\boldsymbol{y}_{\text{cmd}}$ denotes the smooth desired output vector (at least one time differentiable), and $\boldsymbol{K}_P \in \mathscr{R}^{m\times m}$ is a diagonal matrix, whose $i-$th diagonal elements K_{P_i} are chosen so that the polynomials

$$s + K_{P_i}, \qquad i = \{1,\ldots,m\} \tag{4.13}$$

may become Hurwitz. This results in the exponentially stable and decoupled *desired* error dynamics

$$\dot{\boldsymbol{e}} + \boldsymbol{K}_P\, \boldsymbol{e} = \boldsymbol{0}, \tag{4.14}$$

which implies that $e_i(t) \to 0,\ i = \{1,\ldots,m\}$. From this typical tracking problem, and as illustrated in Figure 4.5, it can be seen that the entire control system will have two control loops [4, 78, 79]: the inner linearization loop based on Equation (4.10), and the outer control loop in Equation (10.46) based on linear control.

4.4.1. MULTI-LOOP NDI CONTROL

For preliminary controllability studies, we will be interested in multi-loop cascaded control architectures. Regarding the attitude control concept, designed to track the reference trajectory and its guidance commands, it is composed of two control loops assuming a sufficient time-scale separation between the attitude kinematics (aerodynamic angles outer-loop) and the rotational dynamics (angular rates inner-loop). In other words,

the inner-loop dynamics is assumed to be so fast, that from the outer loop perspective the angular rate commands are achieved instantaneously. With this assumption, the attitude controller is therefore performed in terms of nonlinear dynamic inversion NDI for each loop.

The outer-loop inversion of the attitude kinematics is very commonly done in attitude control to obtain reference commands for the inner-loop dynamics. In terms of the equations of angular motion, depending on the launcher or re-entry vehicle in consideration, the rotational dynamics can take different forms, especially when considering multi-body and variable mass dynamics. In this chapter, we assume that we have an accurate model to invert, and for the preliminary design studies considering stability aspects, we don't consider the effects of uncertainties and disturbances but we rather focus on the nominal behaviour and performance of the plant. In what follows, we denote the states

$$\boldsymbol{x}_1 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} \mu \\ \alpha \\ \beta \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} V \\ \gamma \\ \chi \end{bmatrix}$$

with p, q, r being the body roll, pitch, and yaw rates, respectively; μ, α, β, the aerodynamic bank, angle of attack, and aerodynamic sideslip angles, respectively; and V, γ, χ, the relative velocity of the launch vehicle, the flight path angle, and the flight path azimuth, respectively.

The following two-loop NDI attitude control architecture is largely based on [4, 49, 50, 160].

4.4.2. BODY ANGULAR RATE CONTROL LOOP

Regarding the body angular rate control loop, we are interested in the variable-mass attitude equations of motion as obtained by Eke [151]

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \left(\frac{{}^{R}\mathrm{d}\boldsymbol{I}}{\mathrm{d}t}\right)\boldsymbol{\omega} = \boldsymbol{M}_B + \boldsymbol{M}_V \tag{4.15}$$

where $\boldsymbol{M}_B \in \mathscr{R}^3$ is the external moment vector in body axes, $\boldsymbol{M}_V \in \mathscr{R}^3$ is the internal moment vector due to variable mass dynamics in body axes, $\boldsymbol{\omega} \in \mathscr{R}^3$ is the angular velocity vector, $\boldsymbol{I} \in \mathscr{R}^{3\times3}$ the inertia matrix of the rigid body, and the left superscript indicates that the time derivative is taken in a frame 'R' on the solid portion of the variable mass system.

The external moments in $\boldsymbol{M}_B$ are considered as the sum of moments partially generated by the aerodynamics of the airframe $\boldsymbol{M}_a$ and moments generated by control surface deflections $\boldsymbol{M}_c$, and we describe $\boldsymbol{M}_B$ linearly in the deflection angles $\boldsymbol{\delta}$ assuming the control derivatives to be linear as in [71] with $(\boldsymbol{M}_c)_\delta = \frac{\partial}{\partial\delta}\boldsymbol{M}_c$; therefore

$$\boldsymbol{M}_B = \boldsymbol{M}_a + \boldsymbol{M}_c = \boldsymbol{M}_a + (\boldsymbol{M}_c)_\delta \boldsymbol{\delta} \tag{4.16a}$$

where

$$\boldsymbol{M}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix}, \boldsymbol{M}_a = \begin{bmatrix} L_a \\ M_a \\ N_a \end{bmatrix}, \boldsymbol{M}_c = \begin{bmatrix} L_c \\ M_c \\ N_c \end{bmatrix} \tag{4.16b}$$

with L, M, N, the roll, pitch, and yaw moments, respectively; and

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \tag{4.16c}$$

corresponding to the control inputs: aileron, elevator, and rudder deflection angles, respectively. Furthermore, let $\boldsymbol{M}_V$ be the sum of internal moments generated by the variable mass dynamics as described in [151], where $\boldsymbol{M}_{V_1}$ is the so-called jet damping, $\boldsymbol{M}_{V_2}$ is due to the Coriolis effect (which can be neglected for axisymmetric motion as well as for negligible internal flow), $\boldsymbol{M}_H$ represents the rate of decrease of the system's angular momentum inside its boundary, and $\boldsymbol{M}_{thr}$ the moment of the thrust vector about the mass center; therefore

$$\boldsymbol{M}_V = \boldsymbol{M}_{V_1} + \boldsymbol{M}_{V_2} + \boldsymbol{M}_H + \boldsymbol{M}_{thr}. \tag{4.17}$$

The details of these terms are left to the reader and can be found in [151].

Since we will be interested in the body angular rate inversion, which is a state-input inversion problem [76, 78, 79], after a differentiation of the output variable

$$\boldsymbol{y}_1 = \boldsymbol{x}_1 = \boldsymbol{\omega}, \tag{4.18}$$

we obtain the dynamics of the rotational motion rewritten as the following set of differential equations

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\boldsymbol{M}_B + \boldsymbol{I}^{-1}\left[\, \boldsymbol{M}_V - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \left(\frac{{}^R\mathrm{d}\boldsymbol{I}}{\mathrm{d}t}\right)\boldsymbol{\omega} \,\right] \tag{4.19}$$

which inverted analytically yields

$$\bar{\boldsymbol{M}}_B = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \left[\, \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \left(\frac{{}^R\mathrm{d}\boldsymbol{I}}{\mathrm{d}t}\right)\boldsymbol{\omega} - \boldsymbol{M}_V \,\right] \tag{4.20}$$

where we have used the notation $\bar{\boldsymbol{M}}_B$ to denote that these moments are still commanded to the launch vehicle and that are to be produced by the aerodynamic surface deflections, the TVC, or by the RCS thrusters, depending on the configuration or the phase considered. Introducing the virtual control input

$$\boldsymbol{\nu}_\omega = \dot{\boldsymbol{\omega}}_{\mathrm{des}} = \begin{bmatrix} \dot{p}_{\mathrm{des}} \\ \dot{q}_{\mathrm{des}} \\ \dot{r}_{\mathrm{des}} \end{bmatrix} \tag{4.21}$$

and denoting the internal variable-mass terms as

$$\boldsymbol{M}_i = \left(\frac{{}^R\mathrm{d}\boldsymbol{I}}{\mathrm{d}t}\right)\boldsymbol{\omega} - \boldsymbol{M}_V \tag{4.22}$$

then the NDI control consists in the following transformation [61, 76, 79]

$$\bar{\boldsymbol{M}}_{B_{\mathrm{cmd}}} = \boldsymbol{I}\boldsymbol{\nu}_\omega + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{M}_i. \tag{4.23}$$

In other words

$$\begin{bmatrix} \bar{L}_{\text{cmd}} \\ \bar{M}_{\text{cmd}} \\ \bar{N}_{\text{cmd}} \end{bmatrix} = \begin{bmatrix} \bar{M}^x_{B_{\text{cmd}}} \\ \bar{M}^y_{B_{\text{cmd}}} \\ \bar{M}^z_{B_{\text{cmd}}} \end{bmatrix} = \boldsymbol{I} \begin{bmatrix} \dot{p}_{\text{des}} \\ \dot{q}_{\text{des}} \\ \dot{r}_{\text{des}} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \boldsymbol{M}_i. \tag{4.24}$$

Notice that whenever the variable-mass dynamics in $\boldsymbol{M}_i$ are not considered, then the Newton-Euler equations of motion for a rigid body are recovered and the NDI control design is further simplified. In general, depending on the nature of the propulsion system and its corresponding shape or assumed burn profiles, these terms can be further simplified and implemented in closed form for simulation and control aspects, see [111, 151, 161]. In this way, these loads can be included explicitly in the formulation of the dynamic equations of motion of the corresponding element of the vehicle so that their effect can be included in attitude control system as model-based feedforward terms.

The desired dynamics in Equations (4.21) and (4.23) are specified by prescribing the exponentially stable and decoupled desired error dynamics

$$\dot{\boldsymbol{e}}_\omega + \boldsymbol{K}_\omega \boldsymbol{e}_\omega = \boldsymbol{0}, \tag{4.25a}$$

where

$$\boldsymbol{e}_\omega = \boldsymbol{\omega}_{\text{cmd}} - \boldsymbol{\omega}, \tag{4.25b}$$

and

$$\boldsymbol{K}_\omega(s) = \begin{bmatrix} K_{\omega_p}(s) & 0 & 0 \\ 0 & K_{\omega_q}(s) & 0 \\ 0 & 0 & K_{\omega_r}(s) \end{bmatrix}. \tag{4.25c}$$

Here, $\boldsymbol{\omega}_{\text{cmd}}$ is obtained from the aerodynamic angles outer loop, and we have introduced $\boldsymbol{K}_\omega(s)$ as a diagonal matrix while assuming that the control law in Equation (4.23) is fully decoupling each input-output channel, which is not generally the case. These diagonal terms can be selected, for instance as a classical proportional-integral (PI) control [4, 50] with gains

$$K_{\omega_i}(s) = K_{P_i} + \frac{1}{s} K_{I_i}, \quad i = \{p, q, r\}, \tag{4.26}$$

resulting in the closed loop system

$$\dot{\boldsymbol{\omega}}_{\text{des}} = \dot{\boldsymbol{\omega}}_{\text{cmd}} + \boldsymbol{K}_{P_\omega}(\boldsymbol{\omega}_{\text{cmd}} - \boldsymbol{\omega}) + \boldsymbol{K}_{I_\omega} \int (\boldsymbol{\omega}_{\text{cmd}} - \boldsymbol{\omega}) \mathrm{d}t \tag{4.27}$$

with the gains

$$\boldsymbol{K}_{P_\omega} = \begin{bmatrix} K_{P_p} & 0 & 0 \\ 0 & K_{P_q} & 0 \\ 0 & 0 & K_{P_r} \end{bmatrix}, \boldsymbol{K}_{I_\omega} = \begin{bmatrix} K_{I_p} & 0 & 0 \\ 0 & K_{I_q} & 0 \\ 0 & 0 & K_{I_r} \end{bmatrix}. \tag{4.28}$$

Whenever aerodynamic control surfaces are used, the aerodynamics of the airframe and the moments generated by the control surface deflections plays an important role in the dynamic inversion since these terms are hardly known exactly for model inversion.

Since we assumed in (4.16a) that the control derivatives are linear, the dynamics can be rewritten as the following

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}(\boldsymbol{M}_c)_{\delta}\boldsymbol{\delta} + \boldsymbol{I}^{-1}\left[\, \boldsymbol{M}_a - \boldsymbol{M}_i - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \,\right] \tag{4.29}$$

and then the NDI control consists in the following expression

$$\boldsymbol{\delta}_{\text{cmd}} = (\boldsymbol{M}_c)_{\delta}^{-1}\left[\, \boldsymbol{I}\boldsymbol{\nu}_{\omega} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{M}_i - \boldsymbol{M}_a \,\right]. \tag{4.30}$$

Here, we have also assumed that the control derivatives are invertible in the whole domain of operation, and that $dim(\boldsymbol{\delta}) = dim(\boldsymbol{y})$, meaning that the number of control variables and control effectors are equal. In the usual case where $dim(\boldsymbol{\delta}) \geq dim(\boldsymbol{y})$, meaning that there are more aerodynamic control surfaces than variables to be controlled, control allocation is required. The opposite case, meaning $dim(\boldsymbol{\delta}) \leq dim(\boldsymbol{y})$, leads to internal dynamics that must be studied in terms of stability, and the system is said to be underactuated. These aspects are however out of the scope of this chapter.

4.4.3. AERODYNAMIC ANGLES OUTER-LOOP

The aerodynamic angles outer-loop inversion procedure is the same as shown before. Since we will be interested in the attitude kinematics inversion, which is also a state-input inversion considering the body angular rate as intermediate control inputs, denoting the output vector

$$\boldsymbol{y}_2 = \boldsymbol{x}_2, \tag{4.31}$$

the differentiation of this output variable yields the attitude kinematics in terms of the aerodynamic angles as expressed in [51, 160, 162] by

$$\dot{\boldsymbol{x}}_2 = \begin{bmatrix} \dot{\mu} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \boldsymbol{f}_2 + \boldsymbol{G}_2 \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.32}$$

where the angular velocity terms in $\boldsymbol{f}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2, \boldsymbol{x}_3)$, omitted here, are nonlinear functions of the translational terms $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ and their derivatives [51, 160, 162, 163] and

$$\boldsymbol{G}_2 = \frac{1}{\cos\beta} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ \sin\alpha & 0 & -\cos\alpha \end{bmatrix}. \tag{4.33}$$

Since this kinematic equation is nonlinear but affine in the angular rates, and in the case that the angular velocity terms contained in $\boldsymbol{f}_2$ are assumed or regarded as very small and neglectable for the attitude control subsystem, as it is commonly done in the literature [51, 160], this inner-loop can be readily found by applying the following simple inversion

$$\boldsymbol{\omega}_{\text{cmd}} = \boldsymbol{G}_2^{-1}\boldsymbol{\nu}_{\text{att}} \tag{4.34}$$

where we have introduced the virtual control input for this loop as

$$\boldsymbol{\nu}_{\text{att}} = \dot{\boldsymbol{x}}_{2_{\text{des}}} = \begin{bmatrix} \dot{\mu}_{\text{des}} \\ \dot{\alpha}_{\text{des}} \\ \dot{\beta}_{\text{des}} \end{bmatrix}. \tag{4.35}$$

Otherwise, the angular velocity quantities $\boldsymbol{f}_2$ can be added as model-based feedforward terms to the guidance command generation. In other words, we have obtained the outer-loop rate commands as

$$\begin{bmatrix} p_{\text{cmd}} \\ q_{\text{cmd}} \\ r_{\text{cmd}} \end{bmatrix} = \boldsymbol{G}_2^{-1} \begin{bmatrix} \dot{\mu}_{\text{des}} \\ \dot{\alpha}_{\text{des}} \\ \dot{\beta}_{\text{des}} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & 0 & \sin\alpha \\ \sin\beta & 1 & 0 \\ \sin\alpha\cos\beta & 0 & -\cos\alpha \end{bmatrix} \begin{bmatrix} \dot{\mu}_{\text{des}} \\ \dot{\alpha}_{\text{des}} \\ \dot{\beta}_{\text{des}} \end{bmatrix}. \tag{4.36}$$

To finish the attitude control design, the desired aerodynamic angles are specified by prescribing the exponentially stable and decoupled desired error dynamics

$$\dot{\boldsymbol{e}}_{\text{att}} + \boldsymbol{K}_{\text{att}} \boldsymbol{e}_{\text{att}} = \boldsymbol{0}, \tag{4.37a}$$

where

$$\boldsymbol{e}_{\text{att}} = \boldsymbol{x}_{2\text{cmd}} - \boldsymbol{x}_2, \tag{4.37b}$$

and

$$\boldsymbol{K}_{\text{att}}(s) = \begin{bmatrix} K_{\text{att}_\mu} & 0 & 0 \\ 0 & K_{\text{att}_\alpha} & 0 \\ 0 & 0 & K_{\text{att}_\beta} \end{bmatrix}, \tag{4.37c}$$

with

$$K_{\text{att}_i} = K_{P_i} + \frac{1}{s} K_{I_i}, \quad i = \{\mu, \alpha, \beta\}, \tag{4.37d}$$

resulting in the closed loop system

$$\dot{\boldsymbol{x}}_{2_{\text{des}}} = \dot{\boldsymbol{x}}_{2_{\text{cmd}}} + \boldsymbol{K}_{P_{\text{att}}}(\boldsymbol{x}_{2\text{cmd}} - \boldsymbol{x}_2) + \boldsymbol{K}_{I_{\text{att}}} \int (\boldsymbol{x}_{2\text{cmd}} - \boldsymbol{x}_2)\,\mathrm{d}t, \tag{4.38}$$

with the gains

$$\boldsymbol{K}_{P_{\text{att}}} = \begin{bmatrix} K_{P_\mu} & 0 & 0 \\ 0 & K_{P_\alpha} & 0 \\ 0 & 0 & K_{P_\beta} \end{bmatrix}, \boldsymbol{K}_{I_{\text{att}}} = \begin{bmatrix} K_{I_\mu} & 0 & 0 \\ 0 & K_{I_\alpha} & 0 \\ 0 & 0 & K_{I_\beta} \end{bmatrix}, \tag{4.39}$$

and which concludes the attitude control design.

4.5. NONLINEAR FLIGHT CONTROL SIMULATION

A nonlinear flight control simulation for the position and attitude control of the horizontal takeoff and horizontal landing launch vehicle concept AURORA [5, 158] is here presented.

4.5.1. MISSION PROFILE

The AURORA two-stage-to-orbit (TSTO) concept has been studied at the *Space Launcher System Analysis* (SART) department of the DLR Institute of Space Systems (DLR-RY). This concept considered iterative studies regarding mass budget, propulsion, aerodynamics, and structural optimization amongst many others.

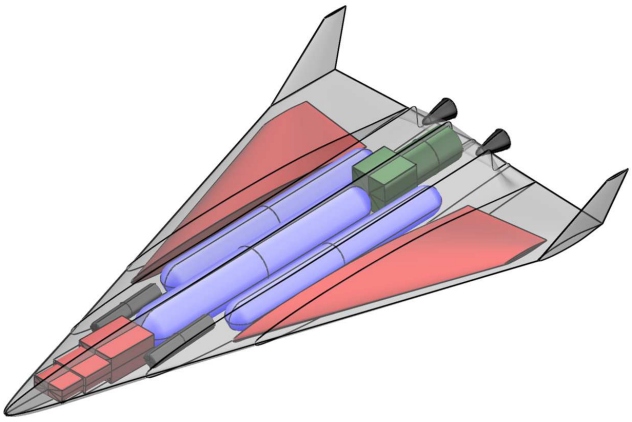


Figure 4.6: AURORA-RLV concept [5].

The concept, shown in Figure 4.6, aims to reduce operational costs while increasing launch frequency [5]. This is done by considering a more 'aircraft-like' operation providing a high lift-to-drag ratio and a propellant combination of LOX/Kerosene allowing placement of the kerosene tanks in the wing structure.

The trajectory optimization of this concept has been performed and shown in [3], where the following goals, requirements, and constraints were considered:

- The ascent of the launch vehicle starts at a launch site located at -52.77° latitude, 5.24° longitude, and zero altitude.
- The descent of the launch vehicle to the landing site (-64.68° latitude, 32.36° longitude, zero altitude) has to be guaranteed within a radius of approximately 25 km.
- The payload mass shall be maximized while the upper stage propellant mass is traded for the payload mass.
- The polar orbit with an apogee altitude of 1200 km has to be reached at an inclination of 90° and maximum perigee.
- The following constraints have to be considered to reduce mechanical or thermal loads on the structure:
  - Maximum acceleration n_x lower than 4.5 g.
  - Maximum acceleration n_z lower than 1.75 g (ascent).
  - Maximum acceleration n_z lower than 4.25 g (descent).
  - Maximum dynamic pressure lower than 50 kPa (ascent).
  - Maximum dynamic pressure lower than 60 kPa for the re-entry and the flight to the landing site.
  - Maximum heat flux lower than 900 kW/m^2 for a theoretic reference nose radius of 0.15 m.

Table 4.1: Trajectory optimization phases considered for AURORA [3].

Phase	Stages	Description
P1	US+MS	Horizontal liftoff
P2	US+MS	Ascent phase (rocket engines)
P3	US+MS	Ballistic phase & separation
P4	US	Ascent of the upper stage
P5	MS	Descent maneuver & return

Table 4.2: Sub-phases for AURORA descent maneuver.

Phase	Actuators	Description
P5-a	RCS (+ Fins & Flaps)	Re-entry
P5-b	Fins & Flaps	Skipping
P5-c	Fins & Flaps	Final approach

The trajectory phases considered with *trajOpt* are listed in Table 4.1. *Phase P1* considers the horizontal liftoff powered by rocket and air-breathing engines up to $M_a \approx 1$ followed by an ascent *Phase P2* powered only by the rocket engines. *Phase P3* represents a ballistic phase up until the separation of the upper stage stored in the payload bay, initiated at a separation velocity of approximately 5 km/s. Consequently, the *Phase P4* represents the ascent of the upper stage. *Phase P5* represents the unpowered re-entry maneuver and the return flight of the launch vehicle to the chosen landing site by using the air-breathing engines. The ferry flight from the landing site back to the launch site is not considered. Furthermore, *Phase P5* is divided in three sub-phases for trajectory optimization as shown in Table 4.2.

4.5.2. NONLINEAR DESCENT FLIGHT CONTROL

Flight simulations on the full 6-DOF nonlinear system are performed for the *Phase P5* since it covers interesting scenarios, such as the re-entry flight and the potential to study the combination of RCS with aerodynamic surface controls during descent. Moreover, there are no variable mass dynamics since this is an unpowered descent maneuver, making the control study much simpler. The simulations are done with the double-loop NDI-based attitude control system to track the generated optimal trajectory of the launch vehicle. We do this preliminarily without being subject to any disturbances or uncertainties, and under the nominal conditions to verify if the plant is controllable during the descent, and within which range in terms of RCS budget and aerodynamic surface controls.

Figure 4.7 shows the descent guidance and control results for the re-entry maneuver of *Phase P5-a* using only RCS control. This is the baseline scenario considering the attitude control entirely actuated by the RCS thrusters. Since only attitude control is performed, Fig. 4.7-a shows the resulting 'open loop' kinematic position trajectory which

is entirely done by means of the nominal attitude tracking control (the relative velocity is shown normalized according to the whole *Phase P5*). Fig. 4.7-b shows the resulting tracking performance of the attitude control system in terms of the commanded aerodynamic angles. In Fig. 4.7-c the resulting and commanded rates from the dynamic inversion can be seen; here, the pitch rate q and its commanded values differ because of the highly cross products involved, which are solved for automatically in the inner loop to obtain the required moments and which are not accounted for in the outer-loop commands. The re-entry maneuver demands quite high pitching moments as demonstrated by Fig. 4.7-d, where the required commanded moments are shown normalized with respect to the complete *Phase P5*. These results show that the system is controllable under the nominal conditions if the obtained bounds of commanded moments are achievable in practice.

As a test-case scenario, we investigate what happens whenever this *Phase P5-a* can be performed in combination with aerodynamic surface controls. This can be done only after $t = 500$ s when the launch vehicle has already entered in the atmosphere below an altitude of $h = 120$ km and therefore commandable in terms of aerodynamic forces and moments. Since the aileron and the elevator commands the flaps simultaneously, we have to restrict these commands such that the combined maximum deflection limit for each flap does not exceed ±30 deg. The same limits apply for the fins which are actuated with the same limits of ±30 deg. In that sense, we limit the elevator commands to ±20 deg and the aileron commands to ±10 deg (as an initial guideline, not optimized). Figure 4.10 shows the descent guidance and control results for the re-entry maneuver of *Phase P5-a* of the combined aerodynamic surface control and RCS. Besides the open-loop flight path and the nominal attitude tracking performance results, this figure shows the resulting impact of the demanded pitching moments of Fig. 4.10-c as compared to the ones with RCS thrusters only in Fig. 4.7-c (normalized). These results show that, while the system is still controllable under the nominal conditions considered, the impact on the RCS budget can be significant while maintaining certain bounds on the aerodynamic actuator efforts. This also showcase the potential benefit in launch vehicle's design that improvements in terms of impulse budgeting (and therefore propellant mass) can already be obtained at preliminary design levels.

In that sense, Fig. 4.8 shows the resulting aerodynamic control surfaces for *Phase P5-a* in combination with RCS control. The allocation of control surfaces vs. RCS thrust could be further optimized to avoid actuator saturations or to minimize fuel consumption within some actuation limits; however, this subject is not further investigated here. This scenario considering the combined RCS thrusters and aerodynamic control surfaces showcase the potential to reduce by more than half the angular impulse budget for the RCS as shown in Fig. 4.9. This impact on the RCS budget can lead to further improvements in terms of the launch vehicle preliminary design, since the dimensioning and location of the RCS thrusters can also have a considerable impact on the vehicle configuration.

To conclude the study, Fig. 4.11 shows the descent guidance and control results for the *Phase P5-c* which is the final approach of the descent. The attitude control of this phase is entirely performed by aerodynamic surfaces since they can produce the aerodynamic moments required. Once again, these results show that the system is controllable

under the nominal conditions and within the bounds of the aerodynamic control surfaces.

4.6. CONCLUSIONS

This chapter presented a baseline and general-purpose off-line G&C architecture for reusable launch vehicles for the early stability and controllability studies during preliminary design phases of generic launcher conceptual designs.

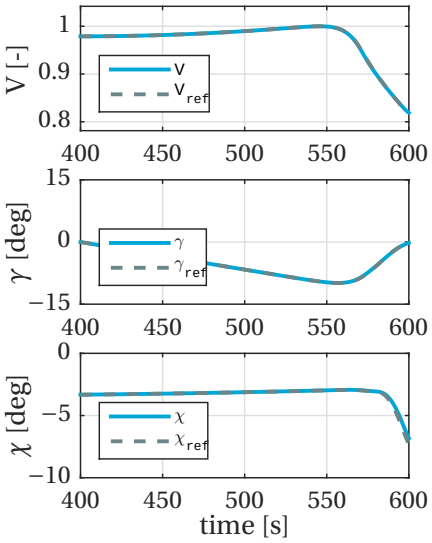
Optimal reference trajectories and guidance commands were obtained with a direct approach using the trajectory optimization package '*trajOpt*' in combination with MODELICA models, while inner-loop attitude control was designed in terms of nonlinear dynamic inversion together with prescribed desired error dynamics. Such optimal reference trajectory tracking helps to answer the motivating questions presented in the introduction.

To demonstrate our integrated approach, the AURORA reusable launch vehicle concept was investigated in the context of the methods presented here. The nonlinear control system, simulated for the descent phase including the re-entry flight and covering a wide flying envelope ranging from Mach 18 to Mach 5 and angles of attack between 50 and 9 deg, demonstrate the controllability of the launch vehicle as well as the potential to reduce more than half the impact on the angular impulse budget for the RCS by combining it with aerodynamic surface controls during the re-entry phase.

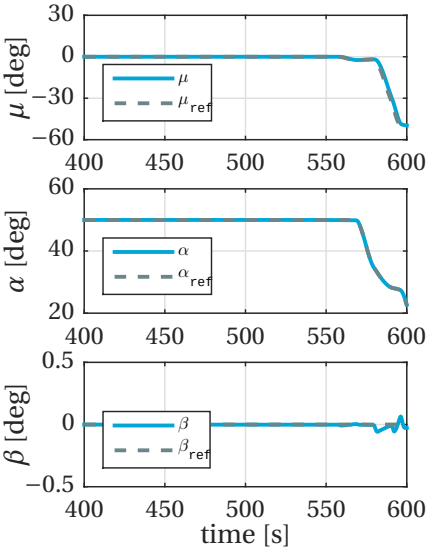
Flight simulations show that the control system accurately tracks commands in aerodynamic angles but preliminarily without being subject to significant aerodynamic uncertainties. This will be part of future work, which will consider and include more detailed analysis of the effect of parametric and aerodynamic uncertainties, as well as external perturbations such as wind and turbulence on the overall G&C and control performance.

ACKNOWLEDGMENTS

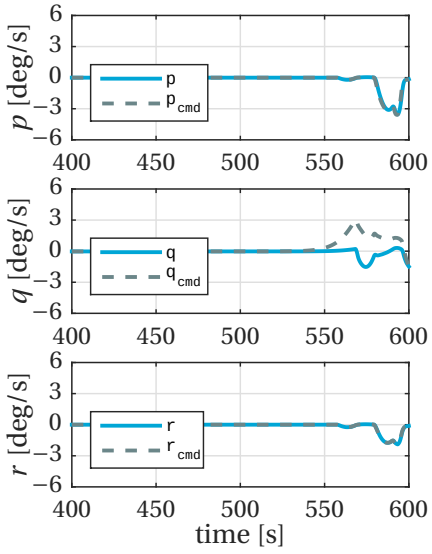
This study was performed within the DLR projects AKIRA and X-TRAS regarding preliminary system studies and evaluation of key technologies for future reusable launch vehicles. The authors would like to thank DLR colleagues from these projects, in particular the colleagues from the *Space Launcher System Analysis* (SART) department of the DLR Institute of Space Systems (DLR-RY) in Bremen for their suggestions and discussions, and for providing data of the considered launch vehicle configurations.



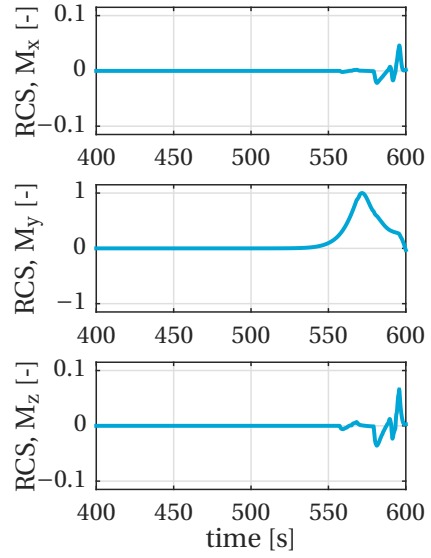
(a) Kinematic position trajectory results, uncontrolled version, nominal case. Reference values from trajectory optimization.



(b) Aerodynamic angles – attitude control results, nominal case. Reference values from trajectory optimization.



(c) Outer-loop angular rates guidance and control results, nominal case. Guidance obtained from model inversion, NDI control.



(d) Inner-loop attitude control results, nominal case. Obtained moments required for RCS control in body-frame.

Figure 4.7: Phase P5-a – Descent guidance and control results for the re-entry maneuver using RCS control.

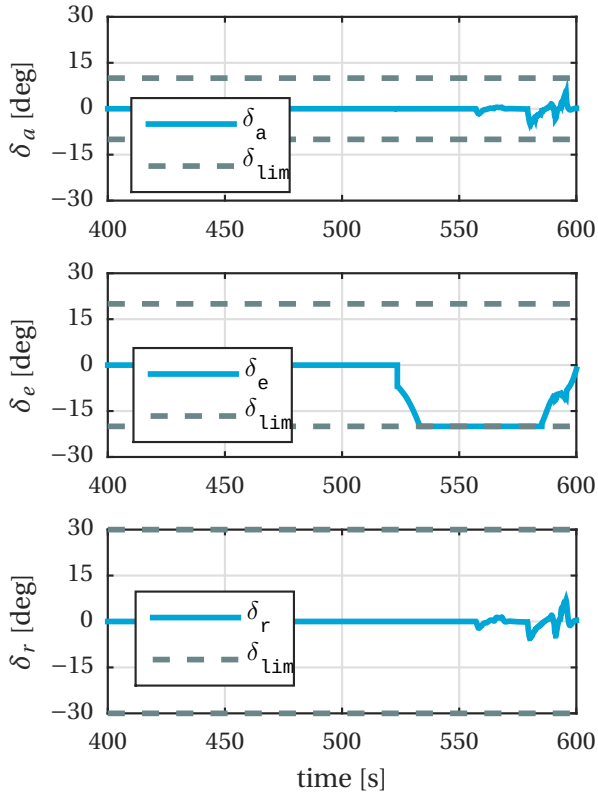


Figure 4.8: Phase P5-a – Aerodynamic control surface deflections used in combination with RCS control during re-entry.

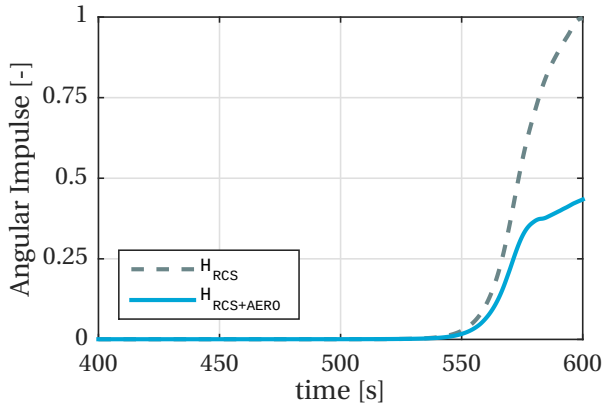
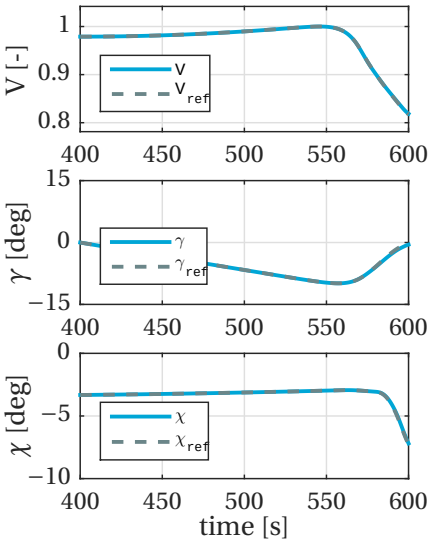
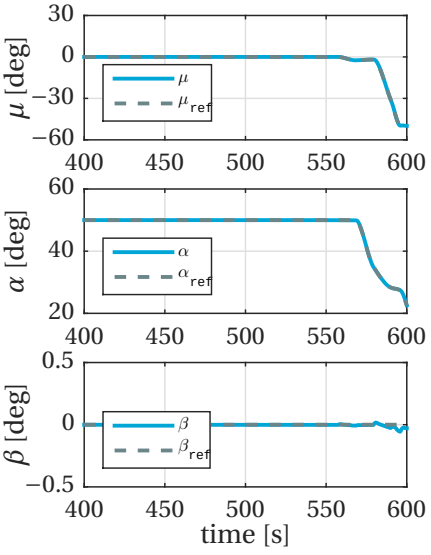


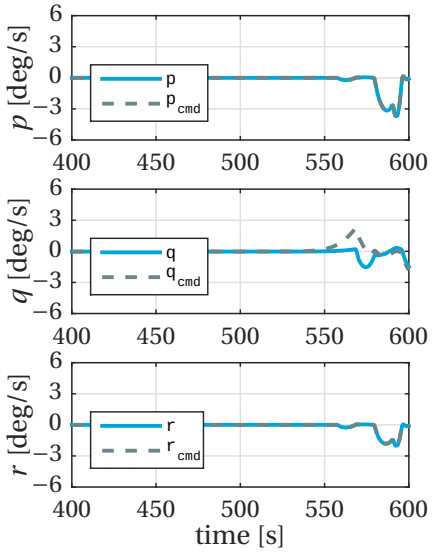
Figure 4.9: Phase P5-a – Impact on the resulting angular impulse required for RCS when using with and without aerodynamic controls.



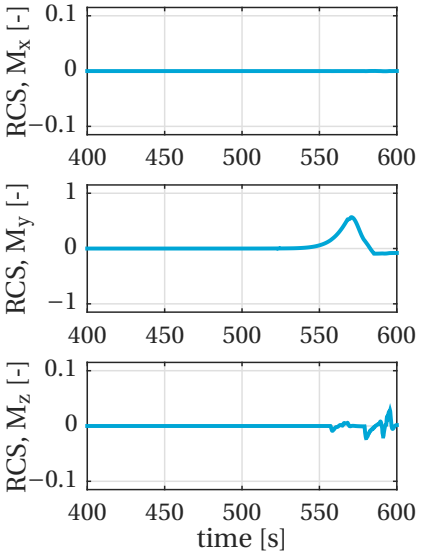
(a) Kinematic position trajectory results, uncontrolled version, nominal case. Reference values from trajectory optimization.



(b) Aerodynamic angles – attitude control results, nominal case. Reference values from trajectory optimization.

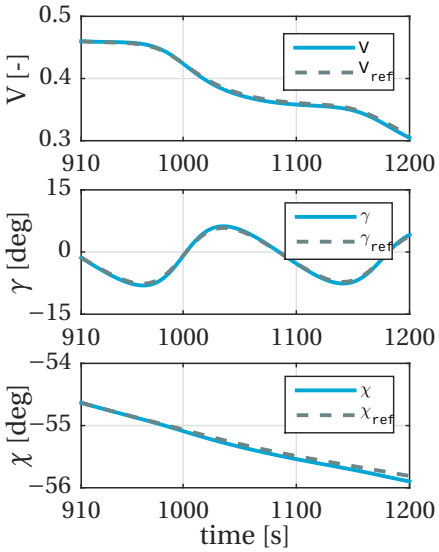


(c) Outer-loop angular rates guidance and control results, nominal case. Guidance obtained from model inversion, NDI control.

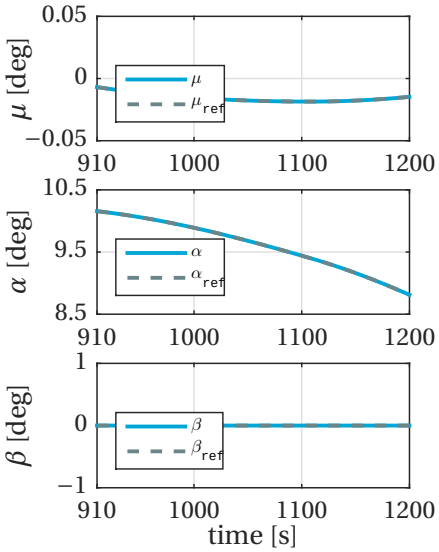


(d) Inner-loop attitude control results, nominal case. Obtained moments required for RCS control in body-frame.

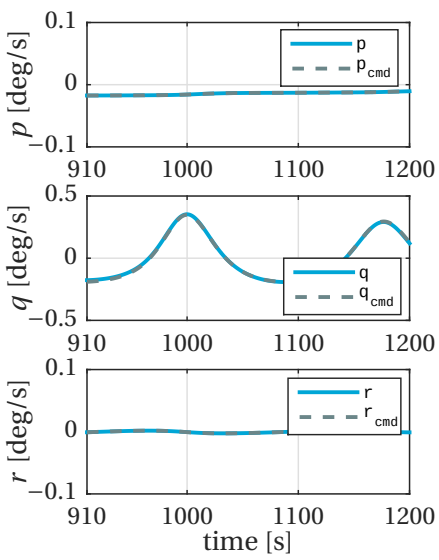
Figure 4.10: Phase P5-a – Descent guidance and control results for the re-entry maneuver using RCS and aerodynamic surface control.



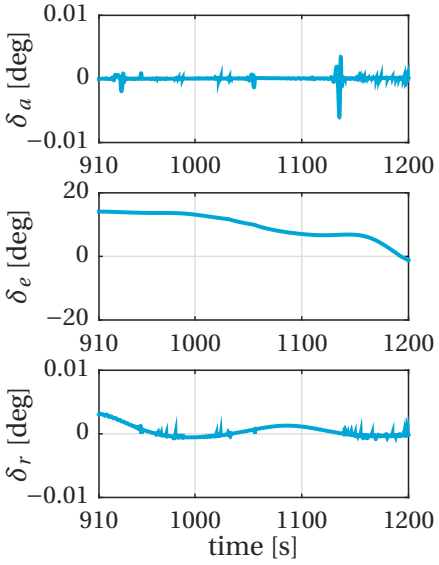
(a) Kinematic position trajectory results, uncontrolled version, nominal case. Reference values from trajectory optimization.



(b) Aerodynamic angles – attitude control results, nominal case. Reference values from trajectory optimization.



(c) Outer-loop angular rates guidance and control results, nominal case. Guidance obtained from model inversion, NDI control.



(d) Inner-loop attitude control results, nominal case. Obtained aerodynamic control surface deflections.

Figure 4.11: Phase P5-c – Descent guidance and control results for final approach of the descent.

5

Fast Slew Maneuvers for the High-Torque-Wheels BIROS Satellite

Abstract

The satellite platform BIROS is the second technology demonstrator of DLR's 'FireBIRD' space mission aiming to provide infrared remote sensing for early fire detection. Among several mission goals and scientific experiments, to demonstrate a high-agility attitude control system, the platform is actuated with an extra array of three orthogonal 'High-Torque-Wheels'. However, to enable agile reorientation, a challenge arises from the fact that time-optimal slew maneuvers are, in general, not of the Euler-axis rotation type; especially whenever the actuators are constrained independently. Moreover, BIROS' on-board computer can only accommodate rotational acceleration commands twice per second. The objective is therefore to find a methodology to design fast slew maneuvers while considering a highly dynamic plant commanded by piecewise-constant sampled-time control inputs. This is achieved by considering a comprehensive analytical nonlinear model for spacecraft equipped with reaction wheels and transcribing a time-optimal control problem formulation into a multi-criteria optimization problem. Solutions are found with a direct approach using the trajectory optimization package 'trajOpt' of DLR-SR's optimization tool, Multi-Objective Parameter Synthesis (MOPS). Results based on numerical simulations are presented to illustrate this method.

Publication

Paul Acquatella B.: *Fast Slew Maneuvers for the High-Torque-Wheels BIROS Satellite.* In: Transactions of the Japan Society of Aeronautical and Space Sciences, Vol. **61**, No. 2, pp. 79–86, 2018; presented at *ISSFD 2017, 26th International Symposium on Space Flight Dynamics, June 3-9, 2017, Matsuyama-Ehime, Japan.*

5.1. INTRODUCTION

THE SATELLITE platform *Bi-spectral InfraRed Optical System* (BIROS) [27], successfully launched into space on 22 June 2016 at 05:55 CEST, is the second technology demonstrator along with the TET-1 satellite of the DLR R&D 'FireBIRD' [28] space mission aiming to provide infrared (IR) remote sensing for early fire detection (forest fires, volcanic activity, gas flares and industrial hotspots). These small satellites are extensions and largely based on the flight-proven *Bi-spectral Infra-Red Detection* (BIRD) [34, 35] satellite bus launched in 2001.

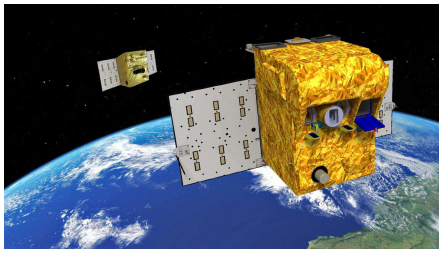


Figure 5.1: FireBIRD – a satellite duo for fire detection. BIROS (front), TET-1 (back). Credit: DLR, CC-BY 3.0.

Among several mission goals and scientific experiments, the platform is actuated with an extra array of three orthogonal *'High-Torque-Wheels' (HTW)* [25, 26] to demonstrate a high-agility attitude control system. Since the fast slew maneuvers are meant to be performed mainly by the *HTW* array, the satellite platform's main torque actuators, as with TET-1, are four precise *'RW-90'* reaction wheels [164] in a redundant tetrahedron configuration. Wheel characteristics for both the *HTW* and the *RW-90* are presented in Table 1, while BIROS' reaction wheel array alignment parameters, as described in the kinematics section, are presented in Table 2.

One of the main requirements for the *HTW* experiment is being able to perform 30deg 1-axis rotations in 10s around a principal axis with an inertia of $10\text{Kg}\cdot\text{m}^2$. For three-axis rotations, rotation paths are, in general, not prescribed in the requirements and these rotations are desired to be performed in *minimal time*. The experiment is originally designed to be implemented in the '*Fast Slew*' mode of BIROS' Attitude Control System (ACS). See Löw et al. [165] for a detailed description of other (main) modes, which are similar to the ones implemented for the TET-1 satellite [166, 167] of the FireBIRD constellation.

However, for agile reorientation, a challenge arises from the fact that time-optimal slew maneuvers are, in general, not of the *Euler-axis* rotation [136, 137] type; especially whenever the actuators are constrained independently [131], as it will be in this case. Moreover, the BIROS on-board computer (OBC) can only accommodate rotational acceleration commands twice per second, which means that these must be piecewise-constant sampled-time control inputs.

The topic of optimal spacecraft rotational maneuvers is quite extensive [127] and has been studied for many decades. Earlier works [128, 129] considered numerical approaches and quasi-closed-form solutions to reorientation problems, while only recently new results have been found for minimum-time and time-optimal reorientation maneu-

Table 5.1: Wheel characteristics [26, 164, 166, 167]

Performance	*RW-90*	*HTW*
Nominal speed [rpm]	6000	1825
Max. speed [rpm]	7800	3000
Nominal torque [Nm]	0.015	0.21
Max. torque [Nm]	0.021	0.23
Nominal ang. momentum [Nms]	0.2639	0.9556
Max. ang. momentum [Nms]	0.3431	1.5708
Mechanics		
Number of wheel units	4	3
Moment of inertia [Kg$\cdot$m^2]	4.2×10^{-4}	5×10^{-3}

vers [130–133] for more generic configurations. Some of these results have been experimentally validated for imaging satellites in-orbit [138]. Time-optimal reorientation solutions for rigid bodies have also been found using a geometric mechanics approach [134, 135] together with indirect optimization. However, most of the work reported in literature does not consider time-optimal control solutions of spacecraft equipped with reaction wheels driven by independently constrained piecewise-constant sampled-time control inputs.

This motivates the objective of this chapter, which is to find a methodology to design fast slew maneuvers for the BIROS *HTW* experiment while considering a highly dynamic plant commanded by piecewise-constant sampled-time control inputs. The offline solutions considered in this chapter are mainly oriented to rest-to-rest maneuvers and will be implemented as sampled-input feedforward commands in combination with error feedback control in a two-degrees-of-freedom control system architecture.

This is achieved by 1) considering a comprehensive analytical nonlinear model for spacecraft equipped with reaction wheels; 2) considering the outer-loop control as the feedforward commands designed here; 3) transcribing a time-optimal control problem formulation into a direct approach involving a multi-criteria optimization problem considering inequality and equality constraints; and 4) solving the transcribed problem directly using the trajectory optimization package *'trajOpt'* of DLR-SR's optimization tool, *Multi-Objective Parameter Synthesis (MOPS)*. To obtain the desired piecewise-constant sampled-time inputs, the methodology proposed follows a sequential three-step procedure. Finally, numerical simulations of the procedure steps proposed are presented.

5.2. MODELING OF SPACECRAFT WITH REACTION WHEELS

In this section a comprehensive nonlinear rotational dynamics model for spacecraft is described. The model includes a generic set of reaction wheels in arbitrary configuration which are driven by exogenous inputs provided by each wheel's powertrain.

Table 5.2: BIROS' reaction wheel array alignment parameters

Alignment vector	Value
$\mathbf{a}_1$ (*HTW*)	$\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\top$
$\mathbf{a}_2$ (*HTW*)	$\begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^\top$
$\mathbf{a}_3$ (*HTW*)	$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$
$\mathbf{a}_4$ (*RW-90*)	$\begin{pmatrix} 0 & -1 & 0 \end{pmatrix}^\top$
$\mathbf{a}_5$ (*RW-90*)	$\begin{pmatrix} \frac{1}{2}\sqrt{3}\sin\theta & \cos\theta & -\frac{1}{2}\sin\theta \end{pmatrix}^\top$
$\mathbf{a}_6$ (*RW-90*)	$\begin{pmatrix} -\frac{1}{2}\sqrt{3}\sin\theta & \cos\theta & -\frac{1}{2}\sin\theta \end{pmatrix}^\top$
$\mathbf{a}_7$ (*RW-90*)	$\begin{pmatrix} 0 & \cos\theta & \sin\theta \end{pmatrix}^\top$
Alignment angle	
θ [rad]	$\arccos(-1/3)$

5.2.1. KINEMATICS

Consider, first, an array consisting of n reaction wheels. Introducing unit vectors $\mathbf{a}_i$, which give the orientation of the spin-axis of each reaction wheel with respect to the spacecraft coordinate system collected in the configuration or alignment matrix

$$A = \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array} \right], \tag{5.1}$$

then each $\mathbf{a}_i$ can define the $i-$th reaction wheel or 'actuator' frame by taking $\mathbf{a}_i$ as the first axis and making the remaining axes constitute an orthogonal frame. The kinematics of the $i-$th reaction wheel with respect to its corresponding actuator frame in terms of its spin-axis angle $\Phi_{w,i}$ and angular velocity $\Omega_{w,i}$ is simply given by

$$\dot{\Phi}_{w,i} = \Omega_{w,i} \qquad i = 1, \ldots, n. \tag{5.2}$$

Next, consider the spacecraft equipped with the n reaction wheels just introduced. Rotation matrices $\mathbf{R} \in \mathscr{SO}(3)$, representing a linear transformation of vectors in body-fixed or 'hub' frame into the inertial frame, are preferred as the attitude parameterization since they are both global and unique [168], where the configuration space or manifold of rotation matrices [134] is given by the special orthogonal group $\mathscr{SO}(3)$ with the conditions

$$\mathscr{SO}(3) = \{\mathbf{R} \in \mathscr{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathscr{I}_{3\times 3},\ \det[\mathbf{R}] = 1\}.$$

In that sense, the kinematics of the full spacecraft with respect to the inertial frame, and in terms of its rotation matrix $\mathbf{R}$ and angular velocity $\boldsymbol{\omega} \in \mathscr{R}^3$, is given by

$$\dot{\mathbf{R}} = \mathbf{R} \cdot S(\boldsymbol{\omega}). \tag{5.3}$$

The *skew map* $S(\cdot) : \mathscr{R}^3 \mapsto \mathfrak{so}(3)$ is a linear isomorphism between $\mathscr{R}^3$ and the Lie algebra $\mathfrak{so}(3)$, which represents 3×3 skew-symmetric matrices, and is defined by the condition

that $S(x)\ y = x \times y$ for any $x, y \in \mathscr{R}^3$, or algebraically as

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

The inverse of the skew map is denoted by the *vee map* $\vee : \mathfrak{so}(3) \mapsto \mathscr{R}^3$.

5.2.2. DYNAMICS

Following the derivations in Karpenko et al. [138], a rotational dynamics model is obtained as follows. First, consider the angular momentum of the spacecraft equipped with the reaction wheel array in question

$$\boldsymbol{H} = \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{h} \tag{5.4}$$

where, expressed in a body-fixed frame, $\boldsymbol{H} \in \mathscr{R}^3$ is the total angular momentum of the system, $\boldsymbol{I} \in \mathscr{R}^{3\times3}$ is the constant inertia matrix of the spacecraft including the reaction wheels, $\boldsymbol{\omega} \in \mathscr{R}^3$ is the spacecraft angular velocity, and $\boldsymbol{h} \in \mathscr{R}^3$ is the total angular momentum vector associated with the reaction wheel array. The angular momentum $\boldsymbol{h}$ can be expressed from individual actuator frames to the body-fixed frame as

$$\boldsymbol{h} = \sum_{i=1}^{n} \mathbf{a}_i \boldsymbol{h}_{w,i} = A\,\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}, \tag{5.5}$$

where $\boldsymbol{I}_{\boldsymbol{w}}$ is a diagonal matrix of reaction wheel spin-axis inertia values

$$\boldsymbol{I}_{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{I}_{w,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{I}_{w,n} \end{bmatrix},$$

and $\boldsymbol{\Omega}$ is the inertial angular rate of the reaction wheel array

$$\boldsymbol{\Omega} = \Omega_w + A^\top \boldsymbol{\omega}.$$

The term $A^\top\boldsymbol{\omega}$ is the extra angular motion relative to the spacecraft. Considering the angular momentum associated with the $i-$th reaction wheel in the actuator frame

$$\boldsymbol{h}_{w,i} = \boldsymbol{I}_{w,i}\big(\,\Omega_{w,i} + \mathbf{a}_i^\top\boldsymbol{\omega}\,\big), \qquad i = 1,\ldots,n, \tag{5.6}$$

after which the differential equation describing the reaction wheel dynamics in terms of reaction wheel torques $\tau_{w,i}$ can be obtained

$$\dot{\Omega}_{w,i} = \boldsymbol{I}_{w,i}^{-1}\tau_{w,i} - \mathbf{a}_i^\top\dot{\boldsymbol{\omega}}, \qquad i = 1,\ldots,n. \tag{5.7}$$

Here, the reaction wheel torques are considered as the exogenous inputs to the system and are provided by the wheel's powertrain.

Because the angular momentum must be conserved in the absence of external perturbations, applying the transport theorem [127, 138] to Eq. (10.7), the following relation is obtained

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H}+\boldsymbol{\omega}\times\boldsymbol{H}=0, \tag{5.8}$$

which can be further expanded as

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}+A\boldsymbol{I}_{\boldsymbol{w}}\dot{\boldsymbol{\Omega}}+\boldsymbol{\omega}\times\Big(\boldsymbol{I}\boldsymbol{\omega}+A\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}\Big)=0. \tag{5.9}$$

Combining Eqs. (5.5), (5.7), and (5.9), the comprehensive nonlinear model for spacecraft dynamics equipped with reaction wheels [138] is given by

$$\boldsymbol{\Gamma}\begin{bmatrix}\dot{\boldsymbol{\omega}}\\ \dot{\Omega}_{w,1}\\ \vdots\\ \dot{\Omega}_{w,n}\end{bmatrix}=\begin{bmatrix}-\boldsymbol{\omega}\times\Big(\boldsymbol{I}\boldsymbol{\omega}+A\boldsymbol{I}_{\boldsymbol{w}}\Omega_w+A\boldsymbol{I}_{\boldsymbol{w}}A^{\top}\boldsymbol{\omega}\Big)\\ \tau_{w,1}\\ \vdots\\ \tau_{w,n}\end{bmatrix} \tag{5.10}$$

where

$$\boldsymbol{\Gamma}=\begin{bmatrix}\boldsymbol{I}+A\boldsymbol{I}_{\boldsymbol{w}}A^{\top} & \mathbf{a}_1\boldsymbol{I}_{w,1} & \cdots & \mathbf{a}_n\boldsymbol{I}_{w,n}\\ \boldsymbol{I}_{w,1}\mathbf{a}_1^{\top} & \boldsymbol{I}_{w,1} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{I}_{w,n}\mathbf{a}_n^{\top} & 0 & \cdots & \boldsymbol{I}_{w,n}\end{bmatrix}$$

is an augmented inertia coupling matrix for the full system.

5.3. ATTITUDE CONTROL

5.3.1. REACTION WHEEL INNER-LOOP CONTROL

Each wheel torque $\tau_{w,i}$ consists of a motor-provided torque $\tau_{m,i}$ and an undesired friction torque $\tau_{f,i}$

$$\tau_{w,i}=\tau_{m,i}+\tau_{f,i},\qquad i=1,\ldots,n, \tag{5.11}$$

where the friction torque results from static, viscous, Coulomb, an other nonlinear friction torques related to stiction and to extreme conditions of the space environment. The friction torque is estimated with a simple model as

$$\hat{\tau}_{f,i}=M_{\text{vis}}\Omega_{w,i}+M_{\text{Coul}}\,\text{sign}(\Omega_{w,i}),\qquad i=1,\ldots,n, \tag{5.12}$$

where M_{vis} and M_{Coul} are viscous and Coulomb friction parameters, respectively. When no gearboxes are present, and neglecting the dynamics of the DC-motor's electrical current i_c, the relationship between motor current and motor output can be assumed as

$$\tau_{m,i}=\eta_m K_m i_c, \tag{5.13}$$

where η_m and K_m are the motor efficiency and motor constant, respectively. However, to compensate for undesired friction torques $\tau_{f,i}$, a reaction-wheel inner-loop controller

embedded in the actuator and operating at a sampling rate of 100Hz is designed to compensate the effect of undesired and estimated wheel friction torques as a nonlinear function

$$\tau_{m,i} = f_w\left(\tau_{w,i_{cmd}}, \hat{\tau}_{f,i}, \hat{\Omega}_{w,i}, \hat{\Phi}_{w,i}\right) \tag{5.14}$$

which tracks the wheel-torque reference command $\tau_{w,i_{cmd}}$ with the estimated quantities for friction, wheel velocity, and wheel angle. The torque reference command can be related to a desired wheel acceleration whenever wheel-rate control is required by

$$\tau_{w,i_{cmd}} = \hat{\boldsymbol{I}}_{w,i}\dot{\Omega}_{w,i_{des}} \tag{5.15}$$

where $\hat{\boldsymbol{I}}_{w,i}$ is an estimate of the $i-$th wheel inertia. Finally, the $i-$terms $\tau_{w,i_{cmd}}$ are collected on a single vector as

$$\mathbf{u}_w = \begin{bmatrix} \tau_{w,1} \\ \vdots \\ \tau_{w,n} \end{bmatrix}_{cmd}. \tag{5.16}$$

As mentioned in the introduction, BIROS' OBC can only accommodate commands at a sampling rate of 2Hz; therefore, to perform fast slew maneuvers, an outer-loop controller is designed such that it commands the wheel torques in $k-$sampled times as $\mathbf{u}_w = \mathbf{u}_w(k)$ for $k \in \{0, \ldots, N\}$, where N represents the maneuver's final time sample.

5

5.3.2. ATTITUDE AND RATE OUTER-LOOP CONTROL

Analogous to Eq. (5.3), a smooth attitude command $\mathbf{R}_d \in \mathscr{SO}(3)$ satisfying

$$\dot{\mathbf{R}}_d = \mathbf{R}_d \cdot S(\boldsymbol{\omega}_d) \tag{5.17}$$

is considered, where $\boldsymbol{\omega}_d$ is the desired angular velocity assumed to be uniformly bounded. Lee [135] showed that careful selection of an attitude error function can guarantee good tracking performance of nontrivial slew maneuvers involving large initial attitude errors. This is because the magnitude of an attitude error vector should be proportional to the rotation about the Euler-axis between the current and desired attitude. In this sense, an attitude error function $\Psi : \mathscr{SO}(3) \times \mathscr{SO}(3) \mapsto \mathscr{R}$, as in Goodarzi et al., [169] is selected such that

$$\Psi(\mathbf{R}, \mathbf{R}_d) = \frac{1}{2}\mathrm{tr}\left(I - \mathbf{R}_d^\top \mathbf{R}\right), \tag{5.18}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. With this choice, an attitude error vector $\boldsymbol{e}_R \in \mathscr{R}^3$ and angular velocity error vector $\boldsymbol{e}_\omega \in \mathscr{R}^3$ can be defined as

$$\boldsymbol{e}_R = \frac{1}{2}(\mathbf{R}_d^\top \mathbf{R} - \mathbf{R}^\top \mathbf{R}_d)^\vee, \tag{5.19}$$

$$\boldsymbol{e}_\omega = \boldsymbol{\omega} - \mathbf{R}^\top \mathbf{R}_d\, \boldsymbol{\omega}_d, \tag{5.20}$$

recalling that $\vee$ denotes the vee map as defined in Section 5.2. Note that the magnitude of the dimensionless attitude error vector is bounded [135] as $0 \leq |\boldsymbol{e}_R| \leq 1$. The sampled-time tracking error state $\boldsymbol{x}_{\boldsymbol{e}}(k) \in \mathscr{R}^6$ is defined as

$$\boldsymbol{x}_e(k) = \begin{bmatrix} \boldsymbol{e}_R(k) \\ \boldsymbol{e}_\omega(k) \end{bmatrix} \tag{5.21}$$

and the objective is therefore to design an attitude control law having $\boldsymbol{x_e} \to 0$ as $k \to N$. This means that $\boldsymbol{x}_e = 0$ if and only if $\mathbf{R} = \mathbf{R}_d$ and therefore $\boldsymbol{\omega} = \mathbf{R}^\top \mathbf{R}_d \, \boldsymbol{\omega}_d = \boldsymbol{\omega}_d$. Sampled-time nonlinear attitude control is given by a combination of feedback and feedforward control laws

$$\mathbf{u}_w(k) = \mathbf{u}_{FB}(k) + \mathbf{u}_{FF}(k), \tag{5.22}$$

where $\mathbf{u}_{FB}$ can be the discrete version of the geometric PID attitude controller proposed in Goodarzi et al. [169] without the feedforward terms. In the next section, the feedforward commands $\mathbf{u}_{FF}(k)$ obtained as solutions of time-optimal control problems are described.

5.4. OPTIMAL GUIDANCE

In this section, a methodology is presented for the generation of offline fast slew maneuvers as solutions of time-optimal control problems. The solutions serve as the basis for the attitude control system where they will be implemented as the feedforward control commands $\mathbf{u}_{FF}(k)$ in sampled-time.

5.4.1. TIME-OPTIMAL SLEW MANEUVER PROBLEM FORMULATION

The objective of time-optimal slew maneuver problems[134, 138] consists on finding optimal wheel-motor torque commands $\tau_{w,i}$ $(i = 1, \ldots, n)$ that transfer any given initial attitude $\mathbf{R}(t_0)$, angular velocity $\boldsymbol{\omega}(t_0)$, and wheel speed $\Omega_w(t_0)$ of the rigid body to a desired final attitude $\mathbf{R}(t_f)$, angular velocity $\boldsymbol{\omega}(t_f)$, and wheel speed $\Omega_w(t_f)$ within a *minimal time* t_f. Such time-optimal maneuvers can be mathematically formulated as the following optimization problem

$$\underset{\tau_{w,i},\ (i=1,\ldots,n)}{\text{minimize}} \left\{ J = \int_{t_0}^{t_f} 1 \mathrm{d}t \right\}, \tag{5.23a}$$

subject to the dynamic Eqs. (5.3) and (10.15), $\forall\, t \in [t_0, t_f]$,

$$\begin{aligned} \text{such that } \mathbf{R}(t_0) &= \mathbf{R}_0, \\ \mathbf{R}(t_f) &= \mathbf{R}_f, \\ \boldsymbol{\omega}(t_0) &= \boldsymbol{\omega}_0, \\ \boldsymbol{\omega}(t_f) &= \boldsymbol{\omega}_f, \\ \Omega_w(t_0) &= \Omega_{w0}, \\ \Omega_w(t_f) &= \Omega_{wf}, \end{aligned}$$

with

$$\left\| \tau_{w,i}(t) \right\| \leq \tau_{w,i_{\max}},\ (i = 1, \ldots, n),\ \forall\, t \in [t_0, t_f]. \tag{5.23b}$$

Without loss of generality, only *rest-to-rest* maneuvers are considered in this work, where initial and final angular velocities are directly imposed to be zero:

$$\boldsymbol{\omega}(t_0) = \boldsymbol{\omega}(t_f) = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^\top \text{rad/s}. \tag{5.24}$$

Moreover, in the remainder of this chapter, initial *HTW* speeds are also considered to be zero $\Omega_{w,i}(t_0) = 0,\ i = 1,2,3$, and their final wheel speed is set to be free. The remaining *RW-90* wheels, $i = 4,\ldots,7$, are set-point regulated according to their initial values with a simple proportional control law as

$$\dot{\Omega}_{w,i} = -k_p\Big[\Omega_{w,i} - \Omega_{w,i}(t_0)\Big],\ k_p = 1\times 10^{-4} \tag{5.25}$$

giving rise to non-cooperating angular momentum for the slew maneuvers. Although it has already been mentioned that time-optimal maneuvers are, in general, not Euler-axis rotations whenever the actuators can be saturated independently, it is not straightforward to conclude whether or not a local solution of this problem corresponds to a global solution.

5.4.2. TRANSCRIPTION OF THE TIME-OPTIMAL SLEW MANEUVER PROBLEM FORMULATION INTO A DIRECT APPROACH

Because the problem formulation of time-optimal slew maneuvers does not involve a prescribed path to be followed a-priori, it can be considered a trajectory optimization problem that minimizes the total maneuver time according to the set of constraints presented.

In this sense, the trajectory optimization problem consists of transcribing the time-optimal control problem into a constrained parameter optimization problem and solving it with a direct approach using DLR's *Trajectory Optimization Package* [2] *'trajOpt'*, that is included in the *MOPS* software environment[52–54] and implemented in MATLAB[170], which solves multi-objective design problems that are mapped to weighted *min-max* optimization problems. *MOPS* is a versatile tool widely used in the aeronautical community[52–59] to support many aspects of general control design processes, such as multi-model and multi-case design problems, robust tuning via Monte-Carlo simulations, control law robustness assessment, worst-case analysis, and parameter estimation. A key advantage of using the trajectory optimization package *trajOpt/MOPS*, originally designed to solve hybrid multi-phase trajectory optimization problems for launch vehicles, is that boundary conditions at the beginning and end of each phase of the desired maneuvers are considered in an efficient way.

Transcription of the original constrained minimization problem into a direct approach consists on defining the original k design objectives mathematically as positive criteria c_k to be minimized against demanded values d_k, and considering the following *min-max* multi-criteria optimization problem, which is the *MOPS* synthesis[53, 54, 56] formula

$$\min_{\mathscr{T}}\left\{\max_{k\in\mathscr{S}_m}\left\{\frac{c_k(\mathscr{T})}{d_k}\right\}\right\}, \tag{5.26a}$$

$$\begin{aligned}\text{subject to } c_k(\mathscr{T}) &= d_k, \quad k\in\mathscr{S}_{\text{eq}},\\ c_k(\mathscr{T}) &\le d_k, \quad k\in\mathscr{S}_{\text{ineq}},\end{aligned}$$

$$\text{with}$$

$$\mathscr{T}_{\min,l} \le \mathscr{T}_l \le \mathscr{T}_{\max,l},\ \text{over } [0,t_f]. \tag{5.26b}$$

Here, $\mathscr{S}_m$ is the index set of criteria to be minimized, $\mathscr{S}_{\text{eq}}$ is the index set of equality constraints and $\mathscr{S}_{\text{ineq}}$ is the index set of inequality constraints; $\mathscr{T}$ is the vector containing the tuning parameters $\mathscr{T}_l$ to be optimized, which lies in between upper and lower bounds $\mathscr{T}_{\min,l}$ and $\mathscr{T}_{\max,l}$, respectively; $c_k\,(k \in \mathscr{S}_m)$ is the $k-$th normalized criterion; $d_k\,(k \in \mathscr{S}_m)$ is the corresponding demand value which serves as a criterion weight; $c_k\,(k \in \mathscr{S}_{\text{eq}}, \mathscr{S}_{\text{ineq}})$ is normalized criteria used as equality or inequality constraints, respectively; and lastly, $d_k\,(k \in \mathscr{S}_{\text{eq}}, \mathscr{S}_{\text{ineq}})$ is the corresponding demand value. Finally, the newly formulated multi-criteria optimization problem in Eq. (5.26) can be solved using standard *nonlinear programming* (NLP) methods for the objective function with equality and inequality constraints.

5.4.3. METHODOLOGY TO OBTAIN PIECEWISE-CONSTANT SAMPLED-TIME OPTIMAL MANEUVERS

For the main objective of this chapter, which is to design fast slew rest-to-rest maneuvers for BIROS' *HTW* experiments with piecewise-constant sampled-time inputs as feedforward control commands, a methodology is presented. It consists of an iterative procedure that finds solutions to three consecutive problems that are solved using the direct approach previously outlined. Table 5.3 presents the criteria c_k, demands d_k, and tuners $\mathscr{T}$ used for designing the maneuvers considered in this iterative procedure. Note that criteria scaling[59] can be performed by dividing each criterion by the value demanded $\hat{c}_k(\mathscr{T}) = c_k(\mathscr{T})/d_k$, and therefore, the value demanded for minimizing the slew time is set to $d_1 = 1$s. The three consecutive problems to be solved are described in detail as follows.

Problem I First, the criteria $c_1 - c_3$ are used together with their demands $d_1 - d_3$ and tuners $\mathscr{T}_1$ and $\mathscr{T}_2$ to obtain a candidate minimum maneuver time t_f. Here, the input control commands are interpolated with *piecewise cubic Hermite interpolating polynomials ('pchip')* available in the *trajOpt* package in order to obtain a smooth solution for these inputs. The optimal slew time t_f is approximated towards a new demanded fixed-time t_f^*, which must be a multiple of the desired frequency of 2Hz. Additionally, the optimal control inputs are re-sampled at this frequency since they are meant to be used as initial guesses for the subsequent optimization problem. With the solution of this problem, an insight can be obtained not only on the minimum time required to complete the maneuver, but also on the maneuver itself since these can be compared, for instance, to Euler-axis rotations which are generally not time-optimal.

Problem II Here, the interest is towards *fixed-time* solutions for the same problem setup as before, but considering sampled-time control inputs at the sampling rate of 2Hz. The new demanded fixed-time t_f^* and the initial guess for the solution are obtained as described in the previous problem. This problem is solved considering criteria $c_2 - c_3$ together with demands $d_2 - d_3$ and tuner $\mathscr{T}_2$. In this case, the inputs are obtained as piecewise-linear control commands in order to obtain a sampled-time solution close to the previous one. Once finished, these piecewise-linear solutions

are interpolated with a mid-point rule in order to be considered as initial guesses for the next and final optimization problem.

Problem III Here, the criteria $c_2 - c_3$ are again considered together with demands $d_2 - d_3$ and tuner $\mathcal{T}_2$. The goal is to find *piecewise-constant* control inputs for the original problem within the minimum fixed-time t_f^* approximation obtained before, which represents the final goal of this procedure. The initial guesses obtained from the piecewise-linear inputs of the previous problem are of great help for final optimization since the resulting sampled-time piecewise-constant control inputs are, in general, already sufficiently close to the optimal desired solution.

Table 5.3: Design criteria c_k, demands d_k, and tuners $\mathcal{T}_k$ used for the design of fast slew maneuvers with *trajOpt/MOPS*.

Criteria c_k		
n°	Criteria specification	Description
c_1	Minimum slew time t_f	t_f
c_2	Final attitude error $\boldsymbol{e}_R(t_f)$	$\lvert\, \boldsymbol{e}_R(t_f) \,\rvert$
c_3	Final angular velocity error $\boldsymbol{e}_\omega(t_f)$	$\lvert\, \boldsymbol{e}_\omega(t_f) \,\rvert$
Demands d_k		
n°	Demands	Value
d_1	Slew time t_f	1 [s]
d_2	Final attitude error $\boldsymbol{e}_R(t_f)$	$\leq 1 \times 10^{-7}$ [$-$]
d_3	Final angular velocity error $\boldsymbol{e}_\omega(t_f)$	$\leq 1 \times 10^{-5}$ [rad/s]
Tuners $\mathcal{T}_k$		
n°	Tuner	Description
$\mathcal{T}_1$	Slew time t_f	t_f
$\mathcal{T}_2$	*HTW* torque commands	$\tau_{w,i}$ $(i = 1,2,3)$

Figure 2 presents a diagram of the steps involved in these three consecutive problems. Whenever one of these problems fail to give a feasible solution, a new iteration process is required where the criteria and their demands are re-evaluated. For instance, if no feasible solution for Problem II is found, a good starting point is reconsidering the fixed-time for this problem to be one sample higher, giving an extra control command for the potential new solution. This process is repeated until a satisfactory outcome is achieved.

5.5. SIMULATION

For numerical simulations using the comprehensive analytical nonlinear model of Section 2, the *High-Torque-Wheels* BIROS satellite is considered with an approximated iner-

tia matrix of

$$\boldsymbol{I} = \text{diag}\begin{bmatrix} 9 & 6 & 9 \end{bmatrix} \text{Kg}\cdot\text{m}^2.$$

The time-optimal rest-to-rest maneuver is designed to achieve the desired final attitude described by the $(3-2-1)$ Euler angles ψ, θ, and ϕ, i.e. $\mathbf{R}(t_f) = \mathbf{R}(\phi(t_f), \theta(t_f), \psi(t_f))$, where $\phi(t_f) = 0\text{deg}$, $\theta(t_f) = 5\text{deg}$, and $\psi(t_f) = 30\text{deg}$, respectively. This translates into the following initial and final (objective) attitudes in terms of rotation matrices

$$\mathbf{R}(t_0) = \mathscr{I}_{3\times 3}, \quad \mathbf{R}(t_f) = \begin{bmatrix} 0.8627 & 0.4981 & -0.0872 \\ -0.5000 & 0.8660 & 0 \\ 0.0755 & 0.0436 & 0.9962 \end{bmatrix}.$$

It is important to mention that time-dependent attitude paths like Euler-axis rotations or Euler angles are not specified a-priori, giving the optimization solver the possibility to find a time optimal path dynamically. The initial *HTW* wheel speeds are zero since the experiments consider these wheels only for agile reorientation; while the initial *RW-90* wheel speeds are set to $\Omega_{w,i}(t_0) = -200\text{rad/s}$ $(i = 4,\ldots,7)$ to simulate a realistic scenario of initial angular momentum stored in the platform. The final *HTW* and *RW-90* wheel speeds are set to be free; but actually, the final state of the latter set of wheels depends on the performance of the wheel-controller in Eq. (5.25) during the maneuver. Lastly, the nominal values presented in Table 10.1 are considered as the actuator limits to allow some margin in case the wheels must be saturated by their inner-loop controls.

The simulation results are as follows. Figure 10.1 presents the torque command solutions using the methodology described in Section 4.3, where the three consecutive optimal control solutions are denoted as $\tau_{w,I}, \tau_{w,II}$, and $\tau_{w,III}$ for each problem I, II, and III, respectively. The solution of problem I gives $t_f = 9.43\text{s}$, so the new demanded fixed-time for problems II and III is set to $t_f^* = 9.5\text{s}$ (first multiple of 2Hz after t_f). For the optimal control inputs obtained, Fig. 5.4 presents the simulation results for attitude errors, angular velocities, and reaction wheel speeds, respectively. These results show that the maneuvers are almost identical for the three consecutive solutions and only differ in the input solutions due to the parameterization of the torque commands. Finding time-optimal maneuvers departing with piecewise-constant control inputs is, in general, a very challenging and time consuming task. Using the three-step methodology presented in this study, the final goal of obtaining fast slew maneuver solutions with piecewise-constant control commands is achieved much more efficiently.

5.6. CONCLUSIONS AND OUTLOOK

The objective of this chapter was to investigate a high-agility attitude control system by finding a methodology to design time-optimal slew maneuvers for BIROS' *High-Torque-Wheels* experiments. This is achieved by considering a comprehensive analytical nonlinear model for spacecraft equipped with reaction wheels and formulating the problem as a constrained nonlinear optimal control problem including both satellite's continuous-time dynamics and piecewise-constant sampled-time control inputs.

A methodology that utilizes three consecutive multi-criteria optimization problems is proposed to obtain solutions via a direct approach that applies the *'trajOpt'* trajectory

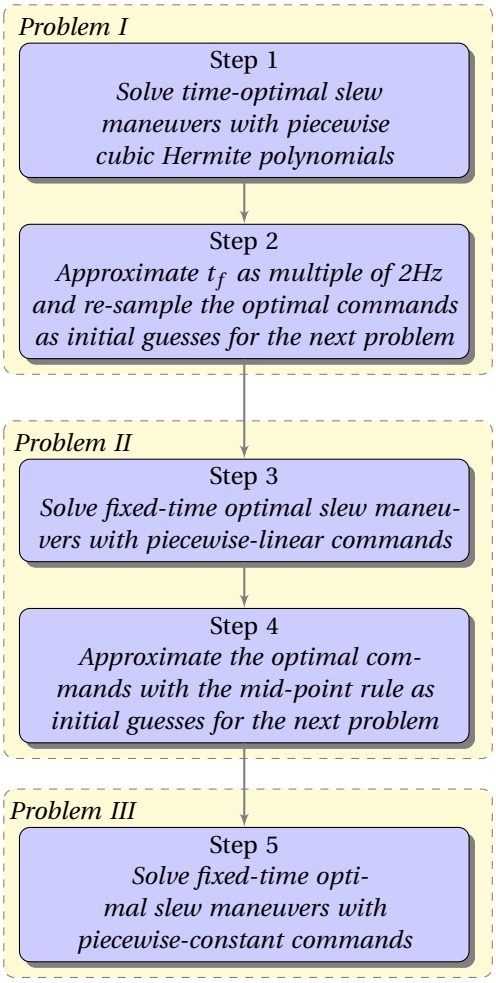


Figure 5.2: Diagram of the sequential three-step procedure to obtain fast slew maneuvers with piecewise-constant control commands.

optimization package of the DLR-SR optimization tool, *Multi-Objective Parameter Synthesis (MOPS)*. Results based on numerical simulations performed with the nonlinear spacecraft dynamics model were presented.

Hardware-in-the-loop simulations will be used to validate the attitude control system with a three-axis air-bearing testbed featuring the BIROS engineering model. Once tested, the experiment can be implemented in the '*Fast Slew*' mode of the BIROS attitude control system for in-orbit tests.

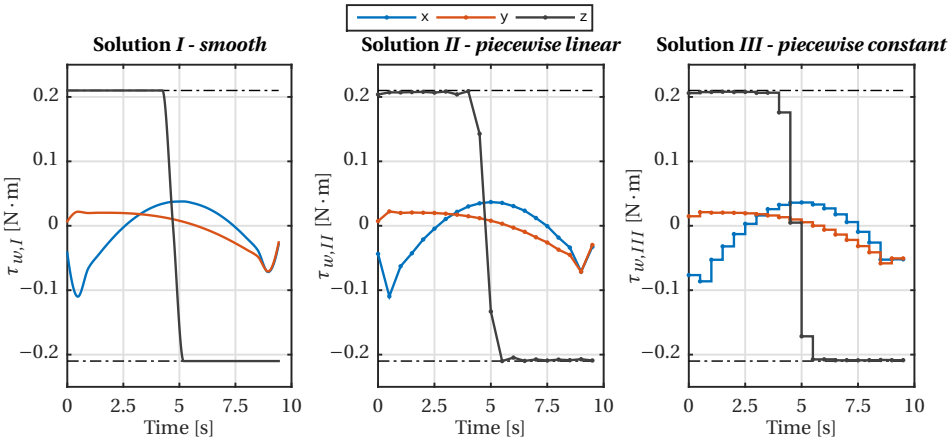


Figure 5.3: Torque command results using the sequential methodology to obtain sampled-time fast slew maneuvers; *I*) first solution finding the minimum time with smooth control inputs; *II*) second solution with fixed-time and piecewise-linear control inputs; and *III*) final solution of the original problem with fixed-time and piecewise-constant control inputs.

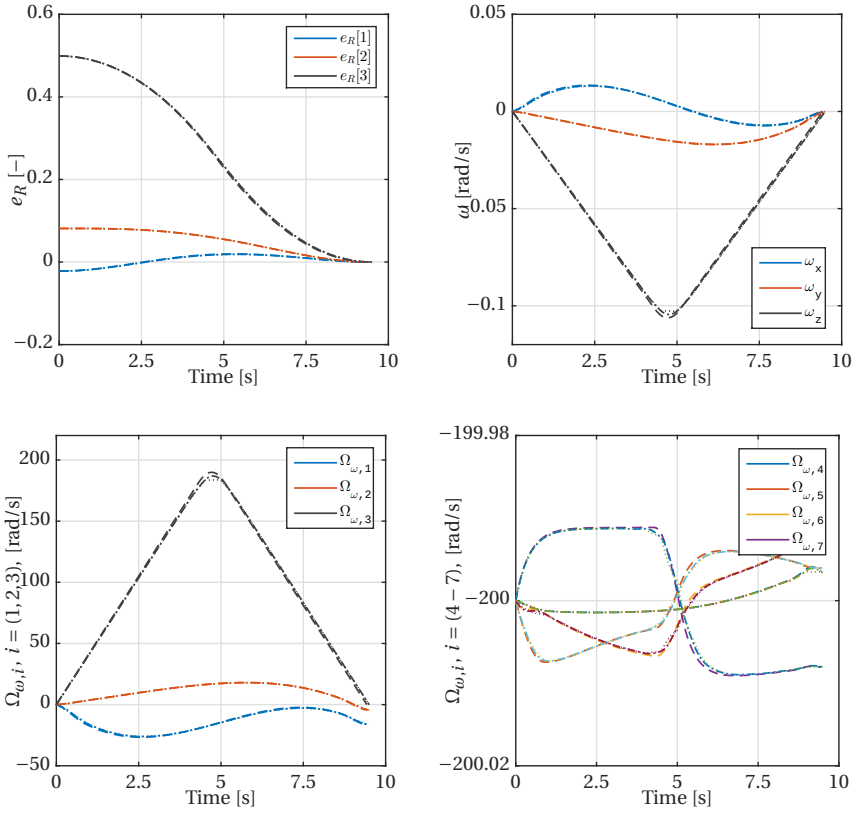


Figure 5.4: Simulation results for the attitude error, angular velocity, and reaction wheel speeds, respectively; using the optimal control inputs obtained with solution *I* (– – –), solution *II* (·····), and solution *III* (—·—).

III

Robust Nonlinear Attitude Control

6

Incremental Backstepping for Robust Nonlinear Flight Control

Abstract

This paper presents a robust nonlinear flight control strategy based on results combining incremental control action and the backstepping design methodology for vehicles described by strict-feedback (cascaded) nonlinear systems. The approach, referred to as incremental backstepping, uses feedback of actuator states and acceleration estimates to allow the design of increments of control action. In combination with backstepping, the proposed approach stabilizes or tracks outer-loop control variables of the nonlinear system incrementally, accounting for large model and parametric uncertainties, besides undesired factors such as external perturbations and aerodynamic modeling errors. With this result, dependency on the modeled aircraft system is greatly reduced, overcoming the major robustness flaw of conventional model-based flight control strategies. This suggested methodology implies a trade-off between accurate knowledge of the dynamic model and accurate knowledge of the vehicle sensors and actuators, which makes it more suitable for practical application than identification or model based adaptive control architectures. Simulation results verify the tracking capability and superior robustness of the proposed controller under aerodynamic uncertainty with respect to standard backstepping methodologies for a simple flight control example.

Publication

Paul Acquatella B., Erik-Jan van Kampen, Qi Ping Chu: *Incremental Backstepping for Robust Nonlinear Flight Control.* In: Proceedings of EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control, April 10-12, 2013. Delft, The Netherlands.

6.1. INTRODUCTION

THE DESIGN of a generic robust nonlinear flight control strategy is considered in this chapter. The strategy is based on recent results combining incremental control action and the backstepping design methodology for strict-feedback (cascaded) nonlinear systems, called incremental backstepping. The main design issue is dealing with large model and parametric uncertainties present in flight control systems, mainly because of aerodynamic and unmodeled dynamics.

Incremental backstepping is presented by means of a modification to the standard backstepping design methodology that reduces its dependency on the baseline aircraft model, through the use of actuator states and acceleration estimates. These considerations allow the design of increments of control action which, in combination with backstepping, helps to stabilize or track outer-loop control variables of the nonlinear system incrementally. In contrast to regular backstepping, this method is inherently *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be canceled, but which results when the feedback loops are closed.

Theoretical development of increments of nonlinear control action date back from the late nineties and started with activities concerning 'Implicit Dynamic Inversion' for DI-based flight control [70, 73], where the architectures considered in this chapter were firstly described. Other designations for these developments found in the literature are 'Modified NDI' and 'Simplified NDI', but the designation 'Incremental NDI' is considered to describe the methodology and nature of these type of control laws better [71, 77–79]. INDI has been elaborated and applied theoretically in the past decade for flight control and space applications [4, 71–75].

The main motivation of this approach is to bring the implicitness of such sensor-based architectures with Lyapunov-based controller design such as backstepping for aerospace applications. This topic has been introduced in the literature recently, but from a singular perturbations approach, in [100]. The recursive step-by-step procedure of the backstepping methodology can be exploited for the design of a single and generic control law for cascaded systems, retaining by definition its stability and convergence properties, and with the possibility to retain stabilizing nonlinearities in the closed-loop system description.

The remainder of the chapter is organized as follows. Section 6.2 presents the main results of this chapter, namely the incremental backstepping approach. In Section 6.3 we present the generic flight control law design with this method and for the particular case of attitude control. Section 6.5 illustrates the design of incremental backstepping control for an exemplary longitudinal missile tracking control, including simulations of such control strategy. Conclusions are provided in Section 6.6.

6.2. INCREMENTAL BACKSTEPPING

This section presents the proposed incremental backstepping approach. Its design departure is from a stability and convergence viewpoint due to *control Lyapunov function* augmentations rather than forcing linear behaviour through conventional feedback linearization. Because of its advantage of stabilizing or tracking one or more loops within a single control command maintaining desired properties, the motivation for this ap-

proach also stems to the combined flexibility of this method over conventional approaches such as robust nonlinear dynamic inversion (NDI) [171–180], and its adaptive [68, 181, 182] and incremental counterparts [4, 70–75, 77].

For the discussion, we will consider physical systems or vehicle dynamics which are represented by the following strict-feedback second order cascaded form:

$$\dot{\xi} = \mathbf{h}(\xi) + \mathbf{k}(\xi)\mathbf{x} \tag{6.1a}$$

$$\dot{\mathbf{x}} = \mathbf{f}(\xi, \mathbf{x}) + \mathbf{G}(\xi, \mathbf{x})\mathbf{u} \tag{6.1b}$$

We assume that Eq. (6.1a) may represent a kinematic equation, i.e., a relation between (angular) velocities and positions (orientations), while Eq. (6.1b) may represent a dynamic equation relating forces and torques to the former (angular) velocities, see Figure 6.1. In flight control, Eq. (6.1a) may also have a control input dependency, if not always, but this term is ignored during the control design of the kinematic loop since the backstepping method can only handle nonlinear systems of lower-triangular form (e.g., for attitude control the assumption is made that the fin surface is a pure moment generator). Although this method is presented for second-order strict feedback (cascaded) nonlinear systems, its extension to higher-order systems by continuation of the backstepping design methodology is straightforward. This is of particular interest, if for instance, several control loops are to be considered for the control law design (e.g., position control, etc.). The closed-loop stability of the complete system for this cascaded in-

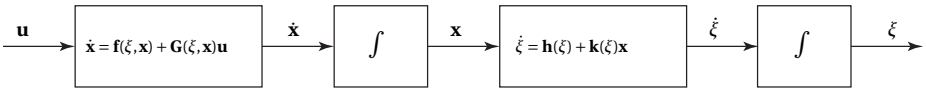


Figure 6.1: Cascade structure of the system in Eqs. (6.1).

terconnection will rely on the efficient design of a control law $\mathbf{u}$. We start the discussion with a brief review of the backstepping (denoted 'BKS') procedure [65, 76] for stabilization, in this case as follows:

STEP 1

1. Promoting $\mathbf{x}$ as the virtual control in Eq. (6.1a), introduce the error state as:

$$\mathbf{z} = \mathbf{x} - \mathbf{x}_{\text{des}} = \mathbf{x} - \alpha(\xi)$$

where $\alpha(\xi)$ is a stabilizing feedback that will be designed in the following substeps. Such intermediate control law is referred as a *stabilizing function*. Rewriting Eq. (6.1a) in terms of this error state results in:

$$\dot{\xi} = \mathbf{h}(\xi) + \mathbf{k}(\xi)(\mathbf{z} + \alpha)$$

2. Construct any positive definite, radially unbounded function $V_1(\xi) : \mathscr{R}^3 \mapsto \mathscr{R}^+$ as a *control Lyapunov function* (CLF) for the system, treating it as a final stage, e.g.,

$$V_1(\xi) = \frac{1}{2}\xi^\top \xi$$

This choice of a CLF may depend on the kinematic equation considered and may trade-off its complexity with the resulting control law.

3. To find a stabilizing function $\alpha(\xi)$ for the virtual control in this step ($\mathbf{x}$), we need to make the derivative of $V_1(\xi)$ nonpositive when $\mathbf{x} = \alpha$. Such continuously differentiable feedback control law $\alpha(\xi)$ hence need to satisfy:

$$\dot{V}_1 = \frac{\partial V_1(\xi)}{\partial \xi}\Big[\mathbf{h}(\xi) + \mathbf{k}(\xi)\alpha(\xi)\Big] \leq -W(\xi) \leq 0, \quad \forall \xi \in \mathscr{R}^n$$

where $W : \mathscr{R}^n \mapsto \mathscr{R}$ is positive semi-definite. Moreover, for the subsequent steps, the following notation for the derivative of the current stabilizing function $\alpha(\xi)$ is introduced:

$$\dot{\alpha}(\xi, \mathbf{x}) = \frac{\partial \alpha(\xi)}{\partial \xi}\dot{\xi} = \frac{\partial \alpha(\xi)}{\partial \xi}\Big[\mathbf{h}(\xi) + \mathbf{k}(\xi)\Big(\mathbf{z} + \alpha(\xi)\Big)\Big]$$

STEP 2

This step consists of calculating the final control law $\mathbf{u}$ as follows.

1. With $\alpha(\xi)$ determined, the next step is to consider the subsequent state equation, the dynamics in Eq. (6.1b), in terms of the error state:

$$\dot{\mathbf{z}} = \dot{\mathbf{x}} - \dot{\alpha}(\xi, \mathbf{x}) = \mathbf{f}(\xi, \mathbf{x}) + \mathbf{G}(\xi, \mathbf{x})\mathbf{u} - \dot{\alpha}(\xi, \mathbf{x})$$

2. Construct an augmented CLF for the system, treating it as a final stage:

$$V_2(\xi, \mathbf{x}) = V_1 + \frac{1}{2}\mathbf{z}^\top \mathbf{z}$$

3. To find the final control law $\mathbf{u}$ in this step, we need to make the derivative of $V_2(\xi, \mathbf{x})$ nonpositive when $\xi \neq \alpha$

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \mathbf{z}^\top \dot{\mathbf{z}} \\ &\leq -W(\xi) + \mathbf{z}^\top \Big[\mathbf{f}(\xi, \mathbf{x}) + \mathbf{G}(\xi, \mathbf{x})\mathbf{u} - \dot{\alpha}(\xi, \mathbf{x}) + \frac{\partial V_1(\xi)}{\partial \xi}\mathbf{k}(\xi)\Big]\end{aligned}$$

If $\mathbf{G}(\xi, \mathbf{x}) \neq 0$ and invertible for all $\mathbf{x}$ and ξ, *one* possible choice for $\mathbf{u}$ is:

$$\mathbf{u} = \mathbf{G}^{-1}(\xi, \mathbf{x})\Big[-c_1\mathbf{z} - \mathbf{f}(\mathbf{x}) + \dot{\alpha}(\xi, \mathbf{x}) - \frac{\partial V_1(\xi)}{\partial \xi}\mathbf{k}(\xi)\Big] \tag{6.2}$$

with $c_1 > 0$, which yields $\dot{V}_2 \leq -W(\xi) - c_1\mathbf{z}^\top \mathbf{z} \leq 0$. However, as we pointed out before, many other, possibly better, choices for α could be available, even if $\mathbf{G}(\xi, \mathbf{x}) = 0$ at some points.

It should be clear that this result of backstepping for cascaded second order systems is not the specific form of the control law (6.2), but rather the construction of a stabilizing function for the kinematic equation that depends on the choice of a Lyapunov function

whose derivative can be made negative by a wide variety of family of control laws. Also, the augmentation of this selected Lyapunov function in the second step may have other structure, which could result in a different family of controllers. This flexibility in backstepping gives a great advantage to the control engineer, in which the complexity of the CLFs can be traded with the complexity of the resulting controller structure. This backstepping procedure can be illustrated as in Figure 6.2.

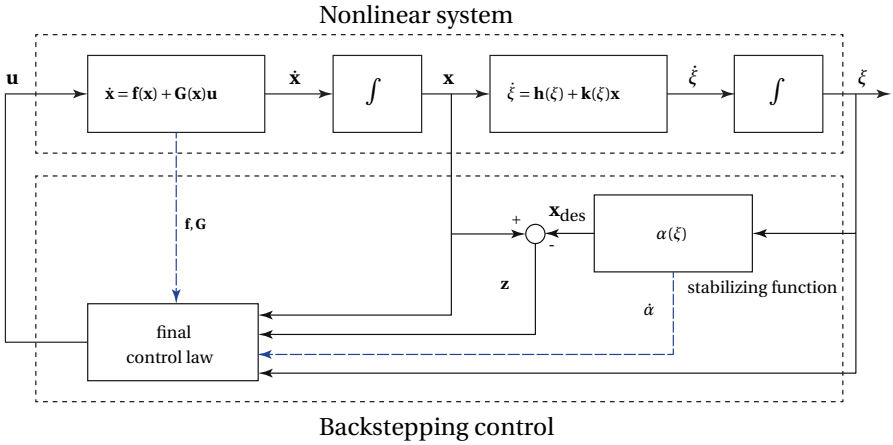


Figure 6.2: Backstepping control block diagram for second order cascaded systems. Dashed arrows represent information required for control design. Notice that the final control law requires knowledge of both **f** and **G**.

The incremental backstepping (denoted 'IBKS') is derived from expressing or approximating the dynamics into an incremental form. This incremental form of the dynamic equation is obtained as follows [73]. Consider a generic form of an affine nonlinear dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \tag{6.3}$$

where $\mathbf{x} \in \mathscr{R}^n$ is the state vector, $\mathbf{u} \in \mathscr{R}^m$ is the control input vector, **f** and **h** are smooth vector fields on $\mathscr{R}^n$, and $\mathbf{G} \in \mathscr{R}^{n\times m}$ is a matrix whose columns are smooth vector fields $\mathbf{g}_j$. A standard Taylor series expansion provides the following first-order approximation of $\dot{\mathbf{x}}$ for $\mathbf{x}$ and $\boldsymbol{\delta}$ in the neighborhood of $[\mathbf{x}_0, \mathbf{u}_0]$:

$$\dot{\mathbf{x}} \cong \mathbf{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 + \frac{\partial}{\partial \mathbf{x}}\left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{x}-\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)(\mathbf{u}-\mathbf{u}_0) + H.O.T. \tag{6.4}$$

where the current state and control, $\mathbf{x}_0$ and $\mathbf{u}_0$ respectively, represent for each time instance the *reference* an incremental instance in time before $\mathbf{x}$ and $\mathbf{u}$ for the construction of the first-order approximation of $\dot{\mathbf{x}}$, and $H.O.T.$ the higher order terms that can be neglected. By definition, the corresponding state derivative $\dot{\mathbf{x}}_0$ satisfies:

$$\dot{\mathbf{x}}_0 \equiv \mathbf{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 \tag{6.5}$$

Using this expression and the standard linear definition,

$$\mathbf{A}_0 = \frac{\partial}{\partial \mathbf{x}} \left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} \tag{6.6a}$$

$$\mathbf{B}_0 = \frac{\partial}{\partial \mathbf{u}} \left[\mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}} = \mathbf{G}(\mathbf{x}_0) \tag{6.6b}$$

with $\mathbf{A}_0$ and $\mathbf{B}_0$ being the partials evaluated at the current reference point $[\mathbf{x}_0, \mathbf{u}_0]$ on the state/control trajectory; Equation (6.4), i.e., the approximation of $\dot{\mathbf{x}}$ for $\mathbf{x}$ and $\mathbf{u}$ in the neighborhood of $[\mathbf{x}_0, \mathbf{u}_0]$ can be written as:

$$\dot{\mathbf{x}} \cong \dot{\mathbf{x}}_0 + \mathbf{A}_0 (\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0 \Delta\mathbf{u} \tag{6.7}$$

where $\Delta\mathbf{u} = (\mathbf{u} - \mathbf{u}_0)$ represents the incremental control command. This suggests that in a small neighborhood of the reference state we can approximate the nonlinear system (6.3) by its linearization about that reference state.

Considering this linear approximation in the second step of the backstepping procedure presented, we obtain the following control law for the increments of nonlinear control:

$$\Delta\mathbf{u} = \mathbf{G}^{-1}(\mathbf{x}_0) \left[-c_1\mathbf{z} - \dot{\mathbf{x}}_0 - \mathbf{A}_0 (\mathbf{x} - \mathbf{x}_0) + \dot{\alpha} - \frac{\partial V_1(\xi)}{\partial \xi}\mathbf{k}(\xi)\right] \tag{6.8}$$

Moreover, considering small time increments and a sufficiently high control update rate, $\mathbf{x}$ approaches $\mathbf{x}_0$ much faster than an incremental change of the dynamics due to an incremental input, hence the incremental backstepping control law becomes:

$$\Delta\mathbf{u} = \mathbf{G}^{-1}(\mathbf{x}_0) \left[-c_1\mathbf{z} - \dot{\mathbf{x}}_0 + \dot{\alpha} - \frac{\partial V_1(\xi)}{\partial \xi}\mathbf{k}(\xi)\right] \tag{6.9}$$

This control ensures $\mathbf{z}$ to be uniformly ultimately bounded. Note that this control law results in increments of control commands; these changes must be added to the current reference command to obtain the full new control command input. Hence, the total control command is obtained as:

$$\mathbf{u} = \mathbf{u}_0 + \Delta\mathbf{u} \tag{6.10}$$

The incremental backstepping control law (6.10), as the application of backstepping to a system expressed in an incremental form, results in a control law that is not depending on the plant dynamics $\mathbf{f}(\mathbf{x})$ explicitly. This results in a *implicit*-control approach where the dependency of $\mathbf{f}(\mathbf{x})$ of the closed-loop system under feedback control is largely decreased, improving the system robustness against model mismatch and model uncertainties. Remaining dependency is due to changes in $\mathbf{f}(\mathbf{x})$ that are reflected in $\dot{\mathbf{x}}_0$, and since the control approach does require estimates of $\dot{\mathbf{x}}_0$ and $\mathbf{u}_0$, the control strategy is more sensor/actuator dependent. Moreover, apart from the aspects considered, the control needs as well the vehicle control derivatives $\mathbf{G}(\mathbf{x}_0)$. To make a clear difference with respect to standard (Jacobian) linearization over *operating* points, a graphical interpretation of the implicit nature of increments of control is depicted in Figure 6.3-(c). The incremental backstepping block diagram is illustrated in Figure 6.4.

The implementation of incremental-based controllers considers the following assumptions:

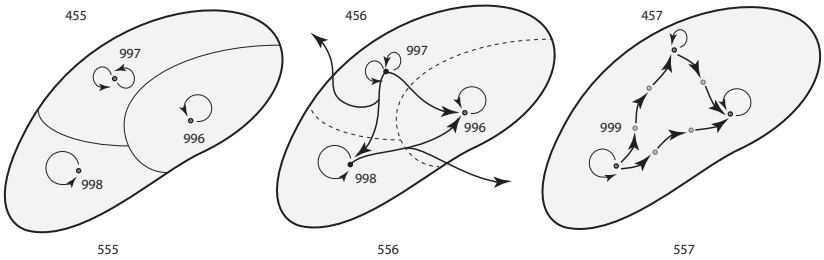


Figure 6.3: Graphical interpretation of three control strategies: (a) some linear controllers designed over some operating points by standard (Jacobian) linearization of the system; (b) the concept of gain-scheduling between these operating points, where stability and convergence are not guaranteed overall; (c) the implicit nature of increments of control action, the current state represents a new reference and the control strategy acts stabilizing or tracking incrementally, and without the need of scheduling or the design of multiple controllers.

(i) It is assumed to have complete and accurate knowledge about the state of the system. State derivatives (acceleration) sensors are considered to be available for this study as well. In the case of angular acceleration measurements, they may be measured directly or derived by differentiation from inertial measurement unit (IMU) gyro measurements and filtered accordingly;

(ii) For small time increments, state derivatives evolve faster than the state upon fast control action, which directly influences the dynamics of the rigid body. In other words, the state only change by integrating state derivatives, hence making the difference $(\mathbf{x}-\mathbf{x}_0)$ negligible for small time increments as compared to $\dot{\mathbf{x}}$;

(iii) Fast control action is assumed. This assumption complements the previous one in the sense that the dynamics of the actuators are considered to evolve much faster than the states. For this study a linear second order dynamics for the actuators is assumed, and considering an actuator undamped natural frequency ω_{n_c} sufficiently high guarantees the fast actuator requirement of incremental control action.

Regarding the actuator state requirement, Fig.6.5-(a) illustrates a sensor-dependency configuration, where the actuator state measurements are readily available (e.g. known current surface deflection), and Fig.6.5-(b) illustrates the model-dependent approach, where actuator state measurements are not readily available and a high-fidelity model of actuator dynamics are to be included in the control architecture as to supply the required control input reference $\mathbf{u}_0$. The mismatch of such measurements with respect to reality must be studied in order to avoid wind-up effects. Moreover, actuator state measurements may contain noise, biases, and delays. Of course, physical limitations exists and the attitude control system will depend on appropriate choice of sensors and actuators. In some particular cases, a combination of these two approaches may be necessary.

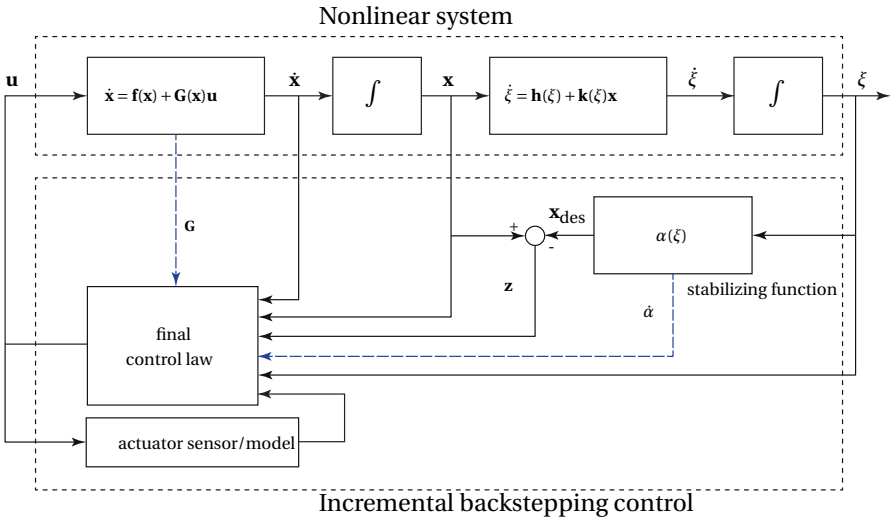


Figure 6.4: Incremental backstepping control block diagram for second order cascaded systems. Dashed arrows represent information required for control design. Notice that the final control law in this case requires knowledge of **G**, but also of $\dot{\mathbf{x}}$ and $\mathbf{u}_0$.

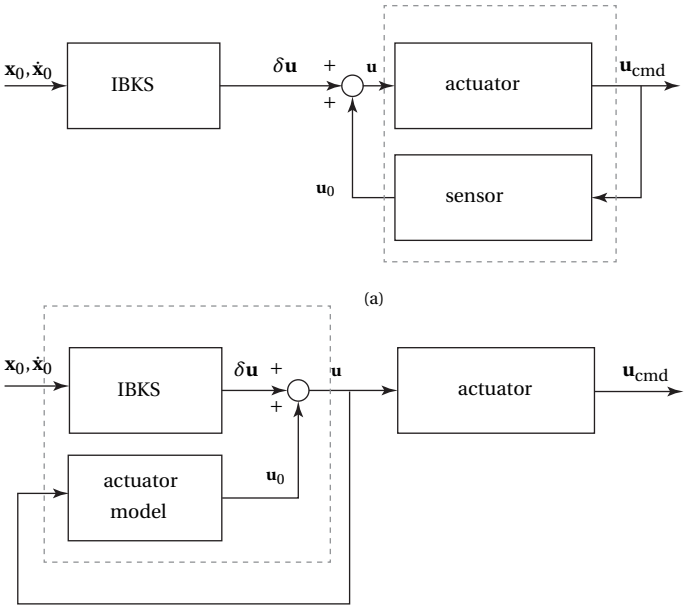


Figure 6.5: Actuator state measurement/estimation architectures for incremental backstepping: (a) sensor-dependent. (b) model-dependent.

6.3. FLIGHT CONTROL LAW DESIGN

The incremental backstepping methodology has remained quite general up to this point. In the following, for flight control law design, we will demonstrate this concept consider-

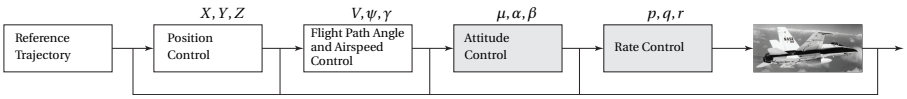


Figure 6.6: Four loop feedback design for flight control. Grey boxes represent the attitude and rate control systems considered for flight control law design in the following. Image credits: [6].

ing attitude and rate control, outer and inner loop, respectively, by applying the methodology as a single-loop control for both systems simultaneously. Extra outer loops, see Fig. 6.6, could be also considered in such control law design with backstepping, but not shown here. Notice that in general, structures for flight control have at their core several blocks of dynamic inversion [173]. Such architectures are difficult to study from the stability point of view due to the multi-loop interconnection and time-scale separation, in contrast with backstepping-based design which starts from the subsystem farthest from the control input and steps back through the integrators by considering augmented control Lyapunov functions (and hence from a stability view point) in a step-by-step fashion to obtain control laws for some desired motion with known stability and convergence properties.

In this sense, we demonstrate the incremental backstepping by considering Euler's equation of motion for the angular velocities of a vehicle in vector form:

$$\boldsymbol{M}_B = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \tag{6.11}$$

where $\boldsymbol{\omega} \in \mathscr{R}^3$ is the angular velocity vector, $\boldsymbol{M}_B \in \mathscr{R}^3$ is the external (unknown) moment vector in body axes, and $\boldsymbol{I}$ the inertia matrix of the rigid body (with $x-z$ a plane of symmetry). We will be interested in the time history of the angular velocity vector, hence the dynamics of the rotational motion of a vehicle in Eq. (6.11) can be rewritten as the following set of differential equations:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_B - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \tag{6.12}$$

where:

$$\boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad \boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix} \qquad \boldsymbol{M}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = SQ \begin{bmatrix} bC_l \\ \overline{c}C_m \\ bC_n \end{bmatrix}$$

with p, q, r, the body roll, pitch, and yaw rates, respectively; L, M, N, the roll, pitch, and yaw moments, respectively; and S the wing surface area, Q the dynamic pressure, b the wing span, $\overline{c}$ the mean aerodynamic chord, and C_l, C_m, C_n the moment coefficients for roll, pitch, and yaw, respectively. Furthermore, let $\boldsymbol{M}_B$ be the sum of moments partially generated by the aerodynamics of the airframe (subscript a), moments generated by the control derivatives (subscript c) times the deflection of control surfaces (δ), and external disturbance moments (subscript d):

$$\boldsymbol{M}_B = \boldsymbol{M}_a + \boldsymbol{M}_c\boldsymbol{\delta} + \boldsymbol{M}_d \tag{6.13}$$

where:

$$\boldsymbol{M}_a = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_a \qquad \boldsymbol{M}_c = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_c \qquad \boldsymbol{\delta} = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \qquad \boldsymbol{M}_d = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_d$$

and $\boldsymbol{\delta}$ corresponds to the control inputs: aileron, elevator, and rudder deflection angles, respectively. Hence, the dynamic equation in consideration can be rewritten as:

$$\dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}, \varsigma) + \mathbf{g}(\varsigma)\boldsymbol{\delta} + \mathbf{d} \tag{6.14}$$

with:

$$\mathbf{f}(\boldsymbol{\omega}, \varsigma) = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_a - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \qquad \mathbf{g}(\varsigma) = \boldsymbol{I}^{-1}\boldsymbol{M}_c \qquad \mathbf{d} = \boldsymbol{I}^{-1}\boldsymbol{M}_d$$

and $\varsigma \in \mathscr{R}^p$ a parameter vector. For the rotational motion, this equation becomes:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_a - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) + \boldsymbol{I}^{-1}\boldsymbol{M}_c\boldsymbol{\delta} + \boldsymbol{I}^{-1}\boldsymbol{M}_d \tag{6.15}$$

Without knowledge of the disturbances, and introducing the virtual control input $\boldsymbol{\nu} = \dot{\boldsymbol{\omega}}_{\text{des}}$, applying nonlinear dynamic inversion (NDI) to Eq. (6.15) results in an expression for the control input of the vehicle as:

$$\boldsymbol{\delta} = \boldsymbol{M}_c^{-1}\left(\boldsymbol{I}\boldsymbol{\nu} - \boldsymbol{M}_a + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \tag{6.16}$$

This resulting NDI control law depends on accurate (full) knowledge of the aerodynamic model contained in both $\boldsymbol{M}_a$ and $\boldsymbol{M}_c$, and hence depends on the model uncertainties contained therein. Furthermore it also depends on parametric uncertainties regarding inertia parameters, center of gravity, misalignment, etc. Such a dynamic inversion control law is intended to linearize and decouple the (inner loop) rotational dynamics in order to obtain an explicit desired closed loop dynamics to be followed. Notice that this result does not consider the effect of the external disturbance $\mathbf{d}$, and hence does not reject it properly. In the following, we are interested to go further using the result from backstepping for a more flexible and augmented design.

For the sake of simplicity, we will depart the study from *Step 2* of the backstepping design procedure explained before, assuming that outer-subsystem's stabilizing control laws are already obtained and stepped back up to the dynamic equation in consideration. In this sense, we depart from the final error-dynamics equation:

$$\dot{\mathbf{z}} = \dot{\boldsymbol{\omega}} - \dot{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\omega}) = \mathbf{f}(\boldsymbol{\omega}, \varsigma) + \mathbf{g}(\varsigma)\boldsymbol{\delta} - \dot{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\omega}) \tag{6.17}$$

where $\boldsymbol{\sigma}$ may represent a kinematic variable or a state stepped back from the outer-subsystems. For flight control law design, the goal is to stabilize the complete system described by the following augmented equation:

$$\dot{\mathbf{z}} = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_a - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) + \boldsymbol{I}^{-1}\boldsymbol{M}_c\boldsymbol{\delta} + \boldsymbol{I}^{-1}\boldsymbol{M}_d - \dot{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\omega}) \tag{6.18}$$

and with partial knowledge of the disturbance (full knowledge is practically impossible), and applying backstepping to Eq. (6.18) in combination with a nonlinear damping term

Γ_d [65, 76, 183] to handle the disturbance effect and control input uncertainty, a plausible expression for the control input of the vehicle results in:

$$\boldsymbol{\delta} = \boldsymbol{M}_c^{-1}\boldsymbol{I}\Big[-\mathbf{K}_{\boldsymbol{\omega}}\mathbf{z} - \boldsymbol{I}^{-1}\big(\boldsymbol{M}_a - \boldsymbol{\omega}\times\boldsymbol{I}\boldsymbol{\omega}\big) + \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega}) + \Gamma_d\Big] \tag{6.19}$$

with $\mathbf{K}_{\boldsymbol{\omega}} > 0_{3\times3}$. This control ensures $\mathbf{z}$ to be uniformly ultimately bounded, meaning that the complete system is stabilized, and the flexibility of the method allows to consider several families of control laws apart from a pure linearizing one. Moreover, the flexibility due to CLF augmentation and redesign allowes the inclusion of a nonlinear damping term Γ_d to reject external disturbance effect and possible input uncertainty. Again, the resulting control law depends on accurate (full) knowledge of the aerodynamic model contained in both $\boldsymbol{M}_a$ and $\boldsymbol{M}_c$, and hence also depends on the model uncertainties contained therein. For this reason, we complete the study by improving the robustness of such backstepping design by introducing its incremental counterpart, using the implicit approach with the recursive control law:

$$\boldsymbol{\delta} = \boldsymbol{\delta}_0 + \boldsymbol{M}_c^{-1}\boldsymbol{I}\Big[-\mathbf{K}_{\boldsymbol{\omega}}\mathbf{z} - \dot{\boldsymbol{\omega}}_0 + \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega}) + \Gamma_d\Big] \tag{6.20}$$

Which results in a stabilizing control law for outer-loop variables that is not depending on the aerodynamic model $\boldsymbol{M}_a$, hence it will not be affected by its uncertainties. In this case, the aerodynamic (control input) uncertainty present in $\boldsymbol{M}_c$, the parametric uncertainty, and the effect of external disturbance, are captured by the vehicle's accelerations and by the implicit architecture of the closed-loop system. Moreover, the extra nonlinear damping term may be suitable to alleviate this problem even further, but its contribution to the closed-loop robustness is not studied here.

6.4. ROBUSTNESS

Apart from the robustness properties already discussed before, the present section shows briefly closed-loop forms of the systems in consideration under feedback control for particular uncertainty structures. Ignoring the external disturbance for this analysis (and hence the nonlinear damping term), the application of the backstepping control law in Eq. (6.19) on the nominal system (6.18) results in the following stable closed-loop error-dynamics:

$$\dot{\mathbf{z}} = -\mathbf{K}_{\boldsymbol{\omega}}\mathbf{z} \tag{6.21}$$

Instead, if we consider the uncertain system with the fact that the error-dynamics (6.17) may contain uncertainties from the original dynamics as, for instance:

$$\dot{\mathbf{z}} = \mathbf{f}(\boldsymbol{\omega},\varsigma) + \Delta\mathbf{f}(\boldsymbol{\omega},\varsigma) + \big[\mathbf{g}(\varsigma) + \Delta\mathbf{g}(\varsigma)\big]\boldsymbol{\delta} - \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega}) \tag{6.22}$$

the application of the backstepping control law in Eq. (6.19) does not robustify the closed-loop dynamics against model and parametric uncertainty present in both $\Delta\mathbf{f}(\boldsymbol{\omega},\varsigma)$ and $\Delta\mathbf{g}(\varsigma)$, besides from the aerodynamic uncertainty contained therein,

$$\dot{\mathbf{z}} = -\left[\boldsymbol{I} + \frac{\Delta\mathbf{g}(\varsigma)}{\mathbf{g}(\varsigma)}\right]\mathbf{K}_{\boldsymbol{\omega}}\mathbf{z} + \Delta\mathbf{f}(\boldsymbol{\omega},\varsigma) - \frac{\Delta\mathbf{g}(\varsigma)}{\mathbf{g}(\varsigma)}\Big[\mathbf{f}(\boldsymbol{\omega},\varsigma) + \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega})\Big] \tag{6.23}$$

unless considering the robustification with a better nonlinear damping design or via robust backstepping, which will make the control law more conservative, see [65, 183].

As a matter of fact, we are interested on robustness properties from incremental backstepping. For the *partly*-linearized nonlinear system, recall we assume in this case angular accelerations to be known accurately, hence $\mathbf{f}(\boldsymbol{\omega},\varsigma)$ represents $\dot{\boldsymbol{\omega}}_0$ and not $\boldsymbol{I}^{-1}\big(\boldsymbol{M}_a - \boldsymbol{\omega}\times\boldsymbol{I}\boldsymbol{\omega}\big)$. Such difference is important since it not only represents a measurement versus an explicit model containing aerodynamic terms and parameters, but also because the term $\Delta\mathbf{f}(\boldsymbol{\omega},\varsigma)$ is no longer present in such case since such measurement uncertainty is considered negligible. For this reason, the uncertain system is rewritten as:

$$\dot{\mathbf{z}} = \dot{\boldsymbol{\omega}}_0 + \big[\mathbf{g}(\varsigma) + \Delta\mathbf{g}(\varsigma)\big]\Delta\boldsymbol{\delta} - \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega}) \tag{6.24}$$

and applying the incremental backstepping control law to such uncertain system results in:

$$\dot{\mathbf{z}} = -\left[\boldsymbol{I} + \frac{\Delta\mathbf{g}(\varsigma)}{\mathbf{g}(\varsigma)}\right]\mathbf{K}_{\boldsymbol{\omega}}\mathbf{z} - \frac{\Delta\mathbf{g}(\varsigma)}{\mathbf{g}(\varsigma)}\Big[\dot{\boldsymbol{\omega}}_0 + \dot{\alpha}(\boldsymbol{\sigma},\boldsymbol{\omega})\Big] \tag{6.25}$$

which only contains uncertainties in the control derivatives and moments of inertia.

6.5. EXAMPLE: LONGITUDINAL MISSILE CONTROL

In this section the advantage of incremental backstepping is demonstrated with an example consisting on the tracking control design for a longitudinal missile model. This example is adapted from [6]. A second order nonlinear model of a generic surface-to-air missile as obtained from [184] is considered. The model consists of the longitudinal force and moment equations representative of a missile traveling at an altitude of approximately 6000 meters, with aerodynamic coefficients represented as third order polynomials in angle of attack α and Mach number M.

The nonlinear equations of motion in the pitch plane are given by

$$\dot{\alpha} = q + \frac{\bar{q}S}{mV_T}\Big[C_z(\alpha,M) + b_z(M)\delta\Big] \tag{6.26a}$$

$$\dot{q} = \frac{\bar{q}Sd}{I_{yy}}\Big[C_m(\alpha,M) + b_m(M)\delta\Big] \tag{6.26b}$$

where:

$$C_z(\alpha,M) = \varphi_{z1}(\alpha) + \varphi_{z2}(\alpha)M \qquad b_z(M) = 1.6238M - 6.7240$$
$$C_m(\alpha,M) = \varphi_{m1}(\alpha) + \varphi_{m2}(\alpha)M \qquad b_m(M) = 12.0393M - 48.2246$$

and:

$$\varphi_{z1}(\alpha) = -288.7\alpha^3 + 50.32\alpha|\alpha| - 23.89\alpha \qquad \varphi_{z2}(\alpha) = -13.53\alpha|\alpha| + 4.185\alpha$$
$$\varphi_{m1}(\alpha) = 303.1\alpha^3 - 246.3\alpha|\alpha| - 37.56\alpha \qquad \varphi_{m2}(\alpha) = 71.51\alpha|\alpha| + 10.01\alpha$$

These approximations are valid for the flight envelope $-10^\circ \leq \alpha \leq 10^\circ$ and $1.8 \leq M \leq 2.6$. To facilitate the control design, the nonlinear missile model is rewritten in the more

general state-space form as:

$$\dot{x}_1 = x_2 + f_1(x_1) + g_1 u \tag{6.27a}$$
$$\dot{x}_2 = f_2(x_1) + g_2 u \tag{6.27b}$$

where:

$$x_1 = \alpha \qquad x_2 = q$$
$$f_1(x_1) = C_1\left[\varphi_{z1}(x_1) + \varphi_{z2}(x_1)M\right] \qquad f_2(x_1) = C_2\left[\varphi_{m1}(x_1) + \varphi_{m2}(x_1)M\right]$$
$$g_1 = C_1 b_z \qquad g_2 = C_2 b_m$$
$$C_1 = \frac{\bar{q}S}{mV_T} \qquad C_2 = \frac{\bar{q}Sd}{I_{yy}}$$

The control objective considered here is to design an autopilot with the incremental backstepping method that tracks a command reference y_r (all derivatives known and bounded) with the angle of attack x_1. It is assumed that the aerodynamic force and moment functions are *not* exactly known and the Mach number M is treated as a parameter available for measurement. Furthermore, the contribution of the fin deflection on the right-hand side of the force equation (6.27a) is ignored during the control design, since the backstepping method can only handle nonlinear systems of lower-triangular form, i.e. the assumption is made that the fin surface is a pure moment generator. This is a valid assumption for most types of aircraft and aerodynamically controlled missiles, often made in flight control systems design [6].

We begin the control design procedure with standard backstepping for illustration purposes and further comparisons.

Step 1: First, introduce the tracking errors as:

$$z_1 = x_1 - y_r \tag{6.28a}$$
$$z_2 = x_2 - \alpha_1 \tag{6.28b}$$

where α_1 is the stabilizing function to be designed as a first design step (and not to be confused with α, the angle of attack). The z_1-dynamics satisfy:

$$\dot{z}_1 = x_2 + f_1 - \dot{y}_r = z_2 + \alpha_1 + f_1 - \dot{y}_r \tag{6.29}$$

Consider a candidate CLF V_1 for the z_1-subsystem defined as:

$$V_1(z_1) = \frac{1}{2}\left(z_1^2 + k_1\lambda_1^2\right) \tag{6.30}$$

where the gain $k_1 > 0$ and the integrator term $\lambda_1 = \int_0^t z_1 \mathrm{d}t$ are introduced to robustify the control design against the effect of the neglected control term. The derivative of V_1 along the solutions of (6.29) is given by:

$$\dot{V}_1 = z_1\dot{z}_1 + k_1\lambda z_1 = z_1\left(z_2 + \alpha_1 + f_1 - \dot{y}_r + k_1\lambda_1\right) \tag{6.31}$$

The stabilizing function α_1 is selected as:

$$\alpha_1 = -c_1 z_1 - k_1 \lambda_1 - f_1 + \dot{y}_r, \quad c_1 > 0 \tag{6.32}$$

to render the derivative

$$\dot{V}_1 = -c_1 z_1^2 + z_1 z_2 \tag{6.33}$$

The cross term $z_1 z_2$ will be dealt with in the second design step.

Step 2: Second, the z_2-dynamics are given by:

$$\dot{z}_2 = f_2 + g_2 u - \dot{\alpha}_1 \tag{6.34}$$

where $\dot{\alpha}_1 = -c_1(x_2 + f_1 - \dot{y}_r) - k_1 z_1 - \dot{f}_1 + \ddot{y}_r$. The CLF V_1 is augmented with an additional term to penalize z_2:

$$V_2(z_1, z_2) = V_1 + \frac{1}{2} z_2^2 \tag{6.35}$$

The derivative of V_2 along the solutions of (6.29) and (6.34) satisfies

$$\dot{V}_2 = -c_1 z_1^2 + z_1 z_2 + z_2\big(f_2 + g_2 u - \dot{\alpha}_1\big) = -c_1 z_1^2 + z_2\big(z_1 + f_2 + g_2 u - \dot{\alpha}_1\big) \tag{6.36}$$

Notice that the first term in the right-hand of the last expression is already negative semi-definite. Hence, a control law for u can now be defined to cancel all indefinite terms, and the most straightforward choice is given by:

$$u = \frac{1}{g_2}\big(-c_2 z_2 - z_1 - f_2 + \dot{\alpha}_1\big) \tag{6.37}$$

According to the results previously outlined, the incremental backstepping control law design follows from considering the approximate dynamics around the current reference state for the dynamic equation of the pitch rate:

$$\dot{q} \cong \dot{q}_0 + \frac{\bar{q} S d}{I_{yy}} b_m(M) \Delta\delta \tag{6.38}$$

assuming that pitch acceleration is available for measurement, and which is rewritten in our formulation as:

$$\dot{x}_2 \cong \dot{x}_{2_0} + g_2 \Delta u \tag{6.39}$$

From there, the design procedure is the same as before. It suffices to consider the new $f_2 = \dot{x}_{2_0}$, noticing that we are replacing the accurate knowledge of f_2 by a measurement (or an estimate) instead, and this trade-off results in a robustified backstepping control law which is not entirely dependent on a model.

The incremental backstepping control law is hence obtained as:

$$u = u_0 + \frac{1}{g_2}\big(-c_2 z_2 - z_1 - \dot{x}_{2_0} + \dot{\alpha}_1\big) \tag{6.40}$$

Simulation results for the backstepping controller in Eq. (6.37) and the incremental backstepping controller in Eq. (6.40) are now presented. The maneuver simply consists

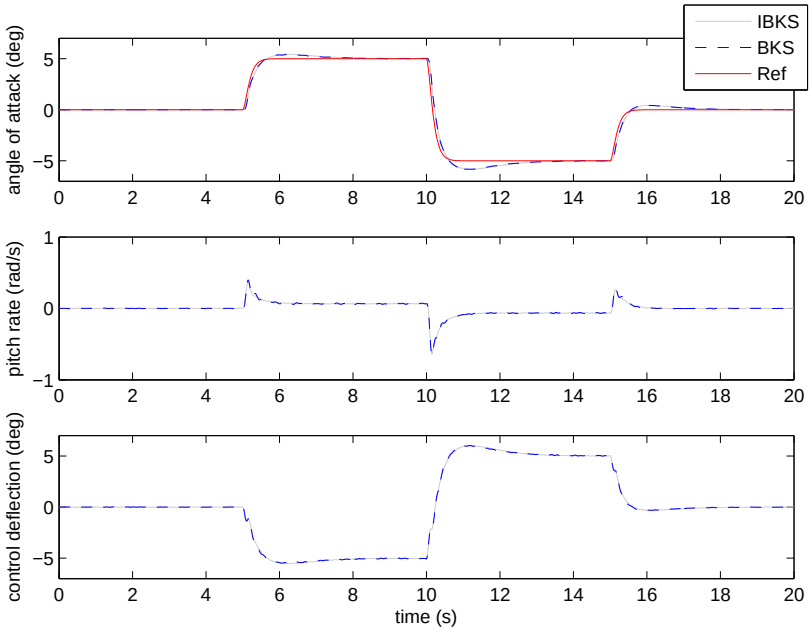


Figure 6.7: Backstepping (6.37) and incremental backstepping (6.40) tracking control numerical simulation of the nominal longitudinal missile model for a gain selection of $k_1 = c_1 = c_2 = 10$.

on a smooth doublet angle-of-attack trajectory for the missile. Figure 6.7 shows the tracking control numerical simulation at Mach 2.0 of the nominal (idealized) longitudinal missile model for the two control laws derived at the same gain selections of $k_1 = c_1 = c_2 = 10$, showing relatively the same performance and closed-loop response as expected with no uncertainty and model mismatch.

Now we introduce aerodynamic uncertainties modeled as real parametric uncertainty of the coefficients present in C_z, b_z, C_m, b_m. The coefficients are perturbed from their nominal value within a $\pm 20\%$ range. Figure 6.8 shows tracking control numerical simulation of the uncertain longitudinal missile model for the backstepping controller in Eq. (6.37) and with the same gain selection. As expected, this conventional backstepping alone is robust but not quite much over large dynamic uncertainties, and hence the nominal performance is lost and/or degraded.

For this particular example, the tracking capability and superior robustness at Mach 2.0 of the uncertain longitudinal missile model are verified as shown in Fig. 6.9, showing a great benefit of the incremental version over conventional backstepping designs since the new structure is able to cope very well with relatively large aerodynamic uncertainty, and hence the nominal performance is not lost and/or degraded significantly.

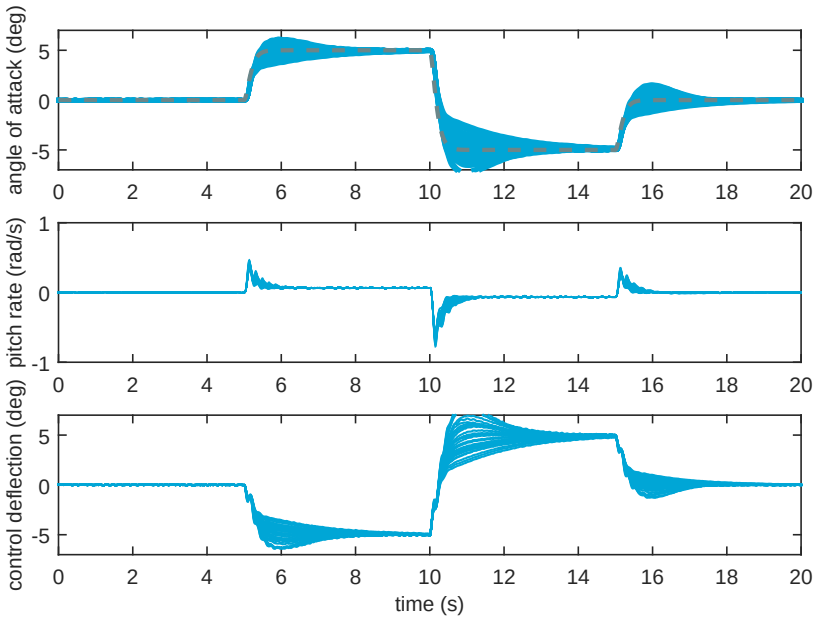


Figure 6.8: Backstepping (6.37) tracking control numerical simulation of the uncertain longitudinal missile model for a gain selection of $k_1 = c_1 = c_2 = 10$. Aerodynamic uncertainties are modeled as real parametric uncertainty of the coefficients present in C_z, b_z, C_m, b_m. The coefficients are perturbed from their nominal value within a ±20% range.

6.6. CONCLUSIONS

This chapter presented an application of the Incremental Nonlinear Dynamic Inversion (INDI) control methodology to the attitude tracking and disturbance rejection problem of rigid spacecraft in presence of model and parametric uncertainties. As a modification of the NDI methodology, the INDI approach enhances its robustness capabilities by reducing feedback control dependency on accurate knowledge of the system dynamics. The use of incremental control action, which requires information of actuator output and angular accelerations, make these sensor-based type of controllers efficient for external disturbance rejection and robust in terms of handling uncertainties. Unlike NDI, this control design technique is implicit in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed. Under the influence of external disturbances, time-delay, and parametric uncertainty, it was shown that incremental nonlinear dynamic inversion performs better than regular NDI and PI-control without compromising nominal performance and stability. However, in practice, INDI-based control rely on accurate actuator output and angular acceleration measurements which may not be readily available or which may contain noise, biases and delays, hence their effect need to be further studied.

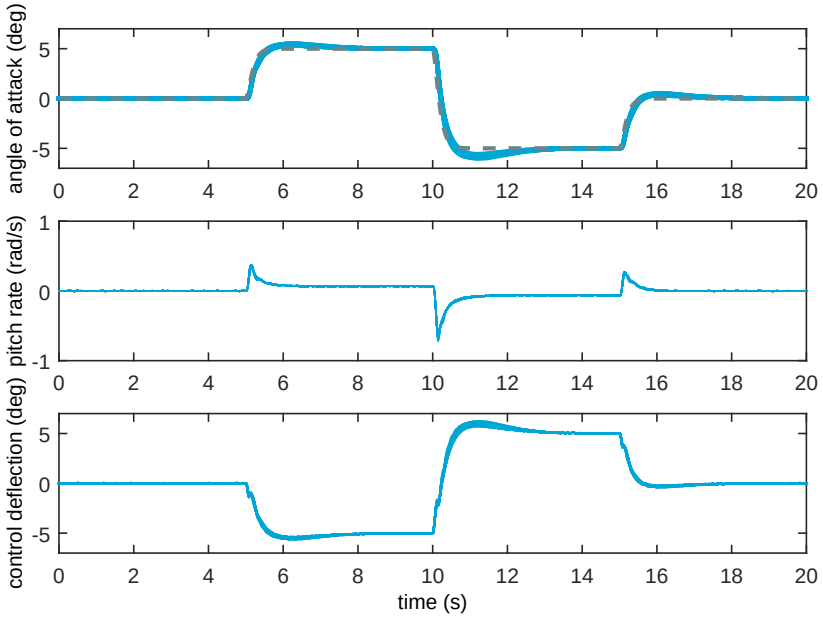


Figure 6.9: Incremental backstepping (6.40) tracking control numerical simulation of the uncertain longitudinal missile model for a gain selection of $k_1 = c_1 = c_2 = 10$. Aerodynamic uncertainties are modeled as real parametric uncertainty of the coefficients present in C_z, b_z, C_m, b_m. The coefficients are perturbed from their nominal value within a ±20% range.

7

PI(D) Tuning for Flight Control Systems via Incremental Nonlinear Dynamic Inversion

Abstract

Previous results reported in the robotics literature show the relationship between time-delay control (TDC) and proportional-integral-derivative control (PID). In this paper, we show that incremental nonlinear dynamic inversion (INDI) — more familiar in the aerospace community — are in fact equivalent to TDC. This leads to a meaningful and systematic method for PI(D)-control tuning of robust nonlinear flight control systems via INDI. We considered a reformulation of the plant dynamics inversion which removes effector blending models from the resulting control law, resulting in robust model-free control laws like PI(D)-control.

Publication

Paul Acquatella B., Wim van Ekeren, Qi Ping Chu: *PI(D) tuning for Flight Control Systems via Incremental Nonlinear Dynamic Inversion.* In: IFAC-PapersOnLine, Vol. **50**, No. 1, pp. 8175–8180, July 2017; presented at *IFAC-WC 2017, 20th World Congress of the International Federation of Automatic Control, July 9-14, 2017. Toulouse, France.*

7.1. INTRODUCTION

ENSURING stability and performance in between operational points of widely-used gain-scheduled linear PID controllers motivates the use of nonlinear dynamic inversion (NDI) for flight control systems. NDI cancels out nonlinearities in the model via state feedback, and then linear control can be subsequently designed to close the systems' outer-loop, hence eliminating the need of linearizing and designing different controllers for several operational points as in gain-scheduling.

In this chapter we consider nonlinear flight control strategies based on incremental nonlinear dynamic inversion (INDI). Using sensor and actuator measurements for feedback allows the design of an incremental control action which, in combination with nonlinear dynamic inversion, stabilizes the *partly*-linearized nonlinear system *incrementally*. With this result, dependency on exact knowledge of the system dynamics is greatly reduced, overcoming this major robustness issue from conventional nonlinear dynamic inversion. INDI has been considered a sensor-based approach because sensor measurements were meant to replace a large part of the vehicle model.

Theoretical development of increments of nonlinear control action date back from the late nineties and started with activities concerning 'implicit dynamic inversion' for inversion-based flight control [70, 73], where the architectures considered in this chapter were firstly described. Other designations for these developments found in the literature are 'modified NDI' and 'simplified NDI', but the designation 'incremental NDI', introduced in [77], is considered to describe the methodology and nature of these type of control laws better [71, 77, 79]. INDI has been elaborated and applied theoretically in the past decade for advanced flight control and space applications [4, 70, 71, 73–75, 80]. More recently, this technique has been applied also in practice for quadrotors and adaptive control [84, 85].

In this chapter, we present three main contributions in the context of nonlinear flight control system design.

1) We revisit the NDI/INDI control laws and we establish the equivalence between INDI and *time-delay control* (TDC).

2) Based on previous results reported in the robotics literature showing the relationship between discrete formulations of TDC and *proportional-integral-derivative control* (PID), we show that an equivalent PI(D) controller with gains $< K, T_i, (T_d) >$ tuned via INDI/TDC is more meaningful and systematic than heuristic methods, since one considers *desired* error dynamics given by Hurwitz gains $< k_P, (k_D) >$. Subsequently, tuning the remaining effector blending gain is much less cumbersome than designing a whole set of gains iteratively.

3) We also consider a reformulation of the plant dynamics inversion as it is done in TDC which removes the effector blending model (control derivatives) from the resulting control law. This has not been the case so far in the reported INDI controllers, causing robustness problems because of their uncertainties. Moreover, this allows to consider the introduced term as a scheduling variable which is only directly related to the proportional gain K.

7.2. FLIGHT VEHICLE MODELING

We are interested in Euler's equation of motion representing flight vehicles' angular velocity dynamics

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{M}_B \tag{7.1}$$

where $\boldsymbol{M}_B \in \mathscr{R}^3$ is the external moment vector in body axes, $\boldsymbol{\omega} \in \mathscr{R}^3$ is the angular velocity vector, and $\boldsymbol{I} \in \mathscr{R}^{3\times 3}$ the inertia matrix of the rigid body assuming symmetry about the plane $x-z$ of the body.

Furthermore, we will be interested in the time history of the angular velocity vector, hence the dynamics of the rotational motion of a vehicle (7.1) can be rewritten as the following set of differential equations

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_B - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \tag{7.2}$$

where

$$\boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \boldsymbol{M}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = SQ \begin{bmatrix} bC_l \\ \overline{c}C_m \\ bC_n \end{bmatrix},$$

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix},$$

with p,q,r, the body roll, pitch, and yaw rates, respectively; L,M,N, the roll, pitch, and yaw moments, respectively; S the wing surface area, Q the dynamic pressure, b the wing span, $\overline{c}$ the mean aerodynamic chord, and C_l, C_m, C_n the moment coefficients for roll, pitch, and yaw, respectively. Furthermore, let $\boldsymbol{M}_B$ be the sum of moments partially generated by the aerodynamics of the airframe $\boldsymbol{M}_a$ and moments generated by control surface deflections $\boldsymbol{M}_c$, and we describe $\boldsymbol{M}_B$ linearly in the deflection angles δ assuming the control derivatives to be linear as in [71] with $(\boldsymbol{M}_c)_\delta = \frac{\partial}{\partial \delta}\boldsymbol{M}_c$; therefore

$$\boldsymbol{M}_B = \boldsymbol{M}_a + \boldsymbol{M}_c = \boldsymbol{M}_a + (\boldsymbol{M}_c)_\delta \boldsymbol{\delta} \tag{7.3}$$

where

$$\boldsymbol{M}_a = \begin{bmatrix} L_a \\ M_a \\ N_a \end{bmatrix}, \boldsymbol{M}_c = \begin{bmatrix} L_c \\ M_c \\ N_c \end{bmatrix}, \boldsymbol{\delta} = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}$$

and $\boldsymbol{\delta}$ corresponding to the control inputs: aileron, elevator, and rudder deflection angles, respectively. Hence the dynamics (7.2) can be rewritten as

$$\dot{\boldsymbol{\omega}} = \boldsymbol{f}(\boldsymbol{\omega}) + \boldsymbol{G}(\boldsymbol{\omega})\boldsymbol{\delta} \tag{7.4}$$

with

$$\boldsymbol{f}(\boldsymbol{\omega}) = \boldsymbol{I}^{-1}\left(\boldsymbol{M}_a - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right), \ \boldsymbol{G}(\boldsymbol{\omega}) = \boldsymbol{I}^{-1}(\boldsymbol{M}_c)_\delta.$$

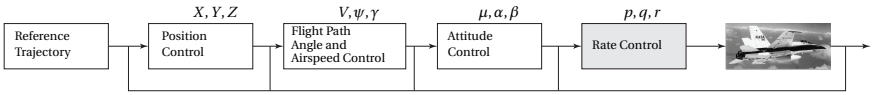


Figure 7.1: Four loop nonlinear flight control design. We are focused on nonlinear dynamic inversion of the rate control loop (grey box) in the following. Image credits: [6].

For practical implementations, we consider first-order actuator dynamics represented by the following transfer function

$$\frac{\delta_i}{\delta_{c_i}} = G_{a_i}(s) = \frac{K_{a_i}}{\tau_{a_i} s + 1}, \qquad (i = a, e, r), \tag{7.5}$$

and furthermore, we do not consider these actuator dynamics in the control design process as it is usually the case for dynamic inversion-based control. For that reason, we assume that these actuators are *sufficiently fast* in the control-bandwidth sense, meaning that $1/\tau_{a_i}$ is higher than the control system closed-loop bandwidth.

7.3. FLIGHT CONTROL LAW DESIGN

7.3.1. NONLINEAR DYNAMIC INVERSION

Let us define the control parameter to be the angular velocities, hence the output is simply $\boldsymbol{y} = \boldsymbol{\omega}$. We then consider an error vector defined as $\boldsymbol{e} = \boldsymbol{y}_d - \boldsymbol{y}$ where $\boldsymbol{y}_d$ denotes the *smooth* desired output vector (at least one time differentiable). Nonlinear dynamic inversion (NDI) is designed to linearize and decouple the rotational dynamics in order to obtain an *explicit* desired closed loop dynamics to be followed. Introducing the virtual control input $\boldsymbol{\nu} = \dot{\boldsymbol{\omega}}_{\text{des}}$, if the matrix $\boldsymbol{G}(\boldsymbol{\omega})$ is non-singular (i.e., invertible) in the domain of interest for all $\boldsymbol{\omega}$, the nonlinear dynamic inversion control consists in the following input transformation [61, 79]

$$\boldsymbol{\delta} = \boldsymbol{G}(\omega)^{-1}\left[\boldsymbol{\nu} - \boldsymbol{f}(\boldsymbol{\omega})\right] \tag{7.6}$$

which cancels all the nonlinearities, and a simple input-output linear relationship between the output y and the new input ν is obtained as

$$\dot{\boldsymbol{y}} = \boldsymbol{\nu} \tag{7.7}$$

Apart from being linear, an interesting result from this relationship is that it is also decoupled since the input ν_i only affects the output y_i. From this fact, the input transformation (7.6) is called a *decoupling control law*, and the resulting linear system (7.7) is called the *single-integrator* form. This single-integrator form (7.7) can be rendered exponentially stable with

$$\boldsymbol{\nu} = \dot{\boldsymbol{y}}_d + \boldsymbol{K}_P \boldsymbol{e} \tag{7.8}$$

where $\dot{\boldsymbol{y}}_d$ is the feedforward term for tracking tasks, and $\boldsymbol{K}_P \in \mathscr{R}^{3\times 3}$ a constant diagonal matrix, whose $i-$th diagonal elements k_{P_i} are chosen so that the polynomials

$$s + k_{P_i}, \qquad (i = p, q, r) \tag{7.9}$$

may become Hurwitz. This results in the exponentially stable and decoupled *desired* error dynamics

$$\dot{\boldsymbol{e}} + \boldsymbol{K}_P \boldsymbol{e} = \boldsymbol{0} \tag{7.10}$$

which implies that $\boldsymbol{e}(t) \to \boldsymbol{0}$. From this typical tracking problem it can be seen that the entire control system will have two control loops [71, 79]: the inner linearization loop (7.6), and the outer control loop (7.8). This resulting NDI control law depends on accurate knowledge of the aerodynamic moments, hence it is susceptible to model uncertainties contained in both $\boldsymbol{M}_a$ and $\boldsymbol{M}_c$.

In NDI control design, we consider outputs with relative degrees of one (rates), meaning a first-order system to be controlled, see Fig. 7.1. Extensions of input-output linearization for systems involving higher relative degrees are done via *feedback linearization* [61, 79].

7.3.2. INCREMENTAL NONLINEAR DYNAMIC INVERSION

The concept of incremental nonlinear dynamic inversion (INDI) amounts to the application of NDI to a system expressed in an incremental form. This improves the robustness of the closed-loop system as compared with conventional NDI since dependency on the accurate knowledge of the plant dynamics is reduced. Unlike NDI, this control design technique is *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed [73, 74].

To obtain an incremental form of system dynamics, we consider a first-order Taylor series expansion of $\dot{\omega}$ [4, 70, 71, 73–75, 95], not in the geometric sense, but with respect to a *sufficiently small* time-delay λ as

$$\begin{aligned}\dot{\boldsymbol{\omega}} &= \dot{\boldsymbol{\omega}}_0 + \frac{\partial}{\partial \boldsymbol{\omega}}\left[\boldsymbol{f}(\boldsymbol{\omega}) + \boldsymbol{G}(\boldsymbol{\omega})\boldsymbol{\delta}\right]\Big|_{\substack{\omega=\omega_0 \\ \delta=\delta_0}} (\boldsymbol{\omega} - \boldsymbol{\omega}_0) \\ &\quad + \frac{\partial}{\partial \boldsymbol{\delta}}\left[\boldsymbol{G}(\boldsymbol{\omega})\boldsymbol{\delta}\right]\Big|_{\substack{\omega=\omega_0 \\ \delta=\delta_0}} (\boldsymbol{\delta} - \boldsymbol{\delta}_0) + \mathcal{O}(\Delta\boldsymbol{\omega}^2, \Delta\boldsymbol{\delta}^2) \\ &\cong \dot{\boldsymbol{\omega}}_0 + \boldsymbol{f}_0(\boldsymbol{\omega} - \boldsymbol{\omega}_0) + \boldsymbol{G}_0(\boldsymbol{\delta} - \boldsymbol{\delta}_0)\end{aligned}$$

with

$$\dot{\boldsymbol{\omega}}_0 \equiv \boldsymbol{f}(\boldsymbol{\omega}_0) + \boldsymbol{G}(\boldsymbol{\omega}_0)\boldsymbol{\delta}_0 = \dot{\boldsymbol{\omega}}(t - \lambda) \tag{7.11a}$$

where $\boldsymbol{\omega}_0 = \boldsymbol{\omega}(t-\lambda)$ and $\boldsymbol{\delta}_0 = \boldsymbol{\delta}(t-\lambda)$ are the time-delayed signals of the current state ω and control δ, respectively. This means an approximate linearization about the $\lambda-$delayed signals is performed *incrementally*.

For such sufficiently small time-delay λ so that $\boldsymbol{f}(\boldsymbol{\omega})$ does not vary significantly during λ, we assume the following approximation to hold

$$\epsilon_{INDI}(t) \equiv \boldsymbol{f}(\boldsymbol{\omega}(t-\lambda)) - \boldsymbol{f}(\boldsymbol{\omega}(t)) \cong 0 \tag{7.12}$$

which leads to

$$\Delta\dot{\boldsymbol{\omega}} \cong \boldsymbol{G}_0 \cdot \Delta\boldsymbol{\delta} \tag{7.13}$$

Here, $\Delta\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0 = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}(t-\lambda)$ represents the incremental acceleration, and $\Delta\boldsymbol{\delta} = \boldsymbol{\delta} - \boldsymbol{\delta}_0$ represents the so-called incremental control input. For the obtained approximation $\dot{\boldsymbol{\omega}} \cong \dot{\boldsymbol{\omega}}_0 + \boldsymbol{G}_0(\boldsymbol{\delta} - \boldsymbol{\delta}_0)$, NDI is applied to obtain a relation between the incremental control input and the output of the system

$$\boldsymbol{\delta} = \boldsymbol{\delta}_0 + \boldsymbol{G}_0^{-1}\left[\boldsymbol{\nu} - \dot{\boldsymbol{\omega}}_0\right] \tag{7.14}$$

Note that the deflection angle $\boldsymbol{\delta}_0$ that corresponds to $\dot{\boldsymbol{\omega}}_0$ is taken from the output of the actuators, and it has been assumed that a commanded control is achieved *sufficiently fast* according to the assumptions of the actuator dynamics in (7.5). The total control command along with the obtained linearizing control $\Delta\boldsymbol{\delta}$ can be rewritten as

$$\boldsymbol{\delta}(t) = \boldsymbol{\delta}(t-\lambda) + \boldsymbol{G}_0^{-1}\Big[\boldsymbol{\nu} - \dot{\boldsymbol{\omega}}(t-\lambda)\Big]. \tag{7.15}$$

The dependency of the closed-loop system on accurate knowledge of the airframe model in $f(\omega)$ is largely decreased, improving robustness against model uncertainties contained therein. Therefore, this implicit control law design is more dependent on accurate measurements or accurate estimates of $\dot{\omega}_0$, the angular acceleration, and δ_0, the deflection angles, respectively.

Remark 1: By using the measured $\dot{\boldsymbol{\omega}}(t-\lambda)$ and $\boldsymbol{\delta}(t-\lambda)$ incrementally we practically obtain a robust, model-free controller with the self-scheduling properties of NDI.

Notice, however, that typical INDI control laws are nevertheless also depending on effector blending models reflected in $\boldsymbol{G}_0$, which makes this implicit controller susceptible to uncertainties in these terms. Instead, consider the following transformation as in [142]

$$\dot{\boldsymbol{\omega}} = \boldsymbol{H} + \bar{\boldsymbol{g}} \cdot \boldsymbol{\delta} \tag{7.16}$$

with

$$\boldsymbol{H}(t) = \boldsymbol{f}(\boldsymbol{\omega}) + (\boldsymbol{G}(\boldsymbol{\omega}) - \bar{\boldsymbol{g}})\boldsymbol{\delta},$$

and with the following (but not limited) options for $\bar{\boldsymbol{g}}$ [142], where $n = 3$ in our case

$$\bar{\boldsymbol{g}}_1 = k_G \cdot \mathscr{I}_n = k_G \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}, \quad \bar{\boldsymbol{g}}_2 = \begin{bmatrix} k_{G_1} & 0 & \cdots & 0 \\ 0 & k_{G_2} & & \\ \vdots & & \ddots & \\ 0 & & & k_{G_n} \end{bmatrix}$$

Applying nonlinear dynamic inversion (NDI) to (7.16) results in an expression for the control input of the vehicle as

$$\boldsymbol{\delta}(t) = \bar{\boldsymbol{g}}^{-1}\big[\boldsymbol{\nu}(t) - \boldsymbol{H}(t)\big]. \tag{7.17}$$

Considering $\boldsymbol{H}_0 = \dot{\boldsymbol{\omega}}_0 - \bar{\boldsymbol{g}} \cdot \boldsymbol{\delta}_0$, the incremental counterpart of (7.17) results in a control law that is neither depending on the airframe model nor the effector blending moments

$$\boldsymbol{\delta}(t) = \boldsymbol{\delta}(t-\lambda) + \bar{\boldsymbol{g}}^{-1}\Big[\boldsymbol{\nu} - \dot{\boldsymbol{\omega}}(t-\lambda)\Big]. \tag{7.18}$$

Remark 2: The self-scheduling properties of INDI in (7.15) due to the term $\boldsymbol{G}_0$ are now lost, suggesting that $\bar{\boldsymbol{g}}$ should be an scheduling variable.

7.3.3. TIME–DELAY CONTROL AND PROPORTIONAL INTEGRAL CONTROL

Time–delay control (TDC) [142] departs from the usual dynamic inversion input transformation of (7.16)

$$\boldsymbol{\delta}(t) = \bar{\boldsymbol{g}}^{-1}\left[\boldsymbol{\nu}(t) - \bar{\boldsymbol{H}}(t)\right] \tag{7.19}$$

where $\bar{\boldsymbol{H}}$ denotes an estimation of $\boldsymbol{H}$, being the nominal case when $\bar{\boldsymbol{H}} = \boldsymbol{H}$ which results in perfect inversion. Instead of having an estimate, the TDC takes the following assumption [142] analogous to (7.12)

$$\epsilon_{TDC}(t) \equiv \boldsymbol{H}(t-\lambda) - \boldsymbol{H}(t) \cong 0. \tag{7.20}$$

This relationship is used together with (7.16) to obtain what is called *time-delay estimation* (TDE) as the following

$$\bar{\boldsymbol{H}} = \boldsymbol{H}(t-\lambda) = \dot{\boldsymbol{\omega}}(t-\lambda) - \bar{\boldsymbol{g}} \cdot \boldsymbol{\delta}(t-\lambda) \tag{7.21}$$

In addition, $\epsilon(t)$ is called TDE *error* at time t. Combining the equations we obtain the following TDC law

$$\boldsymbol{\delta}(t) = \boldsymbol{\delta}(t-\lambda) + \bar{\boldsymbol{g}}^{-1}\left[\boldsymbol{\nu} - \dot{\boldsymbol{\omega}}(t-\lambda)\right] \tag{7.22}$$

which is in fact *equivalent* to the INDI control law obtained in (7.18). Appropriate selection of $\bar{\boldsymbol{g}}$ must ensure stability according to [142], and ideally, this term should be tuned according to the best estimate of the true effector blending moment $\hat{\boldsymbol{g}}(\tilde{\boldsymbol{\omega}})$ for measured angular velocities $\tilde{\boldsymbol{\omega}}$.

So far we have considered derivations in continuous-time. For practical implementations of these controllers and for the matters of upcoming discussions, sampled-time formulations involving continuous and discrete quantities as in [142] are more convenient and restated here. For that, considering that the smallest λ one can consider is the equivalent of the sampling period t_s of the on-board computer. The sampled formulation of (7.22) may be expressed as

$$\boldsymbol{\delta}(k) = \boldsymbol{\delta}(k-1) + \bar{\boldsymbol{g}}^{-1}\left[\boldsymbol{\nu}(k-1) - \dot{\boldsymbol{\omega}}(k-1)\right] \tag{7.23}$$

where it has been necessary to consider $\boldsymbol{\nu}$ at sample $k-1$ for causality reasons. Replacing the sampled virtual control $\boldsymbol{\nu}$ according to (7.8) we have

$$\boldsymbol{\delta}(k) = \boldsymbol{\delta}(k-1) + \bar{\boldsymbol{g}}^{-1}\left[\dot{\boldsymbol{e}}(k-1) + \boldsymbol{k}_P \boldsymbol{e}(k-1)\right] \tag{7.24}$$

and we can consider the following finite difference approximation of the error derivatives as angular accelerations are not directly measured

$$\dot{\boldsymbol{e}}(k) = [\boldsymbol{e}(k) - \boldsymbol{e}(k-1)]/t_s. \tag{7.25}$$

Consider now the standard *proportional-integral* (PI) control

$$\delta(t) = \boldsymbol{K}\left(\boldsymbol{e}(t) + \boldsymbol{T}_I^{-1}\int_0^t \boldsymbol{e}(\sigma)\mathrm{d}\sigma\right) + \boldsymbol{\delta}_{DC}, \tag{7.26}$$

where $\boldsymbol{K} \in \mathscr{R}^{3\times3}$ denotes a diagonal proportional gain matrix, $\boldsymbol{T}_I \in \mathscr{R}^{3\times3}$ a constant diagonal matrix representing a reset or integral time, and $\boldsymbol{\delta}_{DC} \in \mathscr{R}^3$ denotes a constant vector

representing a trim-bias, which acts as a trim setting and is computed by evaluating the initial conditions. The discrete form of the PI is given by

$$\boldsymbol{\delta}(k) = \boldsymbol{K}\Big(\boldsymbol{e}(k-1) + \boldsymbol{T}_I^{-1}\sum_{i=0}^{k-1} t_s \boldsymbol{e}(i)\Big) + \boldsymbol{\delta}_{DC} \tag{7.27}$$

When subtracting two consecutive terms of this discrete formulation, we can remove the integral sum and achieve the so-called PI controller in incremental form

$$\boldsymbol{\delta}(k) = \boldsymbol{\delta}(k-1) + \boldsymbol{K}\cdot t_s\big(\dot{\boldsymbol{e}}(k-1) + \boldsymbol{T}_I^{-1}\cdot \boldsymbol{e}(k-1)\big) \tag{7.28}$$

Following the same steps, and for completeness, we also present the PID extension by simply considering the extra derivative term $\ddot{\boldsymbol{e}}$

$$\boldsymbol{\delta}(k) = \boldsymbol{\delta}(k-1) + \boldsymbol{K}\cdot t_s\big(\boldsymbol{T}_D\ddot{\boldsymbol{e}}(k-1) + \dot{\boldsymbol{e}}(k-1) + \boldsymbol{T}_I^{-1}\cdot \boldsymbol{e}(k-1)\big),$$

where $\boldsymbol{T}_D \in \mathscr{R}^{3\times 3}$ denotes a constant diagonal matrix representing derivative time.

7.3.4. EQUIVALENCE OF INDI/TDC/PI(D)

Having in mind the found the equivalence between INDI and TDC, and comparing terms from (7.24) with (7.28), we have the following relationships as originally found in [142] which are the relationship between the discrete formulations of TDC and PI in incremental form

$$\boldsymbol{K} = (\bar{\boldsymbol{g}}\cdot t_s)^{-1}, \quad \boldsymbol{T}_I = \boldsymbol{k}_P^{-1} \tag{7.29}$$

Whenever the system under consideration is of second-order controller canonical form, we will have error dynamics of the form $\ddot{\boldsymbol{e}} + \boldsymbol{k}_D\dot{\boldsymbol{e}} + \boldsymbol{k}_P\boldsymbol{e} = \boldsymbol{0}$, and considering the newly introduced derivative gain $\boldsymbol{k}_D$ related to $\ddot{\boldsymbol{e}}$ we have

$$\boldsymbol{K} = \boldsymbol{k}_D\cdot(\bar{\boldsymbol{g}}\cdot t_s)^{-1}, \quad \boldsymbol{T}_I = \boldsymbol{k}_D\cdot\boldsymbol{k}_P^{-1}, \quad \boldsymbol{T}_D = \boldsymbol{k}_D^{-1} \tag{7.30}$$

This suggests not only that an equivalent discrete PI(D) controller with gains $< \boldsymbol{K}, \boldsymbol{T}_i, (\boldsymbol{T}_D) >$ can be obtained via INDI/TDC, but doing so is more meaningful and systematic than heuristic methods. This is because we begin the design from *desired* error dynamics given by Hurwitz gains $< k_P, (k_D) >$ and what follows is finding the remaining effector blending gain $\bar{g}$ either analytically whenever G is well known, with a proper estimate $\hat{G}$, or by tuning according to closed-loop requirements. As already mentioned, details on a sufficient condition for closed-loop stability under discrete TDC, and therefore applicable to its equivalent INDI, can be found in [142] and the references therein.

In essence, this procedure is more efficient and much less cumbersome than designing a whole set of gains iteratively. Moreover, for flight control systems, the self-scheduling properties of inversion-based controllers have suggested superior advantages with respect to PID controls since these must be gain-scheduled according to the flight envelope variations. The relationships here outlined suggests that PID-scheduling shall be done at the proportional gain K via the effector blending gain $\bar{g}$, and *not* over the whole set of gains $< K, T_i, (T_d) >$.

7.4. LONGITUDINAL FLIGHT CONTROL SIMULATION

In this section, robust PI tuning via INDI is demonstrated with a simple yet significant example consisting of the tracking control design for a longitudinal launcher vehicle model. The second-order nonlinear model is obtained from [6, 184], and it consists of longitudinal dynamic equations representative of a vehicle traveling at an altitude of approximately 6000 meters, with aerodynamic coefficients represented as third order polynomials in angle of attack α and Mach number M.

The nonlinear equations of motion in the pitch plane are given by

$$\dot{\alpha} = q + \frac{\bar{q}S}{mV_T}\Big[C_z(\alpha, M) + b_z(M)\delta\Big], \tag{7.31a}$$

$$\dot{q} = \frac{\bar{q}Sd}{I_{yy}}\Big[C_m(\alpha, M) + b_m(M)\delta\Big], \tag{7.31b}$$

where

$$\begin{aligned} C_z(\alpha, M) &= \varphi_{z1}(\alpha) + \varphi_{z2}(\alpha)M, \\ C_m(\alpha, M) &= \varphi_{m1}(\alpha) + \varphi_{m2}(\alpha)M, \\ b_z(M) &= 1.6238M - 6.7240, \\ b_m(M) &= 12.0393M - 48.2246, \end{aligned}$$

and

$$\begin{aligned} \varphi_{z1}(\alpha) &= -288.7\alpha^3 + 50.32\alpha|\alpha| - 23.89\alpha, \\ \varphi_{z2}(\alpha) &= -13.53\alpha|\alpha| + 4.185\alpha, \\ \varphi_{m1}(\alpha) &= 303.1\alpha^3 - 246.3\alpha|\alpha| - 37.56\alpha, \\ \varphi_{m2}(\alpha) &= 71.51\alpha|\alpha| + 10.01\alpha. \end{aligned}$$

These approximations are valid for the flight envelope of $-10° \le \alpha \le 10°$ and $1.8 \le M \le 2.6$. To facilitate the control design, the nonlinear longitudinal model is rewritten in the more general state-space form as

$$\dot{x}_1 = x_2 + f_1(x_1) + g_1 u \tag{7.32a}$$

$$\dot{x}_2 = f_2(x_1) + g_2 u \tag{7.32b}$$

where:

$$x_1 = \alpha, \qquad x_2 = q$$
$$g_1 = C_1 b_z, \qquad g_2 = C_2 b_m$$

and

$$f_1(x_1) = C_1\big[\varphi_{z1}(x_1) + \varphi_{z2}(x_1)M\big], \qquad C_1 = \frac{\bar{q}S}{mV_T},$$

$$f_2(x_1) = C_2\big[\varphi_{m1}(x_1) + \varphi_{m2}(x_1)M\big], \qquad C_2 = \frac{\bar{q}Sd}{I_{yy}}.$$

The control objective considered here is to design a PI autopilot via INDI that tracks a smooth command reference y_r with the pitch rate x_2. It is assumed that the aerodynamic force and moment functions are accurately known and the Mach number M is treated as a parameter available for measurement. Moreover, for this second-order system in non-lower triangular form due to $g_1 u$ and $f_2(x_1)$, pitch rate control using INDI is possible due to the time-scale separation principle [71, 79]. With respect to actuator dynamics modeled as in (7.5), we consider $K_a = 1$, and $\tau_a = 1e^{-2}$.

7.4.1. PITCH RATE CONTROL DESIGN

First, introduce the rate-tracking error

$$z_2 = x_2 - x_{2_{ref}} \tag{7.33}$$

the z_2-dynamics satisfy the following error

$$\dot{z}_2 = \dot{x}_2 - \dot{x}_{2_{ref}} \tag{7.34}$$

for which we design the following exponentially stable *desired* error dynamics

$$\dot{z}_2 + k_{P_2} z_2 = 0, \quad k_{P_2} = 50 \text{ rad/s}. \tag{7.35}$$

According to the results previously outlined, the incremental nonlinear dynamic inversion control law design follows from considering the approximate dynamics around the current reference state for the dynamic equation of the pitch rate as in (7.13)

$$\dot{q} \cong \dot{q}_0 + \bar{g} \cdot \Delta\delta \tag{7.36}$$

assuming that pitch acceleration is available for measurement and the scalar $\bar{g}$ to be a factor of the accurately known estimate of g_2

$$\bar{g} = k_G \cdot \hat{g}_2, \quad k_G = 1.$$

This is rewritten in our formulation as

$$\dot{x}_2 \cong \dot{x}_{2_0} + \bar{g} \cdot \Delta u \tag{7.37}$$

where recalling that $\dot{x}_{2_0}$ is an incremental instance before $\dot{x}_2$, and therefore the incremental nonlinear dynamic inversion law is hence obtained as

$$u = u_0 + \bar{g}^{-1}\left(\nu - \dot{x}_{2_0}\right), \tag{7.38}$$

with

$$\nu = -k_{P_2} z_2 + \dot{x}_{2_{ref}}, \tag{7.39}$$

or more compactly

$$u = u_0 + \bar{g}^{-1}\left(-k_{P_2} z_2 - \dot{x}_{2_0} + \dot{x}_{2_{ref}}\right) \tag{7.40}$$

This results as desired, in the following z_2-dynamics

$$\dot{z}_2 = \dot{x}_{2_0} + \bar{g} \cdot \Delta u - \dot{x}_{2_{ref}}. \tag{7.41}$$

Notice that we are replacing the accurate knowledge of f_2 by a measurement (or an estimate) as $f_2 \cong \dot{x}_{2_0}$, which will result in a control law which is not entirely dependent on a model, hence more robust.

We now consider these continuous-time formulations in sampled-time. To that end, we replace the small λ with the sampling period t_s so that $t_k = k \cdot t_s$ is the $k-$th sampling instant at time k, and therefore

$$\begin{aligned} u(k) = u(k-1) + \\ \bar{g}^{-1}\big[-k_{P_2} z_2(k-1) - \dot{x}_2(k-1) + \dot{x}_{2_{ref}}(k-1)\big], \end{aligned} \tag{7.42}$$

where due to causality relationships we need to consider the independent variables at the same sampling time $k-1$.

Referring back to the derived relationship between INDI and PI control, the equivalent PI control in incremental form is

$$u(k) = u(k-1) + K \cdot t_s\big[\dot{z}_2(k-1) + T_I^{-1} z_2(k-1)\big], \tag{7.43}$$

with

$$K = (\bar{g} \cdot t_s)^{-1}, \quad T_I = k_{P_2}^{-1} \tag{7.44}$$

The nature of the desired error dynamics (proportional) gain k_{P_2} is therefore of an integral control action, whereas the effector blending gain $\bar{g}$ act as proportional control. Having designed for desired error dynamics, and for a given sampling time t_s, tuning a pitch rate controller is only a matter of selecting a proper effector blending gain $\bar{g}$ according to performance requirements.

Remark 3: Notice at this point that having the PI control in incremental form introduces a finite difference of the error state, which is the equivalent counterpart of what has been considered the acceleration or state derivative $\dot{x}_{2_0}$ in INDI controllers.

Remark 4: Notice also that designing the PI control gains via INDI is highly beneficial, since only the effector blending gain is the tuning variable. This strongly suggests that robust adaptive control can be achieved by scheduling this variable online during flight and not over the whole set of gains.

Simulation results for the INDI/PI control are presented in Figure 7.2, considering smooth rate doublets for a nominal longitudinal dynamics model at Mach 2. For both controllers, the same zero-mean Gaussian white-noise with standard deviation $s_{d_q} = 1e^{-3}$ rad/s is added to the rates to simulate noisy measurements. The designed INDI gains of $k_{P_2} = 50$ rad/s and $k_G = 1$ are mapped to PI gains resulting in $K = 100\ \hat{g}_2^{-1}$ and $T_I = 0.02$ s, both controllers showing identical closed-loop response as expected.

With this example, it is demonstrated how a self-scheduled PI can be tuned via INDI by departing from desired error dynamics with the gain k_{P_2}, and considering an accurate effector blending model estimate $\bar{g} = \hat{g}_2$.

7.5. CONCLUSIONS

This chapter presented a meaningful and systematic method for PI(D) tuning of robust nonlinear flight control systems based on results previously reported in the robotics lit-

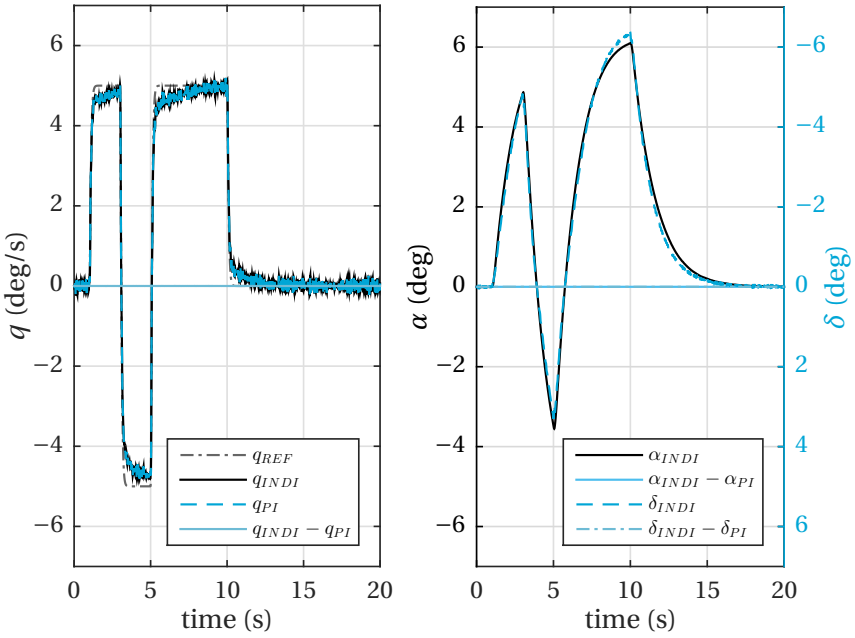


Figure 7.2: INDI/PI nominal tracking control simulation of the flight model (7.31) for $k_{P_2} = 50$ rad/s and $k_G = 1$

erature regarding the relationship between *time-delay control* (TDC) and *proportional-integral-derivative control* (PID). The method was demonstrated in the context of an example for the pitch rate tracking of a conventional longitudinal nonlinear flight model, showing the same tracking performance under nominal conditions.

Being incremental nonlinear dynamic inversion (INDI) equivalent to TDC clearly suggests that imposing *desired* error dynamics, as usual for INDI control laws, and then mapping these into an equivalent incremental PI(D)-controller together with control derivatives leads to a meaningful and systematic PI(D) gain tuning method, which is very difficult to do heuristically.

We considered a reformulation of the plant dynamics inversion which reduces knowledge of the effector blending model (control derivatives) from the resulting control law, reducing feedback control dependency on accurate knowledge of both the aircraft and effector blending models, hence resulting in robust and model-free control laws like the PI(D) control. Since usual flight control systems involves gain scheduling over the flight envelope, another key benefit of this result is that scheduling only the effector blending gain seems promising for adaptive control systems.

8

Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion

Abstract

This paper presents the attitude tracking and disturbance rejection problem of rigid spacecraft in presence of model and parametric uncertainty using nonlinear dynamic inversion. The feedback of actuator output and angular acceleration measurements allows the design of incremental control action which, in combination with nonlinear dynamic inversion, stabilizes the partly-linearized nonlinear system incrementally, accounting for undesired factors such as external perturbations, time-delays, and uncertainties. With this result, the so-called incremental nonlinear dynamic inversion, dependency on exact knowledge of the system dynamics is greatly reduced, overcoming this major robustness flaw from conventional nonlinear dynamic inversion. Moreover, this methodology implies a trade-off between accurate knowledge of the spacecraft dynamic model and accurate knowledge of the spacecraft sensors and actuators, and hence more suitable than identification or model-based adaptive control architectures. Simulation results demonstrate the tracking and external disturbances rejection capabilities of the proposed controller in front of current existing methods. Under the combined effect of external disturbances, time-delay, and parametric uncertainty, incremental nonlinear dynamic inversion performs better than regular NDI and PI-control without compromising nominal performance and stability.

Publication

Paul Acquatella B., Wouter Falken, Erik-Jan van Kampen, Qi Ping Chu: *Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion.* In: Proceedings of the 2012 AIAA Guidance, Navigation, and Control Conference, August 13-16, 2012. Minneapolis, Minnesota, United States.

8.1. INTRODUCTION

SEVERAL nonlinear control methodologies have arisen in the past decades to overcome shortcomings of conventional linearization approaches that approximates a nonlinear system dynamics only over a small domain of interest around equilibria, i.e., Jacobian linearization. One of the most popular of these control methodologies is known as Feedback Linearization (FBL), treated extensively in [61–63] and initially based on early papers of Krener and Brockett in the seventies [185, 186]. Feedback linearization uses an accurate model of the system to entirely or partly cancel its nonlinearities by means of feedback and exact state transformations. This transforms the nonlinear system dynamics into a linear one over a desired region of interest. For the obtained system, conventional linear control techniques can be applied successfully for achieving desired closed-loop dynamics.

The application of FBL in the aerospace field is commonly referred to as Nonlinear Dynamic Inversion (NDI). Although initially intended for flight control, NDI for aerospace applications have also found its way for spacecraft control and re-entry vehicles, see [50, 51, 67] and the references therein. The motivations behind the application of NDI for flight control originate from difficulties with ensuring stability and performance in between operational points of widely-used gain-scheduled controllers. With gain-scheduling, the flight envelope is divided into many smaller operating regimes and conventional controllers can be designed over each of them. Hence, scheduling the controller gains allows to obtain satisfactory performance and desired handling qualities over the entire flight envelope. In contrast, the NDI approach intends to eliminate the nonlinearities in the model by canceling them out with state feedback. In this case, a single classical linear controller can be used to close the outer loop of the system under NDI control, hence eliminating the need of linearizing and designing different controllers for several operational points as in gain-scheduling. This all is done under the assumption of a correct onboard dynamic model. NDI can be seen as a special case of FBL, where only one differentiation of each control variable is required to enable inversion (hence a relative degree of one). Moreover, NDI may be only applied in combination with physical insight [60].

A disadvantage for the construction of NDI-based control laws is that accurate knowledge of the nonlinear system dynamics is required for such an explicit cancellation. For space applications, this means that in order to apply NDI successfully, *both* the model of the system must match the onboard model of the spacecraft, and all system nonlinearities must be accurately known. Such assumption is hardly met in reality and in practice, which is the rationale behind further development on the robustness of this methodology.

For this reason, NDI is considered an *explicit* control method where the desired dynamics of the closed-loop system reside in some explicit model to be followed. Therefore, this explicit aspect of NDI-based control laws is considered to be a disadvantage upfront its abilities to linearize and decouple certain classes of nonlinear MIMO systems when full knowledge of the nonlinearities is available. Moreover, this model-based aspect is also strongly influenced by modeling uncertainties. In reality, the model mismatch in the implementation of NDI control laws, together with all sensor aspects, delays and biases, can compromise tremendously the performance of the controlled system. The high

dependency of the inner loop of the control system on the onboard model is critical, i.e., it can compromise the stability and performance of the system when performing under actuator failures or with model uncertainties.

Many successful attempts have been carried to identify and reduce these aforementioned flaws of NDI-based control laws with regards to robustness. These attempts are focused in improving the robustness of the overall control architecture by means of applying linear robust control in the outer loop of the system. The works [51, 69] combine NDI with the structured singular value (μ-analysis) and $\mathscr{H}_\infty$ synthesis for reentry flight clearance, and significant benefits were found over conventional NDI. However, not all uncertainties were taken into account or they were covered by lumped uncertainties hence introducing conservatism.

In this chapter, the issue of dealing with uncertainties with the NDI approach is presented by means a modification to the NDI framework that reduces its dependency on the onboard model or baseline spacecraft while making use of actuator output and angular acceleration measurement feedback, the so-called *Incremental Nonlinear Dynamic Inversion* (INDI) method. In contrast to regular NDI, this method is inherently *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed.

The theoretical development of INDI date back from the late nineties and started with the work from Smith [70] for NDI-based flight control. The INDI control methodology is also referred in the literature as Modified NDI and Simplified NDI (especially during its origins), but the designation 'Incremental NDI' is considered to describe the methodology and nature of these type of control laws better [71, 78, 79]. INDI has been elaborated and applied theoretically in the past decade for flight control applications [71–75].

The remainder of the chapter is organized as follows. Section 8.2 presents a brief recapitulation of attitude kinematics and dynamics, and the modeling of external disturbances considered. In Sections 8.3 and 8.4 we present the theory behind NDI and INDI, respectively. Section 8.5 illustrates the design of the spacecraft attitude controllers considered in this chapter, and Section 8.6 presents the simulations of these controllers in a rest-to-rest benchmark maneuver. Conclusions are discussed in Section 8.7.

8.2. MODELING

8.2.1. ATTITUDE KINEMATICS

Typically, the Euler angles and quaternions are used to parametrize the attitude kinematics of rigid bodies, and most attitude controllers are based on these parameterization. In general, the best parameterization is problem dependent [187].

In this chapter we will be interested in the Modified Rodrigues Parameters (MRPs), despite of their unpopularity for attitude control, as they represent a suitable kinematic parameterization for the particular application of spacecraft attitude control. They address the problem of singular orientations while using a minimal set of three rigid body attitude coordinates [188, 189]. Being derived from the quaternion through stereographic projection, they result in a well-defined parameterization for all Eigen-axis rotations in the range of $0 \leq \theta < 360^\circ$ and hence their potential advantages in attitude stabilization and control problems [188].

With $\boldsymbol{\omega} \in \mathscr{R}^3$ representing the angular velocity vector, and defining the vector $\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \end{bmatrix}^\top \in \mathscr{R}^3$, the kinematics equations take the form:

$$\dot{\boldsymbol{\sigma}} = \mathbf{N}(\boldsymbol{\sigma})\boldsymbol{\omega}, \qquad \boldsymbol{\sigma}(0) = \boldsymbol{\sigma}_0 \tag{8.1}$$

where:

$$\mathbf{N}(\boldsymbol{\sigma}) = \frac{1}{2}\left[\mathrm{I}_{3\times 3} - S(\boldsymbol{\sigma}) + \boldsymbol{\sigma}\boldsymbol{\sigma}^\top - \frac{1}{2}\left(1 + \boldsymbol{\sigma}^\top\boldsymbol{\sigma}\right)\mathrm{I}_{3\times 3}\right] \tag{8.2}$$

and $S(\boldsymbol{\sigma})$ denotes the 3×3 antisymmetric matrix:

$$S(\boldsymbol{\sigma}) = \begin{bmatrix} 0 & \sigma_3 & -\sigma_2 \\ -\sigma_3 & 0 & \sigma_1 \\ \sigma_2 & -\sigma_1 & 0 \end{bmatrix} \tag{8.3}$$

The vector $\boldsymbol{\sigma}$ of the Modified Rodrigues Parameters is related to the Euler-axis and the principal angle through:

$$\boldsymbol{\sigma} = \boldsymbol{\lambda} \tan\frac{\theta}{4} \tag{8.4}$$

8.2.2. ATTITUDE DYNAMICS

Along with the rotational kinematics which describes the orientation of a rigid body that is in rotational motion without involving any associated forces, the full description of the orientation of a rigid body is described with the dynamics of this rotational motion involving the influence of external forces. Consider Euler's rotational equation of motion in vector form [190]:

$$\mathbf{M} = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} \tag{8.5}$$

where $\boldsymbol{\omega} \in \mathscr{R}^3$ is the angular velocity vector, $\mathbf{M} \in \mathscr{R}^3$ is the external moment vector, and J the inertia matrix of the rigid body. We will be interested in the time history of the angular velocity vector, hence the dynamics of the rotational motion of a rigid body in Eq. (8.5) can be rewritten as the following set of differential equations:

$$\dot{\boldsymbol{\omega}} = J^{-1}S(\boldsymbol{\omega})J\boldsymbol{\omega} + J^{-1}\mathbf{M}, \quad \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0 \tag{8.6}$$

where:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \qquad J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}$$

and $S(\boldsymbol{\omega})$ denotes the 3×3 antisymmetric matrix:

$$S(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{8.7}$$

8.2.3. EXTERNAL DISTURBANCES

A spacecraft in orbit will experience external disturbance torques. The total moment acting on the rigid body can be decomposed in several terms depending on their nature or on the modeling of the problem considered. Here we will decompose the total moment, $\mathbf{M} \equiv \mathbf{u}$, in two terms: the control torque $\mathbf{u}_c$, and the disturbance torque $\mathbf{u}_d$:

$$\mathbf{M} \equiv \mathbf{u} = \mathbf{u}_c + \mathbf{u}_d \tag{8.8}$$

The idea is to model $\mathbf{u}_d$ as an external disturbance and test the disturbance rejection capabilities of the controllers proposed in this chapter. Such case is known in the literature as the *disturbance rejection* problem.

The magnitudes from these external disturbance torques differ greatly, but compared with the attitude control torques from the spacecraft they result to be very small. The most important disturbances for a satellite in orbit are the gravity field of the Earth due to its non radial symmetric mass distribution, atmospheric drag, third body perturbations, solar radiation pressure, and electromagnetic forces. In general, these space environmental external disturbance torques can be modeled as a bias plus cyclic terms in the body-fixed control axes [190], and in this chapter a sinusoidal disturbance torque is assumed as [191]:

$$\mathbf{u}_{d_i} = \begin{pmatrix} 4 + 0.2\sin 0.01\pi t \\ 5 + 0.5\sin 0.01\pi t \\ 4 + 0.2\sin 0.01\pi t \end{pmatrix} \times 10^{-3}\ \mathrm{N\cdot m} \tag{8.9}$$

In this sense, the problem in consideration can be formulated as Eq. (8.6) finally expressed as:

$$\dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}) + \mathbf{G}\mathbf{u}, \quad \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0 \tag{8.10}$$

with:

$$\mathbf{f}(\boldsymbol{\omega}) = J^{-1} S(\boldsymbol{\omega}) J\boldsymbol{\omega}, \qquad \mathbf{G} = J^{-1}$$

8.3. NONLINEAR DYNAMIC INVERSION

The idea of Nonlinear Dynamic Inversion (NDI) consists on canceling the nonlinearities in a nonlinear system so that the closed-loop dynamics is in a linear form. In other words, a nonlinear system is inverted by means of state feedback into a linear structure, and hence conventional linear controllers can be applied. A fundamental assumption is that the model of the system is exactly known, which gives NDI a great disadvantage from the point of view of uncertainties. Formally:

1. It is assumed that the model of the system is accurately known. This is known as the nominal model for which the NDI control laws are derived. Furthermore, this assumption implies that the original system will behave according to this nominal model.

2. It is assumed to have complete and accurate knowledge about the state of the system.

This will be referred as to the *full knowledge assumption*.

Without loss of generality, consider a general MIMO system whose number of inputs are equal to the number of outputs in order to avoid control allocation problems. The dynamics of this system can be expressed as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \tag{8.11a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{8.11b}$$

where $\mathbf{x} \in \mathscr{R}^n$ is the *state* vector, $\mathbf{u} \in \mathscr{R}^m$ is the *control input* vector (of components u_j), $\mathbf{y} \in \mathscr{R}^m$ is the system *output* vector (of components y_j), $\mathbf{f}$ and $\mathbf{h}$ are smooth vector fields on $\mathscr{R}^n$, and $\mathbf{G} \in \mathscr{R}^{n\times m}$ is a matrix whose columns are smooth vector fields $\mathbf{g}_j$. Note that this general dynamic model of a MIMO nonlinear system is considered *affine* in the control input $\mathbf{u}$. As a result there is no need of nonlinear solvers for $\mathbf{u}$ when applying NDI.

The elements of the output vector $\mathbf{y}$ are often selected as the parameters to be controlled, which are commonly directly related to the physical states of the system. For all the outputs y_i, $i = \{1,\dots,m\}$ the number of differentiation needed for the input to appear, i.e., $\{r_1,\dots,r_m\}$ is called the relative degree of the system [61]. In this sense, consider the output of the system to be the state:

$$\mathbf{y} = \mathbf{x} \tag{8.12}$$

Furthermore, if the matrix $\boldsymbol{G}(\mathbf{x})$ is non-singular (i.e., invertible) in the domain of interest for all $\mathbf{x}$, the NDI control consists in the following input transformation:

$$\mathbf{u} = \mathbf{G}^{-1}(\boldsymbol{x})\Big[\boldsymbol{\nu} - \boldsymbol{f}(\boldsymbol{x})\Big] \tag{8.13}$$

which cancels all the nonlinearities, and a simple input-output linear relationship is obtained between the output $\mathbf{y}$ and the new input $\boldsymbol{\nu}$:

$$\dot{\mathbf{y}} = \boldsymbol{\nu} \tag{8.14}$$

Apart from being linear, an interesting result from this relationship is that it is also decoupled since the input ν_i only affects the output y_i. From this fact, the input transformation (8.13) is called a *decoupling control law*, and the resulting linear system (8.14) is called the *single-integrator* form.

This single-integrator form (8.14) can be rendered exponentially stable by means of linear feedback control. In general, the introduced virtual input $\boldsymbol{\nu}$ can be designed to solve the problem of stabilization or output tracking, depending on control requirements. For both cases, given the obtained linear and decoupled relationship, linear controllers can be synthesized to guarantee time- or frequency-domain requirements. For example, the control law for the tracking of a smooth desired output $\mathbf{y}_{\text{ref}}(t)$ can be based upon the tracking error $e = \mathbf{y}_{\text{ref}}(t) - \mathbf{y}(t)$ as follows:

$$\boldsymbol{\nu} = \dot{\mathbf{y}}_{\text{ref}} + K(e, \dot{e}, \dots) \tag{8.15}$$

where $\dot{\mathbf{y}}_{\text{ref}}$ is the feedforward term for tracking tasks, and $K(e,\dot{e},\ldots)$ represents a linear controller to exponentially stabilize the tracking dynamics with proper gain tuning, so that $e(t) \to 0$. From this typical tracking problem, and as illustrated in Figure 8.1, it can be seen that the entire control system will have two control loops: the inner linearization loop based on Equation (8.13), and the outer control loop based on Equation (10.46).

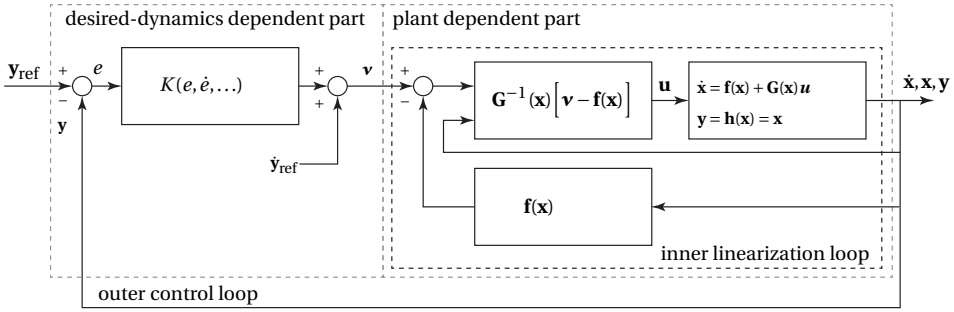


Figure 8.1: Tracking of a MIMO system with Nonlinear Dynamic Inversion. The inner linearization loop is based on Equation (8.13), whereas each channel $y_i - \nu_i$ of the outer control loop is based on linear control.

8.4. INCREMENTAL NONLINEAR DYNAMIC INVERSION

The concept of Incremental Nonlinear Dynamic Inversion (INDI) amounts to the application of NDI to a system expressed in an incremental form in order to improve the robustness of the closed-loop system as compared with conventional NDI-based control by reducing its dependency on the model and exact knowledge on the plant dynamics. Unlike NDI, this control design technique is implicit in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed [73]. The incremental form considers the influence of increments of control commands in the dynamics. INDI provides these increments of control commands as a function of the error of control variables, in contrast to NDI which provides a complete command control input instead.

For INDI-based control laws, and without loosing generality, we will consider a general MIMO system as in Eq. (8.11) which represent the dynamics to be inverted, and whose number of inputs are equal to the number of outputs, otherwise a control allocation problem would arise. Moreover, the output of the system, or the control variables vector, is considered to be the state, hence:

$$\mathbf{y} = \mathbf{x} \tag{8.16}$$

where $\mathbf{y} \in \mathscr{R}^m$ (of components $y_j = x_j$, here $m = n$) and $\mathbf{x} \in \mathscr{R}^n$ is the *state* vector of the system. The dynamics of this system can be expressed as:

$$\dot{\mathbf{y}} = \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\,\boldsymbol{u} \tag{8.17a}$$

where $\mathbf{u} \in \mathscr{R}^m$ is the *control command* vector (of components u_j), $\mathbf{f}$ is a smooth vector field on $\mathscr{R}^n$, and $\mathbf{G} \in \mathscr{R}^{n\times m}$ is a matrix whose columns are smooth vector fields $\mathbf{g}_j$. This

system represents the dynamics to be inverted with INDI-based control. Notice that this system has a relative degree of m since each of the controlled variables y_i has a relative degree $r_i = 1$, $i = \{1,\dots,m\}$, and possesses no internal dynamics. Also note that this general dynamic model of a MIMO nonlinear system is considered *affine* in the control command $\mathbf{u}$, as a result there is no need of nonlinear solvers for $\mathbf{u}$ when applying INDI to such system.

The approximated rotational dynamics of a rigid body in its incremental form [71, 73–75] is presented as follows. A standard Taylor series expansion provides the following first-order approximation of $\dot{\mathbf{x}}$, for $\mathbf{x}$ and $\mathbf{u}$ in the neighborhood of $[\mathbf{x}_0, \mathbf{u}_0]$:

$$\begin{aligned}\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\boldsymbol{u} &\cong \mathbf{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 + \frac{\partial}{\partial \mathbf{x}}\Big[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\Big]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{x}-\mathbf{x}_0) \\ &+ \frac{\partial}{\partial \mathbf{u}}\Big[\mathbf{G}(\mathbf{x})\mathbf{u}\Big]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{u}-\mathbf{u}_0) + \text{H.O.T} \\ &\cong \mathbf{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 + \frac{\partial}{\partial \mathbf{x}}\Big[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\ \mathbf{u}\Big]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{x}-\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\,(\mathbf{u}-\mathbf{u}_0)\end{aligned} \tag{8.18}$$

where the current state and control, $\mathbf{x}_0$ and $\mathbf{u}_0$ respectively, represent for each time instance the reference an incremental instance in time before $\mathbf{x}$ and $\mathbf{u}$ for the construction of the first-order approximation of $\dot{\mathbf{x}}$, and H.O.T the higher order terms that can be neglected further on. By definition, the corresponding state rate $\dot{\mathbf{x}}_0$ satisfies:

$$\dot{\mathbf{x}}_0 \equiv \mathbf{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 \tag{8.19}$$

Using this expression and the standard linear definition,

$$\mathbf{A}_0 = \frac{\partial}{\partial \mathbf{x}}\left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} \tag{8.20a}$$

$$\mathbf{B}_0 = \frac{\partial}{\partial \mathbf{u}}\left[\mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} = \mathbf{G}(\mathbf{x}_0) \tag{8.20b}$$

with $\mathbf{A}_0$ and $\mathbf{B}_0$ the partials evaluated at the current (reference) point $[\mathbf{x}_0, \mathbf{u}_0]$ on the state/-control trajectory; Equation (10.49a), i.e., the approximation of $\dot{\mathbf{x}}$ for $\mathbf{x}$ and $\mathbf{u}$ in the neighborhood of $[\mathbf{x}_0, \mathbf{u}_0]$, can be expressed as:

$$\dot{\mathbf{x}} \cong \dot{\mathbf{x}}_0 + \mathbf{A}_0\,(\mathbf{x}-\mathbf{x}_0) + \mathbf{B}_0\delta\mathbf{u} \tag{8.21}$$

where $\delta\mathbf{u} = (\mathbf{u}-\mathbf{u}_0)$ represents the incremental control command. This suggests that in a small neighborhood of the reference state we can approximate the nonlinear system (8.17) by its linearization about that reference state.

For the obtained approximation, input-output linearization is applied to obtain a relation between the incremental control command and the output of the system. Since each of the controlled variables y_i has a relative degree $r_i = 1$, $i = \{1,\dots,m\}$, the first order derivative of the output function represents an explicit relation between the output $\mathbf{y}$ and the input $\delta\mathbf{u}$:

$$\dot{\mathbf{y}} = \dot{\mathbf{x}} \cong \dot{\mathbf{x}}_0 + \mathbf{A}_0\,(\mathbf{x}-\mathbf{x}_0) + \mathbf{B}_0\delta\mathbf{u} \tag{8.22}$$

Inversion of this equation results in the linearizing input transformation:

$$\delta \mathbf{u} = \mathbf{B}_0^{-1}\left[\boldsymbol{\nu} - \left(\dot{\mathbf{x}}_0 + \mathbf{A}_0\left(\mathbf{x} - \mathbf{x}_0\right)\right)\right] \tag{8.23}$$

that can be applied to obtain the decoupled single-integrator linear relation $\dot{\mathbf{y}} = \boldsymbol{\nu}$. For small time increments and a sufficiently high control update rate, $\mathbf{x}$ approaches $\mathbf{x}_0$ which means that $\mathbf{A}_0\left(\mathbf{x} - \mathbf{x}_0\right) << \mathbf{B}_0 \delta \mathbf{u}$ and hence negligible. As a result, the linearizing control becomes:

$$\delta \mathbf{u} = \mathbf{B}_o^{-1}\left[\boldsymbol{\nu} - \dot{\mathbf{x}}_0\right] = \mathbf{G}(\mathbf{x}_0)^{-1}\left[\boldsymbol{\nu} - \dot{\mathbf{x}}_0\right] \tag{8.24}$$

and the linearized system:

$$\dot{\mathbf{y}} = \boldsymbol{\nu} \tag{8.25}$$

Note that this control law results in increments of control commands; these changes must be added to the current (reference) command to obtain the full new control command input. Also, note that $\dot{\mathbf{x}}_0$ is based on the reference command $\mathbf{u}_0$ and that $\delta \mathbf{u}$ is the commanded change that reference in order to generate $\boldsymbol{\nu}$, i.e., the desired first order derivatives of the controlled variables. Hence, the total control command is obtained as:

$$\mathbf{u} = \mathbf{u}_0 + \delta \mathbf{u} \tag{8.26}$$

Note that the existing $\mathbf{u}_0$ that corresponds to $\dot{\mathbf{x}}_0$ is taken from the output of the actuators, and it has been assumed that a commanded control is achieved instantaneously. The total control command (8.26) along with the obtained linearizing control $\delta \mathbf{u}$ can be rewritten as:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{G}(\mathbf{x}_0)^{-1}\left[\boldsymbol{\nu} - \dot{\mathbf{x}}_0\right] \tag{8.27}$$

INDI, as the application of NDI to a system expressed in an incremental form, results in a control law that is not depending on the exact knowledge of the plant dynamics $\mathbf{f}(\mathbf{x})$. The dependency of the closed-loop system on the model is largely decreased, improving the system robustness against model mismatch and model uncertainties. Moreover, changes in $\mathbf{f}(\mathbf{x})$ are reflected in $\dot{\mathbf{x}}_0$, and the control does require measurements of $\dot{\mathbf{x}}_0$ and $\mathbf{u}_0$, making this control strategy more dependent on the sensor measurements (the angular acceleration measurements in the case of attitude control). It is important to notice that this implicit control law design is not entirely independent on the model since changes in $\mathbf{f}(\mathbf{x})$ are reflected in measurements of $\dot{\mathbf{x}}_0$. However, this control law design is expected to be more dependent of sensor aspects (such as noise, bias, misalignment, etc.) than regular NDI.

The implementation of INDI-based control considers the following assumptions:

1. It is assumed to have complete and accurate knowledge about the state of the system. In practice however, state measurements may contain noise, biases, and delays. Moreover, angular acceleration sensors may exist but they are expensive and not common. As an alternative, angular acceleration may be derived from inertial measurement unit (IMU) gyro measurements;

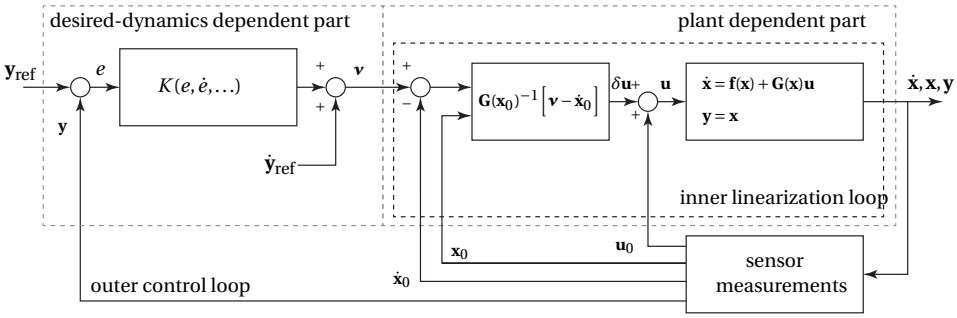


Figure 8.2: Tracking of a MIMO system with Incremental Nonlinear Dynamic Inversion. The inner linearization loop is based on Equation (10.55), whereas each channel $y_i - \nu_i$ of the outer control loop is based on linear control.

2. For small time increments, angular accelerations evolve faster than angular velocities upon control action, which directly influences the moment of the rigid body. In other words, the angular velocities only change by integrating angular accelerations, hence making the difference $(\boldsymbol{\omega} - \boldsymbol{\omega}_0)$ negligible for small time increments as compared to $\dot{\boldsymbol{\omega}}$;

3. Fast control action is assumed. This assumption complements the previous one in the sense that the dynamics of the actuators are considered to evolve much faster than the angular velocities. This assumption can be explained with singular perturbation theory or with time-scale separation, in which introducing the following slow and fast dynamics,

$$\dot{\boldsymbol{\omega}} = \mathbf{f}_1(\boldsymbol{\omega}) + \mathbf{g}_1(\boldsymbol{\omega})\mathbf{u} \tag{8.28}$$

$$\epsilon\ddot{\mathbf{u}} = \mathbf{f}_2(\boldsymbol{\omega}) + \mathbf{g}_2(\boldsymbol{\omega},\mathbf{u}) + \mathbf{g}_3(\boldsymbol{\omega},\mathbf{u},\dot{\mathbf{u}}) \tag{8.29}$$

making the constant $\epsilon > 0$ sufficiently small will make the controller state $\mathbf{u}$ evolve much faster than the slower state of the slow system, the angular velocities $\boldsymbol{\omega}$. Without loss of generality, a linear second order dynamics for the actuators can be assumed [192], and making ϵ sufficiently small in this context means making the actuator undamped natural frequency ω_{n_c} sufficiently high to guarantee the fast actuator requirement of incremental control.

In the case where actuator output measurements are not readily available, i.e. Fig.8.3-(a), a high-fidelity model of actuator dynamics can be included in the controller side as to supply the required control input reference $\mathbf{u}_0$, Fig.8.3-(b). Its mismatch with regards to reality must be studied in order to avoid a wind-up effect. Moreover, actuator output measurements may contain noise, biases, and delays. Of course, physical limitations exists and the attitude control system will depend on appropriate choice of sensors and actuators. Fig.8.3-(a) denotes a sensor-dependency configuration, where the actuator output measurements are readily available (e.g. reaction wheels, with a proper current-to-torque relation), and Fig.8.3-(b) denotes the model-dependent approach, where a

high-fidelity model of the actuator dynamics accompanies the control architecture (e.g. a high-fidelity model of reaction thrusters). In some cases, a combination of these two approaches may be necessary.

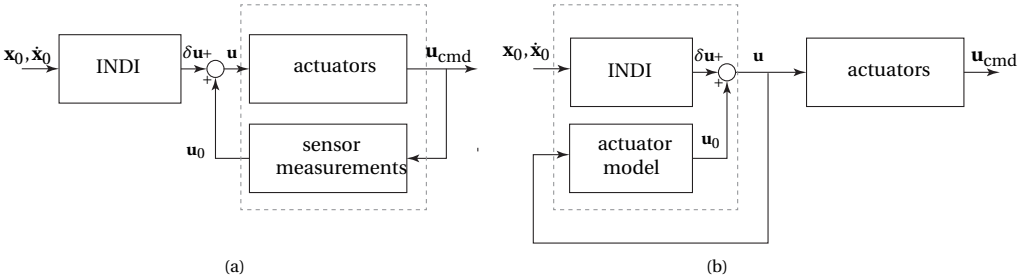


Figure 8.3: Actuator output block diagrams for Incremental Nonlinear Dynamic Inversion: (a) sensor-dependent. (b) model-dependent.

8.5. ATTITUDE CONTROL DESIGN

The interest is to find NDI-based feedback controllers to track an attitude motion reference of a rigid body. In this sense, we apply the results shown in the past sections to the attitude tracking problem. First, we demonstrate the Modified Rodrigues Parameters attitude tracking with NDI-control, followed by the proposed method using INDI-control.

8.5.1. RATE (INNER) CONTROL LOOP

The design objective of the rate (inner) control loop consists in the tracking of a desired angular velocity signal $\boldsymbol{\omega}_{\text{des}}$, obtained by the attitude (outer) control loop, with $\boldsymbol{\omega}(t)$, or similarly, the regulation of $\boldsymbol{\omega}_e(t)$: $\boldsymbol{\omega}_e(t) \to 0$ as $t \to \infty$ for all $\boldsymbol{\omega}_e(0)$ with $\boldsymbol{\omega}_e(t) = (\boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega})$, which must remain bounded. It should be clear that this loop involves the rotational dynamics of the rigid body, hence the controlled variables consist on the angular velocity vector:

$$\mathbf{y}_{\text{in}} = \boldsymbol{\omega} \tag{8.30}$$

Differentiation of this equation for NDI-control design results in the dynamics as in Eq. (8.6), system that can be expressed as:

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}) + \mathbf{G}\mathbf{u}, \quad \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0 \tag{8.31}$$

with:

$$\mathbf{f}(\boldsymbol{\omega}) = J^{-1} S(\boldsymbol{\omega}) J\boldsymbol{\omega} \qquad \mathbf{G} = J^{-1}$$

Recall $\mathbf{f}(\boldsymbol{\omega})$ to be the onboard model of the spacecraft which is necessary for dynamic inversion. Also $\mathbf{y}_{\text{in}}$ represents the control variables which hold a physical interpretation required for dynamic inversion. We can achieve this objective with NDI as follows. Recall the single integrator relation between the control torque and the angular velocities. Hence, the NDI control command is obtained as:

$$\mathbf{u} = \mathbf{G}^{-1}\Big[\boldsymbol{\nu}_{\text{in}} - \mathbf{f}(\boldsymbol{\omega})\Big] = J_{\text{n}}\Big[\boldsymbol{\nu}_{\text{in}} - J_{\text{n}}^{-1} S(\boldsymbol{\omega}) J_{\text{n}}\boldsymbol{\omega}\Big] \tag{8.32}$$

From this controller the following is expected:

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}) + \mathbf{G}\mathbf{u} \tag{8.33}$$

$$= \mathbf{f}(\boldsymbol{\omega}) + \mathbf{G}\left(\mathbf{G}^{-1}\left[\boldsymbol{\nu}_{\text{in}} - \mathbf{f}(\boldsymbol{\omega})\right]\right) \tag{8.34}$$

$$= \boldsymbol{\nu}_{\text{in}} \tag{8.35}$$

However, since in reality uncertainties are present, the system in Eq. (8.31) can be modeled as:

$$\dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}) + \Delta\mathbf{f}(\boldsymbol{\omega}) + \left[\mathbf{G} + \Delta\mathbf{G}\right]\mathbf{u} \tag{8.36}$$

where $\Delta\mathbf{G}$ may also capture control input uncertainty besides parametric uncertainty, and application of the linearizing control results in:

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \mathbf{f}(\boldsymbol{\omega}) + \Delta\mathbf{f}(\boldsymbol{\omega}) + \left[\mathbf{G} + \Delta\mathbf{G}\right]\left[\mathbf{G}^{-1}\left[\boldsymbol{\nu}_{\text{in}} - \mathbf{f}(\boldsymbol{\omega})\right]\right] \tag{8.37}$$

$$= \Delta\mathbf{f}(\boldsymbol{\omega}) + \boldsymbol{\nu}_{\text{in}} + \Delta\mathbf{G}\mathbf{G}^{-1}\boldsymbol{\nu}_{\text{in}} - \Delta\mathbf{G}\mathbf{G}^{-1}\mathbf{f}(\boldsymbol{\omega}) \tag{8.38}$$

$$= \left[\Delta\mathbf{f}(\boldsymbol{\omega}) - \Delta\mathbf{G}\mathbf{G}^{-1}\mathbf{f}(\boldsymbol{\omega})\right] + \left[\boldsymbol{I} + \Delta\mathbf{G}\mathbf{G}^{-1}\right]\boldsymbol{\nu}_{\text{in}} \tag{8.39}$$

Clearly the new system is not necessarily a linear system anymore because of the presence of uncertainties.

To overcome this situation INDI-control is now proposed. This method involves the use of (direct or estimated) angular acceleration and actuator output measurements to reduce the dependency on the onboard model of the spacecraft. The INDI control command is obtained as:

$$\mathbf{u} = \mathbf{u}_0 + \delta\mathbf{u} \tag{8.40}$$

Note that the existing $\mathbf{u}_0$ that corresponds to $\dot{\boldsymbol{\omega}}_0$ is considered in this case to be obtained from the actuator output measurements. The total control command (8.40) is hence obtained incrementally with the linearizing incremental $\delta\mathbf{u}$ which can be obtained by dynamic inversion as:

$$\delta\mathbf{u} = \mathbf{G}^{-1}\left(\boldsymbol{\nu}_{\text{in}} - \dot{\boldsymbol{\omega}}_0\right) = J_{\text{n}}\left(\boldsymbol{\nu}_{\text{in}} - \dot{\boldsymbol{\omega}}_0\right) \tag{8.41}$$

In this case, since the on-board model is replaced by angular acceleration measurements, the are only parametric uncertainties and the system in Eq. (8.31) can be modeled as:

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_0 + \left[\mathbf{G} + \Delta\mathbf{G}\right]\delta\mathbf{u} \tag{8.42}$$

The internal unknown dynamics and model uncertainties are captured by the angular acceleration measurements, then the application of the linearizing control results in:

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_0 + \left[\mathbf{G} + \Delta\mathbf{G}\right]\left[\mathbf{G}^{-1}\left(\boldsymbol{\nu}_{\text{in}} - \dot{\boldsymbol{\omega}}_0\right)\right] \tag{8.43}$$

$$= \boldsymbol{\nu}_{\text{in}} + \Delta\mathbf{G}\mathbf{G}^{-1}\boldsymbol{\nu}_{\text{in}} - \Delta\mathbf{G}\mathbf{G}^{-1}\dot{\boldsymbol{\omega}}_0 \tag{8.44}$$

$$= -\Delta\mathbf{G}\mathbf{G}^{-1}\dot{\boldsymbol{\omega}}_0 + \left[\boldsymbol{I} + \Delta\mathbf{G}\mathbf{G}^{-1}\right]\boldsymbol{\nu}_{\text{in}} \tag{8.45}$$

and using the notation $\mathbf{C} = \Delta\mathbf{G}\mathbf{G}^{-1}$:

$$\dot{\boldsymbol{\omega}} = -\mathbf{C}\dot{\boldsymbol{\omega}}_0 + \Big[\boldsymbol{I} + \mathbf{C}\Big]\boldsymbol{\nu}_{\text{in}} \tag{8.46}$$

which is a relation affine in the virtual control.

Using INDI for spacecraft attitude control only requires knowledge on inertia parameters since model and parametric uncertainty are captured by angular acceleration measurements or estimations. The resulting performance of these type of controllers is hence expected to be strongly dependent on the accuracy of these measurements.

Finally, in the nominal sense and for tuning and gain design purposes, the system:

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \boldsymbol{\nu}_{\text{in}} \tag{8.47}$$

can be designed to behave accordingly as desired. Without loss of generality, we use a simple P(I)-control structure for the virtual control law for the NDI as follows:

$$\boldsymbol{\nu}_{\text{in}} = \mathbf{K}_{p_{\text{in}}}\boldsymbol{\omega}_e + \mathbf{K}_{i_{\text{in}}}\int \boldsymbol{\omega}_e \mathrm{d}t = \mathbf{K}_{p_{\text{in}}}(\boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega}) + \mathbf{K}_{i_{\text{in}}}\int (\boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega})\mathrm{d}t \tag{8.48}$$

where $\boldsymbol{\omega}_{\text{des}}$ is obtained from the outer loop, and denoted NDI/P(I) control. For INDI, only the proportional gain is required as integral control action is implicit, hence denoted as INDI/P control.

To demonstrate the advantage of using INDI/P for rate control versus regular NDI/P(I) (or even linear P(I)-control), consider now the angular velocity tracking of a rigid spacecraft with inertia matrix

$$J_n = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 6.3 & 0 \\ 0 & 0 & 8.5 \end{bmatrix} \text{Kg}\cdot\text{m} \tag{8.49}$$

in presence of external disturbances (8.9), time-delay of 100 ms, and parametric uncertainties considered as follows: Δ_1 represent a 10% decrease (20% increase) in the (off-) diagonal terms, Δ_2 represent a 10% decrease (10% increase) in the (off-) diagonal terms, Δ_3 represent a 10% (10%) increase in the (off-) diagonal terms, and Δ_4 represent a 20% (20%) increase in the (off-) diagonal terms. For fair comparisons, the controller gains were obtained by minimizing the following quadratic performance index,

$$J_\pi = \sum_{k=0}^{k=t_f}\left(\sum_{i=1}^{3}\left\|\boldsymbol{\omega}_{e_i}(k)\right\|^2 + \left\|\mathbf{u}_i(k)\right\|^2\right) \tag{8.50}$$

which penalizes the angular velocity error and the control effort.

The results of the simulations for PI-control, NDI/P, NDI/PI, and INDI/P, for the rate loop tracking of a ramp smooth-command are presented in Figures 8.4, 8.5, 8.6, and 8.7, respectively. The results indicates that PI-control is not suitable for the combined effects considered (and leads to higher costs), NDI/P-control leads to steady state error, NDI/PI-control leads to higher costs and overshoot, and INDI/P rejects disturbances well and is insensitive to parametric uncertainty (without increasing too much the cost, however leading to chattering).

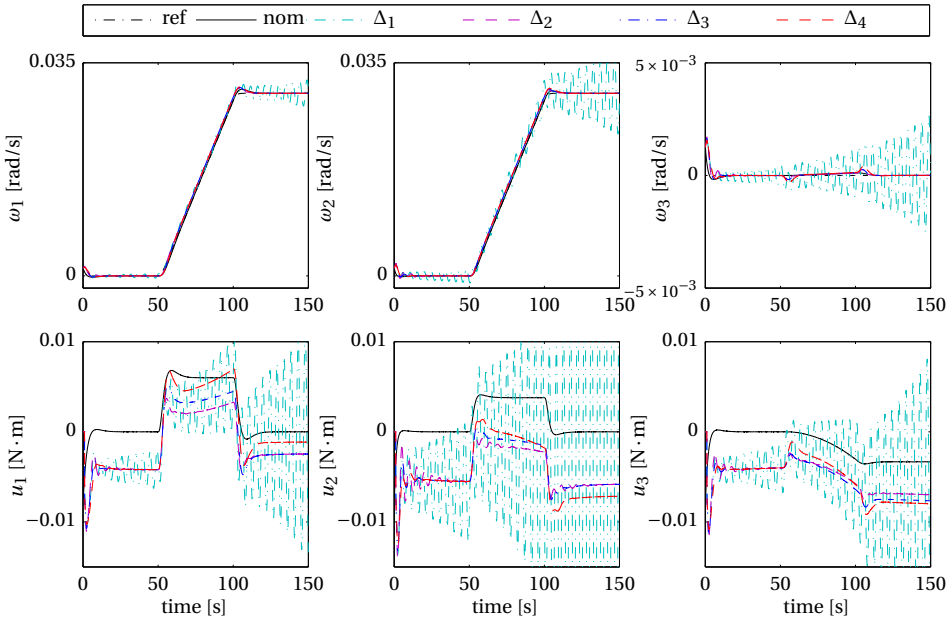


Figure 8.4: PI-control for rate (inner) control loop. $\mathbf{K}_p = 5.9$, $\mathbf{K}_i = 0.26$.

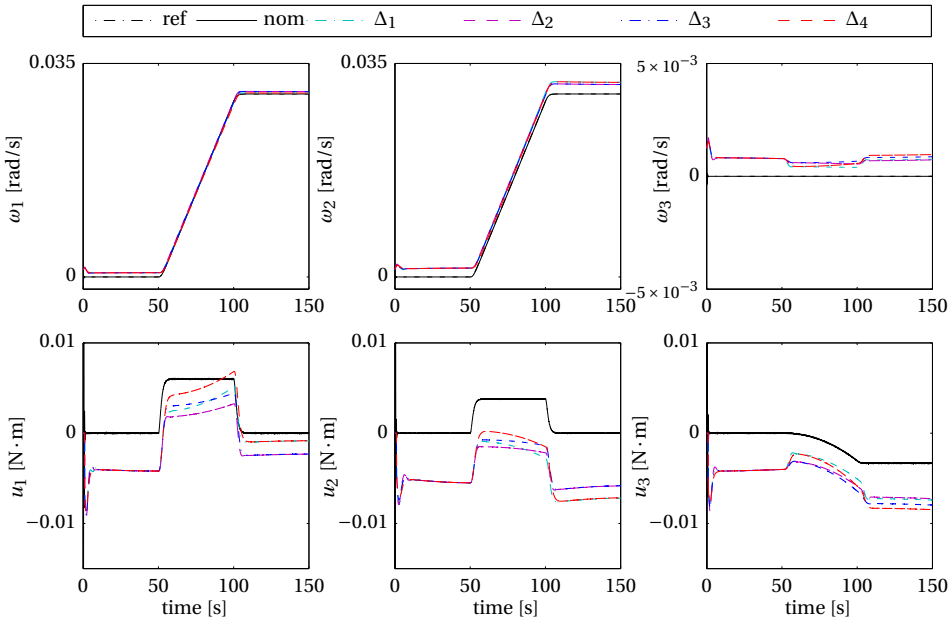


Figure 8.5: NDI P-control for rate (inner) loop. $\mathbf{K}_p = 0.6$.

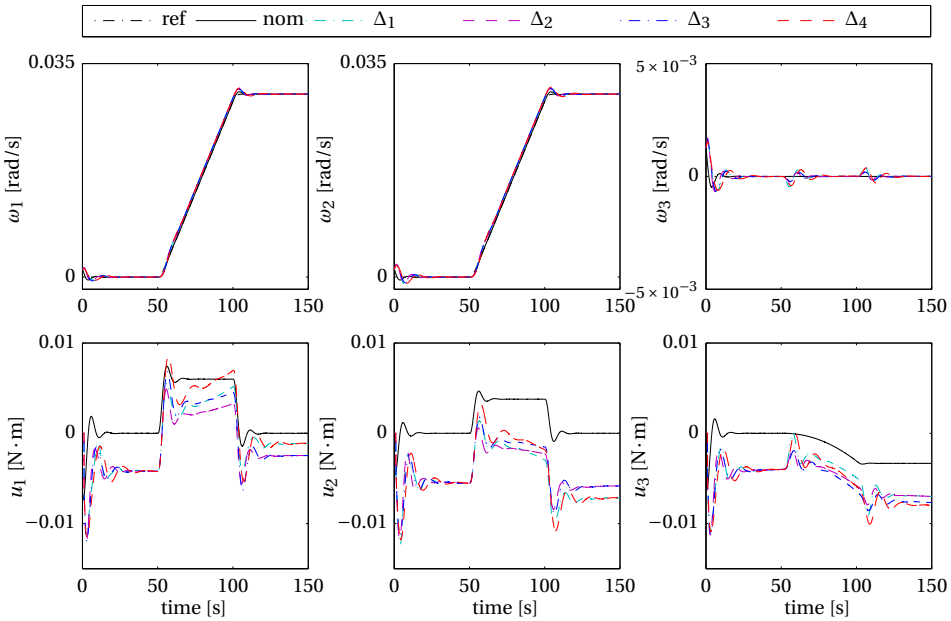


Figure 8.6: NDI PI-control for rate (inner) loop. $\mathbf{K}_p = 0.53$, $\mathbf{K}_i = 0.05$.

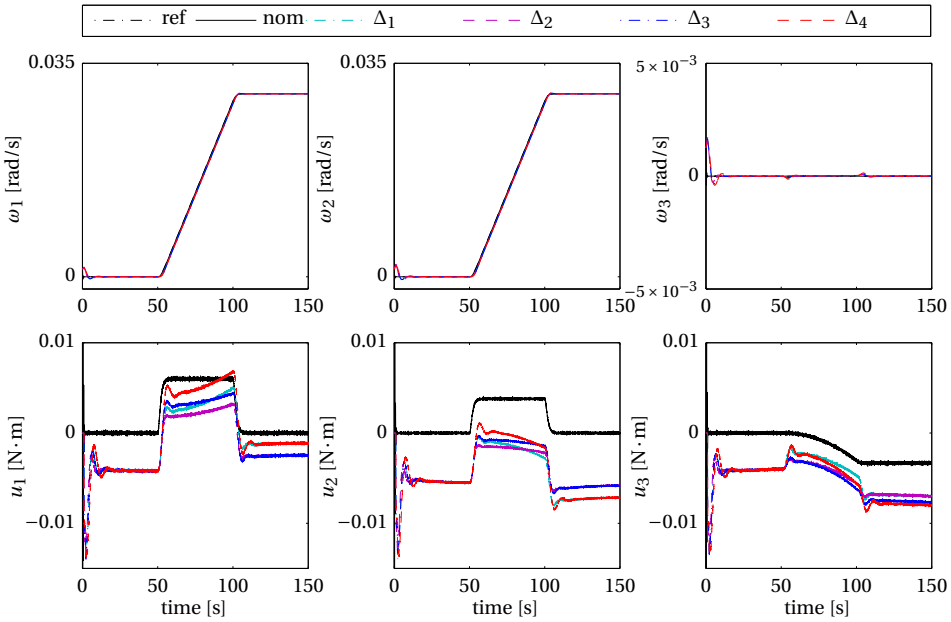


Figure 8.7: INDI P-control for rate (inner loop). $\mathbf{K}_p = 0.59$.

8.5.2. ATTITUDE (OUTER) LOOP

The design objective of the attitude (outer) control loop is twofold. It generates the desired angular velocity reference $\boldsymbol{\omega}_{\text{des}}$ for the inner loop, and tracks a desired attitude reference $\boldsymbol{\sigma}_{\text{des}}$ with $\boldsymbol{\sigma}(t)$. It should be clear that this loop involves the rotational kinematics of the rigid body, for this, consider the control variables:

$$\mathbf{y}_{\text{out}} = \boldsymbol{\sigma} \tag{8.51}$$

and the derivation of this equation for NDI design results in the kinematics:

$$\dot{\mathbf{y}}_{\text{out}} = \dot{\boldsymbol{\sigma}} = \mathbf{N}(\boldsymbol{\sigma})\boldsymbol{\omega}, \qquad \boldsymbol{\sigma}(0) = \boldsymbol{\sigma}_0 \tag{8.52}$$

where:

$$\mathbf{N}(\boldsymbol{\sigma}) = \frac{1}{2}\left[\boldsymbol{I} - S(\boldsymbol{\sigma}) + \boldsymbol{\sigma}\boldsymbol{\sigma}^{\top} - \frac{1}{2}\left(1 + \boldsymbol{\sigma}^{\top}\boldsymbol{\sigma}\right)\boldsymbol{I}\right] \tag{8.53}$$

and the $S(\cdot)$ denotes the 3×3 antisymmetric matrix as in (8.3).

Since this kinematic equation is a given geometrical representation, it does not involves the presence of any uncertainty. Hence, NDI-control is sufficient and this command is obtained as:

$$\boldsymbol{\omega}_{\text{des}} = \mathbf{N}(\boldsymbol{\sigma})^{-1}\left[\boldsymbol{\nu}_{\text{out}}\right] \tag{8.54}$$

and

$$\boldsymbol{\nu}_{\text{out}} = \mathbf{K}_{p_{\text{out}}}(\boldsymbol{\sigma}_e) + \mathbf{K}_{i_{\text{out}}}\int \boldsymbol{\sigma}_e \mathrm{d}t \tag{8.55}$$

where $\boldsymbol{\sigma}_e$ is the Modified Rodrigues Parameter error. The complete attitude control diagram is presented in Figure 8.8.

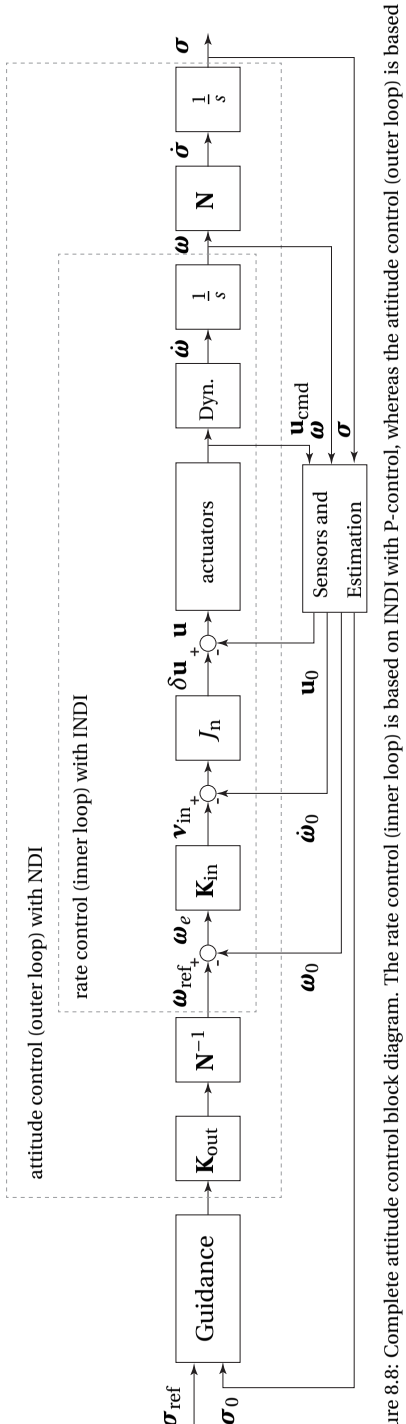


Figure 8.8: Complete attitude control block diagram. The rate control (inner loop) is based on INDI with P-control, whereas the attitude control (outer loop) is based on NDI with P(I)-control.

8.6. SIMULATION

The theoretical results presented and the controllers derived are now demonstrated by a numerical simulation performed in MATLAB&SIMULINK. The simulation compares a complete attitude control using NDI/P and INDI/P, Eqs. (8.32) and (8.40), for the rate control loop (using NDI/PI for the rate loop in combination with the outer loop leads to instabilities), together with NDI/P(I) control for the outer loop. In this sense, the controllers will be referred as NDI/P-P, NDI/PI-P (NDI/P for the inner loop plus NDI/PI for the outer loop), and INDI/P-P respectively. The dynamic inversion controllers will be applied for the attitude tracking of the Modified Rodrigues Parameters of a rigid spacecraft with inertia matrix (8.49). The effect of noise, external disturbances, measurement time-delays, and model and parametric uncertainties will be considered together for the controllers proposed as to an idea of the performance under these situations.

For this particular application, the reference attitude will be given by $\boldsymbol{\sigma}_c$ as a doublet rest-to-rest two-axis re-orientation maneuver. The maneuver consist of an Eigen-axis rotation of $\theta = \pi/12$ rad at $t = 50$ s. Zero-mean Gaussian white-noise is added to the closed-loop system and considered with standard deviations $s_{d_\sigma} = 1 \times 10^{-3}$ for $\boldsymbol{\sigma}$, and $s_{d_\omega} = 1 \times 10^{-6}$ rad/s for $\boldsymbol{\omega}$. The attitude measurements are sampled at 1 Hz, and the angular velocities at 100 Hz. The angular accelerations are estimated as:

$$\dot{\boldsymbol{\omega}}[k] = \frac{\boldsymbol{\omega}[k] - \boldsymbol{\omega}[k-1]}{\mathrm{d}t} \tag{8.56}$$

and filtered appropriately. Moreover, in practical applications model uncertainties and discrepancies exist; the mass properties of the spacecraft may be uncertain or may change due to motion of onboard payload, rotation of solar arrays, liquid sloshing, etc. In this chapter however, the focus is given to constant but uncertain inertia matrix. To this end, consider the inertia matrix represented by:

$$J = J_n + \Delta J \tag{8.57}$$

where J_n and ΔJ are the nominal part and the uncertain part of J, respectively.

The uncertain part of the inertia matrix will be modeled as:

$$\Delta J = \begin{bmatrix} \Delta_1 & \Delta_4 & \Delta_5 \\ \Delta_4 & \Delta_2 & \Delta_6 \\ \Delta_5 & \Delta_6 & \Delta_3 \end{bmatrix} = \begin{bmatrix} 0.2J_{11} & 0.2J_{11} & 0.2J_{22} \\ 0.2J_{11} & 0.2J_{22} & 0.2J_{33} \\ 0.2J_{22} & 0.2J_{33} & 0.2J_{33} \end{bmatrix} \tag{8.58}$$

The results of the simulation are shown in Figures 8.9, 8.10, 8.11, 8.12, and 8.13. The solid green lines represent the trajectories with NDI/P-P control, the dashed red lines represent the trajectories with NDI/PI-P control, the dashed blue lines represent the trajectories with INDI/P-P control, and the dashed black lines represent the reference for this particular maneuver. Figure 8.9 depicts the nominal behavior of the Modified Rodrigues Parameter vector, the angular velocities, and the associated control effort. Figure 8.10 depicts the behavior in the presence of external disturbance modeled as in Eq. (8.9). Figure 8.11 depicts the behavior in the presence of a measurement time-delay of 100 ms.

Figure 8.12 depicts the behavior in the presence of parametric uncertainty modeled as in Eq. (8.58). Finally, Figure 8.12 combines all these situations together. Namely, the presence of external disturbance, measurement time-delays, and parametric uncertainty.

Notice that:

1. In the nominal sense, NDI/P-P and INDI/P-P performs identically as expected;
2. The presence of external disturbance is not fully rejected with NDI/P-P but with NDI/PI-P, at the expense of a higher overshoot. However, INDI/P-P performs better at thistask fully rejecting the disturbance without compromising severely nominal performance;
3. Parametric uncertainty degrades nominal performance in general, but less so for the INDI/P-P control. This is due to the feedback of angular accelerations which captures the mismatch with the model;
4. Combining the effect of external disturbance, measurement time-delays, and parametric uncertainty demonstrates the full capabilities of INDI control. It is shown that INDI/P-P performs well under these considerations, and the performance and trajectories are not so degraded as for the other controllers. This demonstrates the robustness capabilities of such controllers.

8.7. CONCLUSIONS

This chapter presented an application of the Incremental Nonlinear Dynamic Inversion (INDI) control methodology to the attitude tracking and disturbance rejection problem of rigid spacecraft in presence of model and parametric uncertainties. As a modification of the NDI methodology, the INDI approach enhances its robustness capabilities by reducing feedback control dependency on accurate knowledge of the system dynamics. The use of incremental control action, which requires information of actuator output and angular accelerations, make these sensor-based type of controllers efficient for external disturbance rejection and robust in terms of handling uncertainties. Unlike NDI, this control design technique is implicit in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed. Under the influence of external disturbances, time-delay, and parametric uncertainty, it was shown that incremental nonlinear dynamic inversion performs better than regular NDI and PI-control without compromising nominal performance and stability. However, in practice, INDI-based control rely on accurate actuator output and angular acceleration measurements which may not be readily available or which may contain noise, biases and delays, hence their effect need to be further studied.

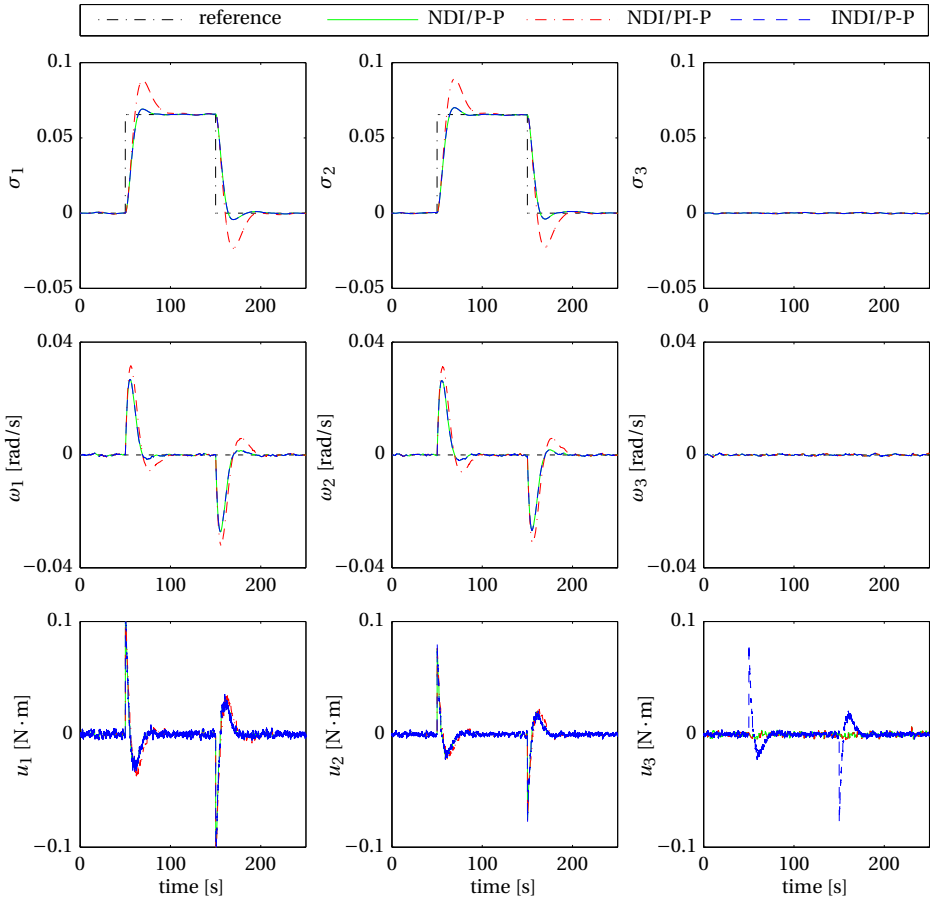


Figure 8.9: Comparison of the nominal attitude tracking of the Modified Rodrigues Parameters ($\boldsymbol{\sigma}$), the angular velocities ($\boldsymbol{\omega}$), and the control effort ($\mathbf{u}$), respectively, for three different controllers: the NDI/P-P control, the NDI/PI-P control, and the INDI/P-P control.

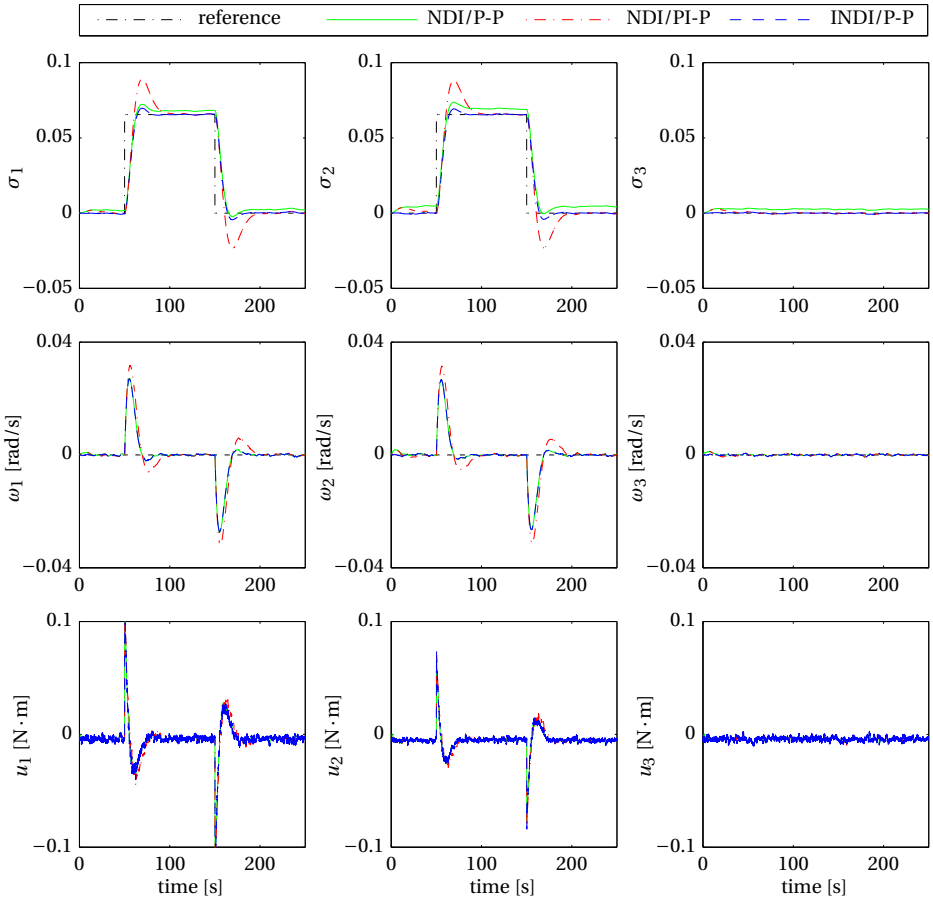


Figure 8.10: Comparison of the attitude tracking of the Modified Rodrigues Parameters ($\boldsymbol{\sigma}$), the angular velocities ($\boldsymbol{\omega}$), and the control effort ($\mathbf{u}$), respectively, in the presence of external disturbance for three different controllers: the NDI/P-P control, the NDI/PI-P control, and the INDI/P-P control.

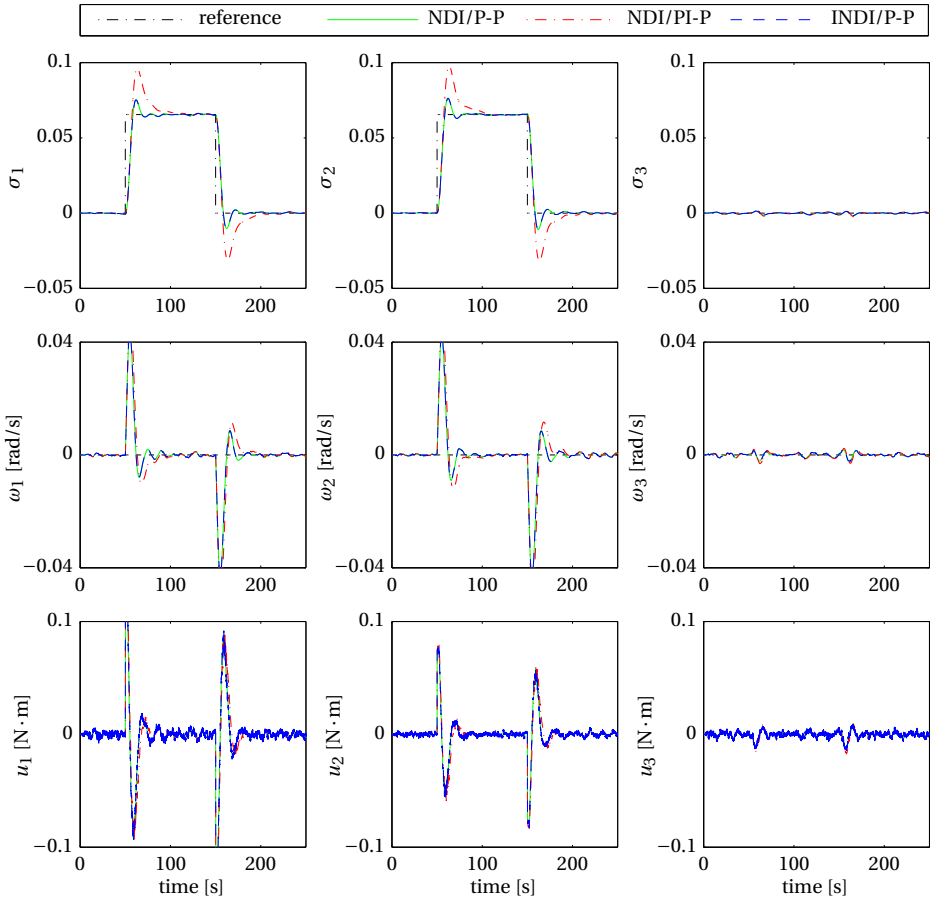


Figure 8.11: Comparison of the attitude tracking of the Modified Rodrigues Parameters ($\boldsymbol{\sigma}$), the angular velocities ($\boldsymbol{\omega}$), and the control effort ($\mathbf{u}$), respectively, in the presence of measurement time-delay of 100 ms for three different controllers: the NDI/P-P control, the NDI/PI-P control, and the INDI/P-P control.

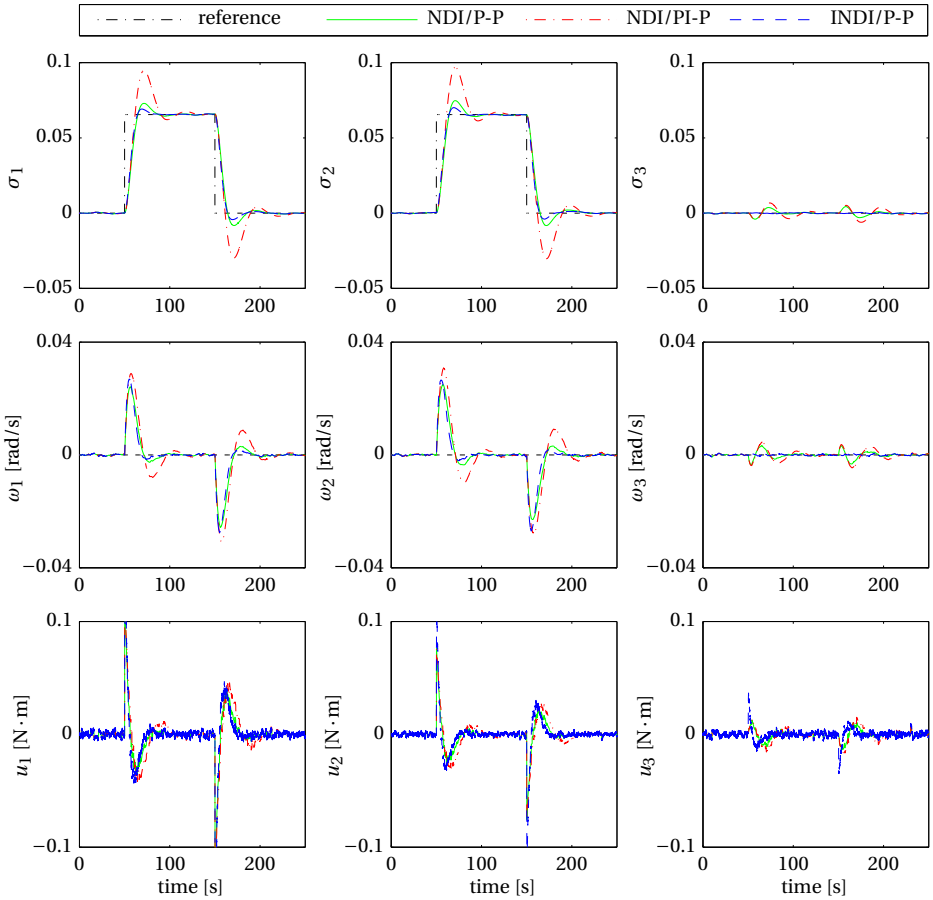


Figure 8.12: Comparison of the attitude tracking of the Modified Rodrigues Parameters ($\boldsymbol{\sigma}$), the angular velocities ($\boldsymbol{\omega}$), and the control effort ($\mathbf{u}$), respectively, in the presence of parametric uncertainty for three different controllers: the NDI/P-P control, the NDI/PI-P control, and the INDI/P-P control.

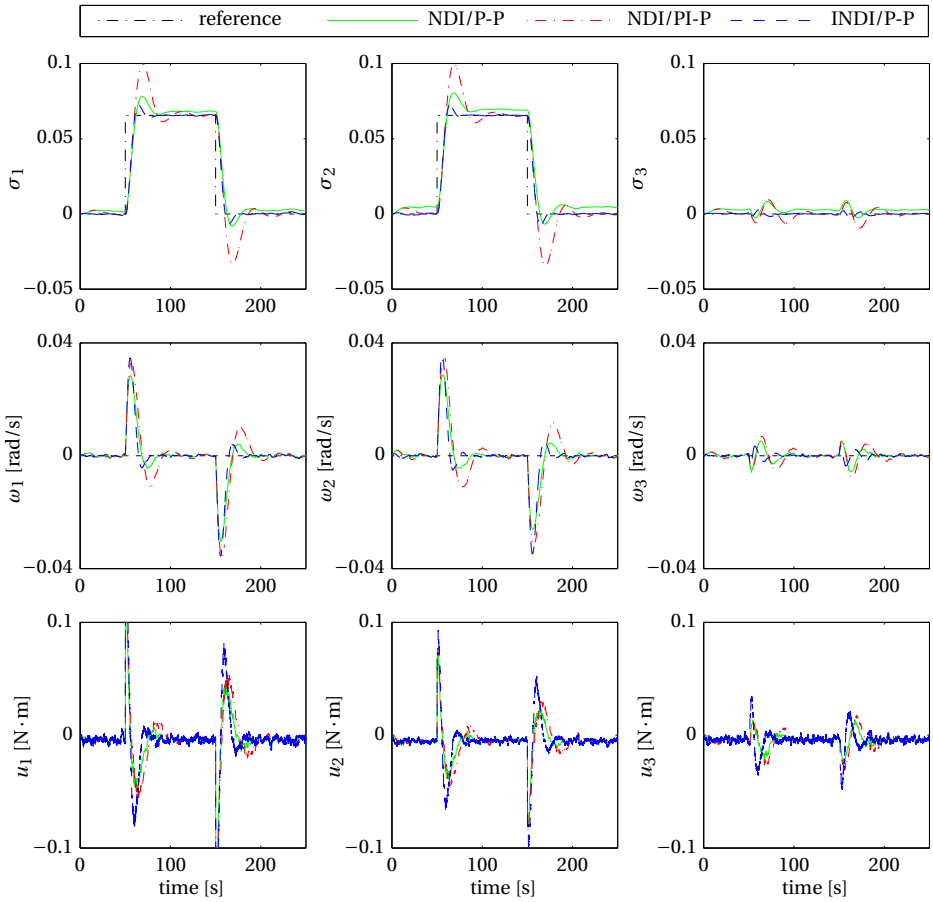


Figure 8.13: Comparison of the attitude tracking of the Modified Rodrigues Parameters ($\boldsymbol{\sigma}$), the angular velocities ($\boldsymbol{\omega}$), and the control effort ($\mathbf{u}$), respectively, in the presence of external disturbance, measurement time-delay of 100 ms, and parametric uncertainty for three different controllers: the NDI/P-P control, the NDI/PI-P control, and the INDI/P-P control.

9

Agile Spacecraft Attitude Control: an Incremental Nonlinear Dynamic Inversion Approach

Abstract

This chapter presents an agile and robust spacecraft attitude tracking controller using the recently reformulated incremental nonlinear dynamic inversion (INDI). INDI is a combined model– and sensor–based control approach that only requires a control effectiveness model and measurements of the state and some of its derivatives, making a reduced dependency on exact system dynamics knowledge. The reformulated INDI allows a non-cascaded dynamic inversion control in terms of Modified Rodrigues Parameters (MRPs) where scheduling of the time-varying control effectiveness is done analytically. This way, the controller is only sensitive to parametric uncertainty of the augmented spacecraft inertia and its wheelset alignment. Moreover, we draw some parallels to time-delay control (TDC) —more familiar in the robotics community— which have been shown to be equivalent to the incremental formulation of proportional-integral-derivative (PID) control for second order nonlinear systems in controller canonical form. Simulation experiments for this particular problem demonstrate that INDI has similar nominal performance as TDC/PID control, but superior robust performance and stability.

Publication

Paul Acquatella B., Qi Ping Chu: *Agile Spacecraft Attitude Control: an Incremental Nonlinear Dynamic Inversion Approach.* In: IFAC-PapersOnLine, Vol. 53, No. 2, 2020; presented at *IFAC-WC 2020, 21st World Congress of the International Federation of Automatic Control, July 12-17, 2020.*

9.1. INTRODUCTION

FUTURE satellite systems are expected to be more performant not only for fine pointing capabilities in data acquisition but also in terms of high agility for maneuverability [33]. This emerging field of *agile Earth Observation* motivated the development of a high-agility attitude control system for the the satellite platform BIROS (*Bispectral InfraRed Optical System*) while actuated with a redundant array of *High-Torque-Wheels (HTW)* [193].

The topic of optimal and agile spacecraft rotational maneuvers is quite extensive and has been studied for many decades [127, 130, 132]. However, most of the work reported in literature relies on optimization and some form of trajectory optimization, which might be difficult to implement on-board. In this chapter, we are motivated to find an agile attitude control solution in closed-loop feedback form. This is challenging because of the many nonlinearities involved.

Incremental nonlinear dynamic inversion (INDI) has been proposed as a promising sensor-based approach providing high performance and robust nonlinear control for aerospace vehicles without requiring a detailed model of the controlled plant. The INDI approach reduces its dependency on onboard or baseline models while making use of actuator output and angular acceleration measurement feedback. Theoretical development of increments of nonlinear control action date back from the late nineties by [70, 73] which were further developed as 'incremental NDI' [71, 77, 79, 80] for flight control as well as for spacecraft attitude control [4]. More recently, this technique has been applied also in practice for quadrotors using adaptive control by [85], and in real flight tests by [88, 89], verifying its performance and robustness properties against aerodynamic model uncertainties and disturbance rejection.

INDI relies on the assumption that for small time increments and high sampling rates, the nonlinear system dynamics in its incremental form is simply approximated by the (linearized) control effectiveness evaluated at the current state. Recently, the INDI control in the literature has been reformulated for systems with arbitrary relative degree and without recurring to cascaded-control structures, i. e., without using a time–scale separation assumption [97]. This reformulation allowed to extended further the incremental nonlinear control approach for Sliding Mode Control by [99]. For these new reformulations and extensions, conditions for stability and robustness analyses have been established and analyzed using Lyapunov-based methods. Another nonlinear control method is *time–delay–control* (TDC) [140–142], more commonly known in the motion control and robotics community and pioneered in the 90's by the works of Hsia, Youcef-Toumi, *et al.* [140]. TDC works by estimating and compensating disturbances and system uncertainties (model and parametric) by utilizing time–delayed signals of some of the system variables.

In this chapter, we present three main contributions in the context of nonlinear spacecraft attitude control system design. 1) We consider the reformulated INDI control for the spacecraft attitude control problem where input-output linearization is done without the usual time scale separation principle. 2) We revisit the reformulated INDI for the attitude control problem and introduce a time–delay explicitly in this reformulation. 3) We revisit TDC and establish the relationship and condition for equivalence between INDI and TDC. Based on previous results reported in the robotics literature showing the

relationship between discrete formulations of TDC and the incremental formulation of *proportional-integral-derivative control* (PID) control, we also establish a clear relationship between INDI and nonlinear-PID control.

9.2. MODELING OF SPACECRAFT WITH REACTION WHEELS

First we describe the comprehensive nonlinear rotational dynamics model for spacecraft including a generic set of reaction wheels as shown in [138, 193]. In this chapter, we consider the *Modified Rodrigues Parameters* (MRPs) [188, 194] as they represent a well defined attitude parameterization for all Eigen-axis rotations in the large domain of $0° \leq \theta < 360°$ where θ is the principal angle rotation around the Euler-axis λ. The MRP attitude is a suitable kinematic parameterization given their potential advantages for spacecraft attitude control [188, 194].

9.2.1. KINEMATICS

Consider first an array consisting of n reaction wheels. Introducing unit vectors $\mathbf{a}_i$ which give the orientation of the spin-axis of each reaction wheel with respect to the spacecraft coordinate system, these are collected in the configuration or alignment matrix $\boldsymbol{A} = [\,\mathbf{a}_1 \quad \ldots \quad \mathbf{a}_n\,]$. In that sense, the kinematics of the $i-$th reaction wheel in terms of its spin-axis angle Φ_w and angular velocity Ω_w, is simply given by $\dot{\Phi}_{w,i} = \Omega_{w,i},\ i = 1,\ldots,n$. The MRP vector $\boldsymbol{\sigma}$ is defined in relation to the Euler-axis λ and principal angle rotation θ as $\boldsymbol{\sigma} = \lambda \tan(\theta/4)$ [194], and the kinematic differential equation relating $\boldsymbol{\sigma}$ with the spacecraft angular velocity $\boldsymbol{\omega} \in \mathscr{R}^3$ (with respect to the body fixed frame) in vector form is given by [194] as

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[(1-\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{I}_{3\times3} + 2S(\boldsymbol{\sigma}) + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top\right]\boldsymbol{\omega} = \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\,\boldsymbol{\omega} \tag{9.1}$$

where $S(\cdot)$ is defined such that $S(x)\ y = x \times y$ for any $x, y \in \mathscr{R}^3$. Moreover, in this chapter we will also interested on the exact relation [194]

$$\ddot{\boldsymbol{\sigma}} = \frac{1}{4}\left[\ \dot{\boldsymbol{B}}(\boldsymbol{\sigma})\cdot\boldsymbol{\omega} + \boldsymbol{B}(\boldsymbol{\sigma})\cdot\dot{\boldsymbol{\omega}}\ \right] = \frac{1}{4}\boldsymbol{C}(\boldsymbol{\sigma},\boldsymbol{\omega},\dot{\boldsymbol{\omega}}) \tag{9.2}$$

where

$$\dot{\boldsymbol{B}}(\boldsymbol{\sigma})\cdot\boldsymbol{\omega} = \frac{1}{2}\left[\ 2\boldsymbol{\sigma}^\top\boldsymbol{\omega}(1-\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega} - (1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega}^\top\boldsymbol{\omega}\boldsymbol{\sigma} - 4\boldsymbol{\sigma}^\top\boldsymbol{\omega}S(\boldsymbol{\omega})\boldsymbol{\sigma} + 4(\boldsymbol{\sigma}^\top\boldsymbol{\omega})^2\boldsymbol{\sigma}\ \right]$$

which relates the MRP "acceleration" $\ddot{\boldsymbol{\sigma}}$ to the rigid body's angular velocity $\boldsymbol{\omega}$ and angular acceleration $\dot{\boldsymbol{\omega}}$. This relationship will be key for the attitude control design as it will be clear later on.

9.2.2. DYNAMICS

Following the derivations in [138], we obtain the rotational dynamics model as follows. First, consider the angular momentum of the spacecraft equipped with the reaction wheel array in question

$$\boldsymbol{H} = \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{h} \tag{9.3}$$

where, expressed in body-fixed frame, $\boldsymbol{H} \in \mathscr{R}^3$ is the total angular momentum of the system; $\boldsymbol{I} \in \mathscr{R}^{3\times3}$ is the constant inertia matrix of the spacecraft including the reaction wheels; $\boldsymbol{\omega} \in \mathscr{R}^3$ is the spacecraft angular velocity; and $\boldsymbol{h} \in \mathscr{R}^3$ is the total angular momentum vector associated with the reaction wheel array. The angular momentum $\boldsymbol{h}$ can be expressed from individual actuator frames to body-fixed frame as

$$\boldsymbol{h} = \sum_{i=1}^{n} \mathbf{a}_i \boldsymbol{h}_{w,i} = A\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}, \tag{9.4}$$

where $\boldsymbol{I}_{\boldsymbol{w}} = \text{diag}[\,\boldsymbol{I}_{w,1} \quad \dots \quad \boldsymbol{I}_{w,n}\,]$ is a diagonal matrix of reaction wheel spin-axis inertia values and $\boldsymbol{\Omega} = \Omega_w + A^\top\boldsymbol{\omega}$ the inertial angular rate of the reaction wheel array, where the term $A^\top\boldsymbol{\omega}$ is the extra angular motion relative to the spacecraft. Considering the angular momentum associated with the $i-$th reaction wheel in actuator frame

$$\boldsymbol{h}_{w,i} = \boldsymbol{I}_{w,i}\left(\,\Omega_{w,i} + \mathbf{a}_i^\top\boldsymbol{\omega}\,\right), \qquad i = 1,\dots,n, \tag{9.5}$$

we can already obtain the differential equation describing the reaction wheel dynamics in terms of reaction wheel torques $\tau_{w,i}$, which are considered as the exogenous inputs to the system provided by the wheel's powertrain

$$\dot{\Omega}_{w,i} = \boldsymbol{I}_{w,i}^{-1}\tau_{w,i} - \mathbf{a}_i^\top\dot{\boldsymbol{\omega}}, \qquad i = 1,\dots,n. \tag{9.6}$$

Because the angular momentum must be conserved in the absence of external perturbations, applying the transport theorem [127, 138]) to Eq. (10.7), the following relation is obtained

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H} + \boldsymbol{\omega}\times\boldsymbol{H} = \mathbf{0}. \tag{9.7}$$

Combining Eqs. (10.8), (10.12), and (9.7), the comprehensive nonlinear model for spacecraft dynamics equipped with reaction wheels [138] is given by

$$\boldsymbol{\Gamma}\begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\Omega}_{w,1} \\ \vdots \\ \dot{\Omega}_{w,n} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}\times\left(\boldsymbol{I}\boldsymbol{\omega} + A\boldsymbol{I}_{\mathbf{w}}\Omega_w + A\boldsymbol{I}_{\mathbf{w}}A^\top\boldsymbol{\omega}\right) \\ \tau_{w,1} \\ \vdots \\ \tau_{w,n} \end{bmatrix} \tag{9.8}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{I} + A\boldsymbol{I}_{\boldsymbol{w}}A^\top & \mathbf{a}_1\boldsymbol{I}_{w,1} & \cdots & \mathbf{a}_n\boldsymbol{I}_{w,n} \\ \boldsymbol{I}_{w,1}\mathbf{a}_1^\top & \boldsymbol{I}_{w,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_{w,n}\mathbf{a}_n^\top & 0 & \cdots & \boldsymbol{I}_{w,n} \end{bmatrix}$$

is an augmented inertia coupling matrix for the full system.

9.2.3. FULL NONLINEAR SPACECRAFT MODEL

The augmentation of the nonlinear spacecraft dynamics model together with the MRP kinematics can be rewritten as a full model in the generic form of affine n-dimensional multivariable nonlinear system with m inputs u_i and m outputs y_i

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{9.9a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{9.9b}$$

where $\boldsymbol{x} \in \mathscr{R}^n$, $\boldsymbol{u} \in \mathscr{R}^m$, and $\boldsymbol{y} \in \mathscr{R}^p$. The functions $\boldsymbol{f}$, $\boldsymbol{g}$, and $\boldsymbol{h}$ are assumed to be smooth vector fields continuously differentiable on $\mathscr{R}^n$. Moreover, the system has a vector of relative degree of $[\,\rho_1 \quad \dots \quad \rho_p\,]^\top$ which represents the number of differentiation of each output y_i $(i = 1,\dots,p)$ needed for the input to appear [61], and the total relative degree is obtained as $\rho = \rho_1 + \cdots + \rho_p$. In this chapter we consider the output MRP as control variables $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\sigma}$, and assume to have three reaction wheels ($n_w = 3$) as actuators, hence $\boldsymbol{u} = [\,\tau_{w,1} \quad \tau_{w,2} \quad \tau_{w,3}\,]^\top$ and $p = m = 3$. Whenever $p < m$, the input-output linearization is not straightforward and some form of control allocation is required. Else, when $p > m$, the control problem is underactuated and the input-output linearization is underdetermined. These aspects are however out of the scope of this chapter. Considering the vector $\boldsymbol{x} = [\,\boldsymbol{\sigma} \quad \boldsymbol{\omega} \quad \boldsymbol{\Omega}_w\,]^\top$ with, respectively, $\boldsymbol{\sigma} = [\,\sigma_1 \quad \sigma_2 \quad \sigma_3\,]^\top$, $\boldsymbol{\omega} = [\,\omega_x \quad \omega_y \quad \omega_z\,]^\top$, and $\boldsymbol{\Omega}_w = [\,\Omega_{w,1} \quad \Omega_{w,2} \quad \Omega_{w,3}\,]^\top$, the full nonlinear system dynamics in (10.16) is obtained with the functions given as

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\,\boldsymbol{\omega} \\ \boldsymbol{\Gamma}^{-1}\begin{bmatrix} -\boldsymbol{\omega} \times \left(\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{A}\boldsymbol{I}_{\mathbf{w}}\boldsymbol{\Omega}_w + \boldsymbol{A}\boldsymbol{I}_{\mathbf{w}}\boldsymbol{A}^\top\boldsymbol{\omega}\right) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix}, \tag{9.10a}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \boldsymbol{\Gamma}^{-1}\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{1}_{3\times 3} \end{bmatrix} \end{bmatrix} = \boldsymbol{G}. \tag{9.10b}$$

9.3. INCREMENTAL NONLINEAR DYNAMIC INVERSION

9.3.1. NONLINEAR DYNAMIC INVERSION CONTROL

Finding an explicit relationship between the input and the output of the system is generally not straightforward because they are not directly related. First, recall

$$\boldsymbol{l}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{L}_f^{\rho_1} h_1(\boldsymbol{x}) \\ \vdots \\ \boldsymbol{L}_f^{\rho_m} h_m(\boldsymbol{x}) \end{bmatrix} \tag{9.11a}$$

$$\boldsymbol{M}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{L}_{g_1}\boldsymbol{L}_f^{\rho_1-1} h_1(\boldsymbol{x}) & \cdots & \boldsymbol{L}_{g_m}\boldsymbol{L}_f^{\rho_1-1} h_1(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{L}_{g_1}\boldsymbol{L}_f^{\rho_m-1} h_m(\boldsymbol{x}) & & \boldsymbol{L}_{g_m}\boldsymbol{L}_f^{\rho_m-1} h_m(\boldsymbol{x}) \end{bmatrix}, \tag{9.11b}$$

where $\boldsymbol{L}_f^{\rho_j} h_j(\boldsymbol{x})$ and $\boldsymbol{L}_{g_i}\boldsymbol{L}_f^{\rho_j-1} h_j(\boldsymbol{x})$ are Lie derivatives of the scalar functions $h_j(\boldsymbol{x})$ with respect to the vectors $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_i(\boldsymbol{x})$, with j, $i = 1$ to m. Denoting the differentiated outputs $\boldsymbol{\zeta} = [\,y_1^{\rho_1-1} \quad \dots \quad y_m^{\rho_m-1}\,]^\top$, the following relation is obtained

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}. \tag{9.12}$$

Denoting $\boldsymbol{\nu}$ as a virtual control input, the vector $\boldsymbol{\varphi}(\boldsymbol{x}) = -\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{l}(\boldsymbol{x})$, and the matrix $\boldsymbol{\vartheta}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})$, then the state feedback control law $\boldsymbol{u}$ defined as

$$\boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{x}) + \boldsymbol{\vartheta}(\boldsymbol{x})\boldsymbol{\nu} \tag{9.13}$$

cancels all nonlinearities in closed-loop, and a simple linear input-output relationship between the new input $\boldsymbol{\nu}$ and the new output $\boldsymbol{\zeta}$ is obtained

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{\nu} \tag{9.14}$$

as long as $\boldsymbol{\vartheta}$ is not singular. Apart from being linear, an interesting result is that the input ν_i only affects the differentiated output ζ_i (decoupled). From this fact, the input transformation (10.70) is called a *decoupling control law*, and the resulting linear system (9.14) is called the *single-integrator* form. The single-integrator form (9.14) is sought to be rendered exponentially stable with the proper design of of $\boldsymbol{\nu}$. From this typical tracking problem it can be seen that the entire control system will have two control loops [71, 79]: the inner linearization loop (10.70), and the outer control loop (9.14). This resulting NDI control law depends on accurate knowledge of the model ($\boldsymbol{l}(\boldsymbol{x})$ and $\boldsymbol{M}(\boldsymbol{x})$) and its parameters, hence it is susceptible to model and parametric uncertainties. For that reason we are now interested in the concept of incremental nonlinear dynamic inversion.

9.3.2. INCREMENTAL NONLINEAR DYNAMIC INVERSION CONTROL

The concept of incremental nonlinear dynamic inversion (INDI) amounts to the application of NDI to a system expressed in an incremental form. This improves the robustness of the closed-loop system as compared with conventional NDI since dependency on the accurate knowledge of the plant dynamics is reduced. First, we introduce a *sufficiently small* time–delay λ and define the following deviation variables $\dot{\boldsymbol{x}}_0 := \dot{\boldsymbol{x}}(t-\lambda)$, $\boldsymbol{x}_0 := \boldsymbol{x}(t-\lambda)$, and $\boldsymbol{u}_0 := \boldsymbol{u}(t-\lambda)$, which are the λ–time–delayed signals of the current state derivative $\dot{\boldsymbol{x}}(t)$, state $\boldsymbol{x}(t)$, and control $\boldsymbol{u}(t)$, respectively. Moreover, we will denote $\Delta\dot{\boldsymbol{x}} := \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0$, $\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_0$, and $\Delta\boldsymbol{u} := \boldsymbol{u} - \boldsymbol{u}_0$ as the incremental state derivative, the incremental state, and the so–called incremental control input, respectively. To obtain an incremental form of system dynamics, we consider a first-order Taylor series expansion of $\dot{\boldsymbol{x}}(t)$ [71, 80], not in the geometric sense, but with respect to the newly introduced time–delay λ as

$$\begin{aligned}\dot{\boldsymbol{\zeta}} &= \dot{\boldsymbol{\zeta}}_0 + \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}\right]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \Delta\boldsymbol{x} \\ &\quad + \boldsymbol{M}(\boldsymbol{x}_0)\Delta\boldsymbol{u} + \mathcal{O}(\Delta\boldsymbol{x}^2) \\ &\cong \dot{\boldsymbol{\zeta}}_0 + \boldsymbol{L}_0(\boldsymbol{x}_0)\Delta\boldsymbol{x} + \boldsymbol{M}(\boldsymbol{x}_0)\Delta\boldsymbol{u}\end{aligned}$$

with

$$\dot{\boldsymbol{\zeta}}_0 := \dot{\boldsymbol{\zeta}}(t-\lambda) = \boldsymbol{l}(\boldsymbol{x}_0) + \boldsymbol{M}(\boldsymbol{x}_0)\boldsymbol{u}_0 \tag{9.15a}$$

and

$$\boldsymbol{L}_0(\boldsymbol{x}_0) = \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}\right]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \tag{9.16a}$$

which represents the Jacobian linearization of the on-board model. We will refer to $\boldsymbol{M}(\boldsymbol{x}_0)$ as the *instantaneous control effectiveness (ICE)* matrix; meaning that this model–based term is sampled at each incremental instant. This means an approximate linearization about the λ–delayed signals is performed *incrementally*, and not with respect to a particular equilibrium or operational point of interest.

Time-scale separation (TSS) assumption: *For a sufficiently small time-delay λ and for any incremental control input, it is assumed that $\Delta\boldsymbol{x}$ does not vary significantly during λ. In other words, the input rate of change is much faster than the state rate of change:*

$$\epsilon_{INDI_{\text{TSS}}}(t) \equiv \Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_0 \cong 0, \ \forall\ \Delta\boldsymbol{u} \tag{9.17}$$

which leads to

$$\dot{\boldsymbol{\zeta}} \cong \dot{\boldsymbol{\zeta}}_0 + \boldsymbol{L}_0 \underbrace{(\boldsymbol{x} - \boldsymbol{x}_0)}_{\cong 0} + \boldsymbol{M}(\boldsymbol{x}_0) \cdot (\boldsymbol{u} - \boldsymbol{u}_0)$$

or simply

$$\Delta\dot{\boldsymbol{\zeta}} \cong \boldsymbol{M}(\boldsymbol{x}_0) \cdot \Delta\boldsymbol{u} \tag{9.18}$$

This assumption shows that for high sampling rates the nonlinear system dynamics in its incremental form is simply approximated by its ICE matrix $\boldsymbol{M}(\boldsymbol{x}_0)$. Since this results in a change of coordinates, the development of control laws in the original set of (absolute) coordinates implies or requires the availability of $\dot{\boldsymbol{\zeta}}_0$ and $\boldsymbol{u}_0$ in (9.18). For the obtained approximation $\Delta\dot{\boldsymbol{\zeta}} \cong \boldsymbol{M}(\boldsymbol{x}_0) \cdot \Delta\boldsymbol{u}$, NDI is applied to obtain a relation between the incremental control input and the output of the system

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{M}(\boldsymbol{x}_0)^{-1}(\boldsymbol{\nu} - \dot{\boldsymbol{\zeta}}_0). \tag{9.19}$$

Note that the incremental input $\boldsymbol{u}_0$ that corresponds to $\dot{\boldsymbol{\zeta}}_0$ is obtained from the output of the actuators, and it has been assumed that a commanded control is achieved *sufficiently fast* in regards to the actuator dynamics. The total control command along with the obtained linearizing control $\Delta\boldsymbol{u} = \boldsymbol{u}(t-\lambda)$ can be rewritten as

$$\boldsymbol{u}(t) = \boldsymbol{u}(t-\lambda) + \boldsymbol{M}(\boldsymbol{x}_0)^{-1}[\boldsymbol{\nu} - \dot{\boldsymbol{\zeta}}(t-\lambda)]. \tag{9.20}$$

The dependency of the closed-loop system on accurate knowledge of the dynamic model in $\boldsymbol{l}(\boldsymbol{x})$ is largely decreased, improving robustness against model uncertainties contained therein. Therefore, this implicit control law design is more dependent on accurate measurements or accurate estimates of $\dot{\boldsymbol{\zeta}}_0$, the state derivatives, and $\boldsymbol{u}_0$, the incremental control input, respectively.

9.3.3. NDI ATTITUDE CONTROL

Since the output of the system has been selected to be the MRP vector $\boldsymbol{y} = \boldsymbol{\sigma}$ the system has a vector of relative degree $[\rho_1 \quad \rho_2 \quad \rho_3]^\top = [2 \quad 2 \quad 2]^\top$ and total relative degree $\rho = 6$. Since $\rho < n$, there are internal states $\boldsymbol{\eta}$ which can be easily proven to lead to marginally stable zero dynamics. Denoting the differentiated outputs $\boldsymbol{\zeta} = [\sigma_1^{\rho_1-1} \quad \sigma_2^{\rho_2-1} \quad \sigma_3^{\rho_3-1}]^\top = [\dot{\sigma}_1 \quad \dot{\sigma}_2 \quad \dot{\sigma}_3]^\top$, the relation (9.12) is obtained, where $\boldsymbol{l}(\boldsymbol{x}) = \boldsymbol{L}_f^2\boldsymbol{\sigma}$ and $\boldsymbol{M}(\boldsymbol{x}) = \boldsymbol{L}_g\boldsymbol{L}_f^1\boldsymbol{\sigma}$. The NDI control law (10.70) cancels all nonlinearities in closed-loop and the nominal closed-loop system (external states) is obtained as

$$\dot{\boldsymbol{\xi}}^{(6)} = \boldsymbol{A}^{(6\times6)}\boldsymbol{\xi}^{(6)} + \boldsymbol{B}^{(6\times3)}\boldsymbol{\nu}^{(3)} \tag{9.21}$$

$$\boldsymbol{y}^{(3)} = \boldsymbol{C}^{(3\times6)}\boldsymbol{\xi}^{(6)} \tag{9.22}$$

where the upper indices indicate the dimensions of the vectors and matrices and the new state vector $\boldsymbol{\xi}$ is defined in terms of the original state $\boldsymbol{x}$ as $\boldsymbol{\xi} = [\sigma_1 \quad \dot{\sigma}_1 \quad \sigma_2 \quad \dot{\sigma}_2 \quad \sigma_3 \quad \dot{\sigma}_3]^\top$ and $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are in Brunovsky block canonical form [97].

Denoting $\boldsymbol{e} = \boldsymbol{\sigma} - \boldsymbol{\sigma}_{ref}$ (valid for small deviations), this single-integrator form can be rendered exponentially stable with

$$\boldsymbol{\nu} = \ddot{\boldsymbol{y}}_d + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} \tag{9.23}$$

where $\ddot{\boldsymbol{y}}_d$ is the feedforward term for tracking tasks, and $\mathbf{k}_D$ and $\mathbf{k}_P$ being 3×3 constant diagonal matrices whose $i-$th diagonal elements $\mathbf{k}_{D_i}$ and $\mathbf{k}_{P_i}$, respectively, are chosen so that the polynomials $s^2 + \mathbf{k}_{D_i} s + \mathbf{k}_{P_i}$ $i = 1, \ldots, n = 3$ may become Hurtwiz. This results in the exponentially stable and decoupled error dynamics

$$\ddot{\boldsymbol{e}} + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} = \mathbf{0} \tag{9.24}$$

which implies that $\boldsymbol{\sigma}(t) \to \boldsymbol{\sigma}_{ref}(t)$ exponentially.

9.3.4. INDI ATTITUDE CONTROL

Since we will consider the dynamics in its incremental form for the control design

$$\dot{\boldsymbol{\zeta}}(t) - \dot{\boldsymbol{\zeta}}(t-\lambda) \cong \boldsymbol{M}(\boldsymbol{x}_0)\,[\boldsymbol{u}(t) - \boldsymbol{u}(t-\lambda)]\,, \tag{9.25}$$

the incremental nonlinear dynamic inversion results in a control law that is only depending on the uncertainties contained within the ICE matrix

$$\boldsymbol{u}(t) = \boldsymbol{u}(t-\lambda) + \boldsymbol{M}(\boldsymbol{x}_0)^{-1}\Big[\boldsymbol{\nu} - \dot{\boldsymbol{\zeta}}(t-\lambda)\Big], \tag{9.26}$$

however, notice that

$$\boldsymbol{M}(\boldsymbol{x}_0) = \underbrace{\frac{\partial\big[\boldsymbol{L}_f^1 \boldsymbol{h}(\boldsymbol{x}_0)\big]}{\partial \boldsymbol{x}}}_{\text{purely kinematic}} \cdot \underbrace{\boldsymbol{G}}_{\text{purely parametric}}. \tag{9.27}$$

This means that in the particular case of this plant, namely a rigid body spacecraft actuated with a non-redundant set of orthogonal reaction wheels and parameterized by MRPs, the incremental nonlinear dynamic inversion is robust since the control law is only exposed to uncertainty in the parametric matrix $\boldsymbol{G}$ which contains information about inertia values (of the rigid body and of the reaction wheels). The term which is purely kinematic in this control law is fully known and contains no uncertainties other than the ones contained within the measured state $\boldsymbol{x}_0$. To conclude the INDI attitude control design, we have made use of the fact that

$$\dot{\boldsymbol{\zeta}}(t-\lambda) = \dot{\boldsymbol{\zeta}}_0 = \ddot{\boldsymbol{\sigma}}_0 = \frac{1}{4}\left[\, \dot{\boldsymbol{B}}(\boldsymbol{\sigma}_0)\cdot\boldsymbol{\omega}_0 + \boldsymbol{B}(\boldsymbol{\sigma}_0)\cdot\dot{\boldsymbol{\omega}}_0 \,\right] \tag{9.28}$$

where the relationship

$$\dot{\boldsymbol{B}}(\boldsymbol{\sigma})\cdot\boldsymbol{\omega} = \frac{1}{2}\big[\, 2\boldsymbol{\sigma}^\top\boldsymbol{\omega}(1-\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega} - (1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega}^\top\boldsymbol{\omega}\boldsymbol{\sigma} \tag{9.29}$$

$$- 4\boldsymbol{\sigma}^\top\boldsymbol{\omega}S(\boldsymbol{\omega})\boldsymbol{\sigma} + 4(\boldsymbol{\sigma}^\top\boldsymbol{\omega})^2\boldsymbol{\sigma} \,\big] \tag{9.30}$$

is highly beneficial to compute $\ddot{\boldsymbol{\sigma}}_0$ which is otherwise very hard to estimate because of the noise contained in the measurements. By using the measured $\dot{\boldsymbol{\zeta}}(t-\lambda)$ and commanded $\boldsymbol{u}(t-\lambda)$ incrementally, we practically obtain a nonlinear 'self-scheduling' NDI control law that is robust to model and parametric uncertainties. The use of $\boldsymbol{M}(\boldsymbol{x}_0)$ in INDI is one of the key differences with respect to *time-delay control*, where the control effectiveness is substituted with a constant gain matrix instead. This method is briefly presented next.

9.3.5. TIME–DELAY CONTROL AND RELATIONSHIP TO INDI

Consider the following transformation as in [142]

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u}) + \bar{\boldsymbol{M}}\boldsymbol{u} \tag{9.31}$$

with

$$\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{l}(\boldsymbol{x}) + \left[\boldsymbol{M}(\boldsymbol{x}) - \bar{\boldsymbol{M}}\right]\boldsymbol{u}, \tag{9.32}$$

and with $\bar{\boldsymbol{M}}$, an scalar-valued and invertible gain matrix referred to as the *incremental gain effectiveness (IGE)* matrix from now on. Defining the vector $\boldsymbol{\alpha}(\boldsymbol{x})$ and matrix $\boldsymbol{\beta}$ as

$$\boldsymbol{\alpha}(\boldsymbol{x}) = -\bar{\boldsymbol{M}}^{-1}\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u}) \tag{9.33a}$$

$$\boldsymbol{\beta} = \bar{\boldsymbol{M}}^{-1} \tag{9.33b}$$

then, the state feedback control law $\boldsymbol{u}$ defined as

$$\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\beta}\cdot\boldsymbol{\nu} = \bar{\boldsymbol{M}}^{-1}\left[\boldsymbol{\nu} - \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u})\right] \tag{9.34}$$

cancels all nonlinearities in the nominal closed-loop case, as shown before, where we have used the virtual control input as $\boldsymbol{\nu} = \dot{\boldsymbol{\zeta}}_{\text{des}}$. Notice however, that still a full model of $\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u})$ is needed. Because this reformulated NDI control law is nevertheless *still* depending on the model represented by $\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u})$, this controller is again susceptible to uncertainties in this term.

To cope with the uncertainty issue, we will consider an estimation of $\boldsymbol{H}$ denoted by $\bar{\boldsymbol{H}}$ along the lines of *time delay control* (TDC) [142], and therefore we will consider the usual dynamic inversion input transformation of (9.31) but with the $\bar{\boldsymbol{H}}$ estimate instead

$$\boldsymbol{u} = \bar{\boldsymbol{M}}^{-1}\left[\boldsymbol{\nu} - \bar{\boldsymbol{H}}(\boldsymbol{x},\boldsymbol{u})\right] \tag{9.35}$$

being the nominal case when $\bar{\boldsymbol{H}} = \boldsymbol{H}$ which results in perfect inversion. Our remaining task is therefore to find a suitable $\bar{\boldsymbol{H}}$ estimate such that, in combination with $\boldsymbol{\nu}$, the closed-loop system converges exponentially fast to Eq. (9.14) while avoiding the uncertain terms to grow unbounded. This means that, ultimately, the control law given by Eq. (9.35) is able to obtain the desired closed-loop dynamics defined by the nominal single integrator form while rejecting the perturbation due to the uncertainties in $\Delta\boldsymbol{H}$. For the sufficiently small time-delay λ already introduced, we consider the following approximation to hold [142] such that $\boldsymbol{H}$ does not vary significantly during λ

$$\boldsymbol{\epsilon}_{\text{TDE}_{error}}(t) \equiv \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},t) - \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u},t-\lambda) \cong 0 \tag{9.36}$$

which is called *time-delay estimation error* at time t. If we write the following current, and delayed dynamics, respectively

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{H}(\boldsymbol{x},\boldsymbol{u}) + \bar{\boldsymbol{M}}\cdot\boldsymbol{u}, \quad \dot{\boldsymbol{\zeta}}_{\boldsymbol{0}} = \boldsymbol{H}(\boldsymbol{x}_0,\boldsymbol{u}_0) + \bar{\boldsymbol{M}}\cdot\boldsymbol{u}_0$$

it is clear that

$$\boldsymbol{H}(\boldsymbol{x},\boldsymbol{u}) - \boldsymbol{H}(\boldsymbol{x}_0,\boldsymbol{u}_0) = (\dot{\boldsymbol{\zeta}} - \dot{\boldsymbol{\zeta}}_{\boldsymbol{0}}) - \bar{\boldsymbol{M}}\,(\boldsymbol{u}-\boldsymbol{u}_0) \cong 0.$$

or simply

$$\Delta\dot{\boldsymbol{\zeta}} \cong \bar{\boldsymbol{M}}\cdot\Delta\boldsymbol{u}. \tag{9.37}$$

This relationship is used together with Eq. (9.31) to obtain what is called *time-delay estimation* (TDE) as the following

$$\bar{\boldsymbol{H}} = \boldsymbol{H}(t-\lambda) = \dot{\boldsymbol{\zeta}}(t-\lambda) - \bar{\boldsymbol{M}}\cdot\boldsymbol{u}(t-\lambda) \tag{9.38}$$

therefore we can rewrite in our usual notation as

$$\bar{\boldsymbol{H}} = \boldsymbol{H}_0 = \dot{\boldsymbol{\zeta}}_0 - \bar{\boldsymbol{M}}\cdot\boldsymbol{u}_0 \tag{9.39}$$

9.3.6. PARALLELS BETWEEN INDI AND TDC

With the TDE, the incremental counterpart of Eq. (10.55) results in a control law that is not depending on the dynamics model in $\boldsymbol{H}$ which contains $\boldsymbol{l}(\boldsymbol{x})$ and the control effectiveness $\boldsymbol{M}(\boldsymbol{x})$, but instead on the IGE matrix $\bar{\boldsymbol{M}}$ as

$$\boldsymbol{u} = \boldsymbol{u}_0 + \bar{\boldsymbol{M}}^{-1}\left[\boldsymbol{\nu} - \dot{\boldsymbol{\zeta}}_0\right]. \tag{9.40}$$

in other words

$$\boldsymbol{u}(t) = \boldsymbol{u}(t-\lambda) + \bar{\boldsymbol{M}}^{-1}\left[\boldsymbol{\nu} - \dot{\boldsymbol{\zeta}}(t-\lambda)\right]. \tag{9.41}$$

This TDC law can be interpreted as an INDI control whenever

$$\bar{\boldsymbol{M}} = \boldsymbol{M}(\boldsymbol{x}_0), \tag{9.42}$$

however, we had taken from the literature of TDC as the IGE being a time-invariant gain matrix, which is the main distinction with regards to INDI control laws. In that regard, we can conclude that the INDI control laws are combined model– and sensor–based control laws which are promising for high-performance nonlinear and robust attitude control because of this self–scheduling property of the ICE matrix $\boldsymbol{M}(\boldsymbol{x}_0)$. Note that the self-scheduling properties of INDI in Eq. (10.55) due to the ICE term $\boldsymbol{M}(\boldsymbol{x}_0)$ were lost in the TDC law of Eq. (9.40), suggesting that $\bar{\boldsymbol{M}}$ should be an scheduling variable as in INDI by imposing the equivalence $\bar{\boldsymbol{M}} = \boldsymbol{M}(\boldsymbol{x}_0)$.

9.3.7. DISCRETE FORMULATIONS OF INDI, TDC, AND PID CONTROL AND THEIR RELATIONSHIPS

For practical implementations, sampled-time formulations involving continuous and discrete quantities as in [142] are more convenient and restated here. For that, the smallest λ one can consider is the equivalent of the sampling period of the on-board computer. The sampled formulation of (9.40) may be expressed as

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \bar{\boldsymbol{M}}^{-1}\left[\boldsymbol{\nu}(k-1) - \dot{\boldsymbol{\zeta}}(k-1)\right] \tag{9.43}$$

where it has been necessary to consider $\boldsymbol{\nu}$ at sample $k-1$ for causality reasons. Replacing the sampled virtual control $\boldsymbol{\nu}$ accordingly, we have

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \bar{\boldsymbol{M}}^{-1}\left[\dot{\boldsymbol{\zeta}}_d(k-1) + \mathbf{k}_D\dot{\boldsymbol{e}}(k-1) + \mathbf{k}_P\boldsymbol{e}(k-1) - \dot{\boldsymbol{\zeta}}(k-1)\right] \tag{9.44}$$

which results in

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \bar{\boldsymbol{M}}^{-1}\left[\ddot{\boldsymbol{e}}(k-1) + \mathbf{k}_D\dot{\boldsymbol{e}}(k-1) + \mathbf{k}_P\boldsymbol{e}(k-1)\right] \tag{9.45}$$

Previous results reported in the robotics literature by [142] show the relationship between this discrete formulation of TDC and proportional-integral-derivative control (PID). [96] showed that INDI is equivalent to TDC but only under the consideration when the ICE matrix was constant. This in turn suggested a meaningful and systematic method for PI(D)-control tuning of robust nonlinear flight control systems via INDI as originally suggested in the systematic method for gain selection of robust PID controllers for nonlinear plants by [142]. [142] showed this relationship first by considering the discrete implementation of a PID control

$$\boldsymbol{u}(k) = \boldsymbol{K}\left[\boldsymbol{e}(k-1) + \boldsymbol{T}_I^{-1}\sum_{i=0}^{k-1} t_s\boldsymbol{e}(i) + \boldsymbol{T}_D\dot{\boldsymbol{e}}(k-1)\right] + \boldsymbol{u}_B. \tag{9.46}$$

where $\boldsymbol{K}$ denotes a diagonal proportional gain matrix, $\boldsymbol{T}_I$ a constant diagonal matrix representing a reset or integral time, $\boldsymbol{T}_D$ denotes a constant diagonal matrix representing derivative time, and $\boldsymbol{u}_B$ denotes a constant vector representing a trim-bias, from initial conditions. When subtracting two consecutive terms of a discrete formulation, the integral sum can be removed and thus the so-called PID controller in incremental form can be obtained

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \boldsymbol{K}\cdot t_s\cdot\left[\boldsymbol{T}_D\,\ddot{\boldsymbol{e}}(k-1) + \dot{\boldsymbol{e}}(k-1) + \boldsymbol{T}_I^{-1}\cdot\boldsymbol{e}(k-1)\right] \tag{9.47}$$

If we consider a nonlinear-PID control in the form

$$\boldsymbol{u}(k) = \boldsymbol{K}(\boldsymbol{x})\left[\boldsymbol{e}(k-1) + \boldsymbol{T}_I^{-1}\sum_{i=0}^{k-1} t_s\boldsymbol{e}(i) + \boldsymbol{T}_D\dot{\boldsymbol{e}}(k-1)\right], \tag{9.48}$$

comparing terms from Eqs. (9.45)-(9.47)-(9.48), we have the following relationships as originally found by [142] which are the relationship between the discrete formulations of TDC and PID in incremental form

$$\boldsymbol{K}(\boldsymbol{x}) = \bar{\boldsymbol{K}} = \mathbf{k}_D\cdot(\bar{\boldsymbol{M}}\cdot t_s)^{-1}, \tag{9.49a}$$

$$\boldsymbol{T}_I = \mathbf{k}_D\cdot\mathbf{k}_P^{-1}, \tag{9.49b}$$

$$\boldsymbol{T}_D = \mathbf{k}_D^{-1}, \tag{9.49c}$$

Referring back to the Eqs. (9.42)-(9.48) which shows the relationship between INDI and TDC, considering the state-dependent (and therefore scheduled) nonlinear–PID proportional gain matrix $\boldsymbol{K}(\boldsymbol{x})$, it is related to the ICE matrix $\boldsymbol{M}(\boldsymbol{x}_0)$ via the relationship

$$\boldsymbol{K}(\boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x}_0) = \mathbf{k}_D\cdot\left[\boldsymbol{M}(\boldsymbol{x}_0)\cdot t_s\right]^{-1}, \tag{9.50}$$

which then clearly suggests not only that an equivalent discrete and incremental PID controller with gains $< \boldsymbol{K}, \boldsymbol{T}_i, \boldsymbol{T}_d >$ can be obtained in relationship to TDC but also in relationship to INDI when considering an incremental and self-scheduled nonlinear-PID controller with gains $< \boldsymbol{K}(\boldsymbol{x}_0), \boldsymbol{T}_i, \boldsymbol{T}_d >$. Moreover, the tuning of these (nonlinear-)PIDs proves to be more meaningful and systematic than heuristic methods as already pointed out in [96, 142]. This is because the design starts from prescribing desired error dynamics $\ddot{\boldsymbol{e}} + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} = \mathbf{0}$ by tuning the Hurwitz gains $< \mathbf{k}_P, \mathbf{k}_D, >$ and what follows is finding the remaining IGE matrix $\bar{\boldsymbol{M}}$ by the TDC approach, or with the ICE matrix $\boldsymbol{M}(\boldsymbol{x}_0)$ with the INDI approach. In essence, this procedure is more efficient and much less cumbersome than designing a whole set of PID gains iteratively. Moreover, for attitude control systems, the self-scheduling properties of inversion-based controllers have suggested superior advantages with respect to PID controls since these are, in general, not gain-scheduled according to the nonlinear motion of the plant [85]. The relationships here outlined suggests that scheduling of incremental PID control shall be done at the level of the proportional gain $\boldsymbol{K}(\boldsymbol{x})$ via the IGE matrix $\bar{\boldsymbol{M}}$ or ICE matrix $\boldsymbol{M}(\boldsymbol{x}_0$, and *not* over the whole set of gains $< \boldsymbol{K}(\boldsymbol{x}), \boldsymbol{T}_i, \boldsymbol{T}_d >$.

9.3.8. STABILITY AND ROBUSTNESS ANALYSIS

INDI relies on the assumption that for small time increments and high sampling rates, the nonlinear system dynamics in its incremental form is simply approximated by the (linearized) control effectiveness evaluated at the current state. However, and owing to the finite time delay one can achieve in digital devices, there exists an error $\boldsymbol{\epsilon}(t)$ [97], called the TDE error in the TDC literature [142], for which the error dynamics can be regarded as

$$\ddot{\boldsymbol{e}} + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} = \boldsymbol{\epsilon}(t). \tag{9.51}$$

Previous theoretical stability and robustness proofs for INDI controllers had the problem of not having considered this important residual error as pointed out by [97]. Recently, the INDI control in the literature has been reformulated for systems with arbitrary relative degree and without recurring to cascaded-control structures, i. e., without using a time–scale separation assumption [97]. This reformulation allowed to extended further the incremental nonlinear control approach for Sliding Mode Control [99]. For these new reformulations and extensions, conditions for stability and robustness analyses of incremental nonlinear control have been finally established and analyzed using Lyapunov-based methods. Details on the sufficient conditions for closed–loop stability under INDI and discrete TDC, and therefore applicable to this problem can be found in [97, 99, 141, 142].

The existing sufficient condition for closed-loop stability of INDI [99] for input–output linearizable plants have been proposed as follows, which is similar to the one proposed for TDC [140–142], and under the condition that zero dynamics of the plant is exponentially stable and the desired trajectory and its derivatives are bounded

$$\left\| \boldsymbol{I}_n - \boldsymbol{M}(\boldsymbol{x}) \cdot \bar{\boldsymbol{M}}^{-1} \right\| \leq \bar{b} < 1 \tag{9.52}$$

However, this condition does not have the sampling time explicitly considered and it has been found that even with a very small sampling time this condition might be vio-

lated [141]. A sufficient condition for closed-loop stability for discrete TDC systems is presented by [141, 142] as the following (taking λ as the sampling):

$$\left\| \boldsymbol{I}_n - \boldsymbol{M}(\boldsymbol{x}) \cdot \bar{\boldsymbol{M}}^{-1} \right\| < \frac{1}{1 + \left[(1 + \beta_1 \gamma_P)\gamma_D + \beta_2 \gamma_{PD} \right] \lambda} \tag{9.53}$$

where β_1, β_2, γ_D, γ_P, and γ_{PD} are tunable gains. To conclude, the influence model uncertainties to the reformulated system can be regarded as

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}) + \Delta \boldsymbol{H}(\boldsymbol{x}, \lambda) + \bar{\boldsymbol{M}} \cdot \boldsymbol{u} \tag{9.54a}$$

and therefore, application of the control law $\boldsymbol{u} = \bar{\boldsymbol{M}}^{-1} \left[\boldsymbol{\nu} - \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}) \right]$ to this uncertain dynamics actually gives $\dot{\boldsymbol{\zeta}} = \boldsymbol{\nu} + \Delta \boldsymbol{H}(\boldsymbol{x}, \lambda)$ which is not linearizing as expected because of the extra uncertain term. This major flaw of NDI-based control systems is well known and also previously demonstrated by [71] among others. [97, 99] proved that

$$\lim_{\lambda \to 0} \| \Delta \boldsymbol{H}(\boldsymbol{x}, \lambda) \| = \boldsymbol{0}, \quad \forall \boldsymbol{x} \in \mathscr{R}^n \tag{9.55}$$

which implies that the term $\Delta \boldsymbol{H}$ becomes negligible for sufficiently high sampling rates, which has been the common assumption behind INDI control laws, and furthermore, asymptotic stability of the nominal system is proven as the closed-loop system can be ultimately bounded by a class $\mathscr{K}$ function of the perturbation bounds.

9.4. ATTITUDE CONTROL SIMULATIONS

For numerical simulations to demonstrate the high-agility attitude control system, we use the comprehensive analytical nonlinear model of Section I for a small satellite with an inertia matrix of

$$\boldsymbol{I} = \begin{bmatrix} 10 & 1 & 0.5 \\ 1 & 7 & 0.2 \\ 0.5 & 0.2 & 9 \end{bmatrix} \text{Kg} \cdot \text{m}^2,$$

and as main torque actuators, an array of three *'High-Torque-Wheels'* (*HTW*) in orthogonal configuration (and aligned with the principal axes). Wheel characteristics for these *HTW*s are presented in ([193]), where the most important ones are their max. torque of 0.23 [Nm] and moment of inertia of 5×10^{-3} [Kg$\cdot$m^2].

The initial *HTW* wheel speeds are zero; normally during operation, initial wheel speeds represent the angular momentum stored in the satellite. The MRP tracking reference commands are designed smooth up to a second order with a simple reference trajectory generator. The second derivative of these reference commands will act as feedforward acceleration commands. We restrict these maneuvers according to the actuator limits in order to avoid the case of actuator saturation. For all simulations we consider the virtual controller $\boldsymbol{\nu} = \ddot{\boldsymbol{y}}_d + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e}$ so that the error dynamics are equivalent across different scenarios. This is a classical second order dynamics where considering a natural frequency $\omega_n = 3$ rad/s and damping coefficient of $\zeta = 0.707$ we can obtain the gains $\mathbf{k}_{D_i} = 2 \cdot \zeta \cdot \omega_n = 4.242$ and $\mathbf{k}_{P_i} = \omega_n^2 = 9$, $i = 1, 2, 3$.

Simulation results in nominal condition verifies that INDI and TDC/PID control perform quite similarly. To study the performance under realistic conditions, we apply uncertainty in the inertia matrix of the satellite platform and perform Monte-Carlo simulations. Figure 1 presents the performance of the INDI attitude control under the uncertainty considered by showcasing the attitude tracking for the MRP reference maneuver commanded and the respective tracking error. Further simulations showcase a similar performance of the TDC/PID attitude control under the same uncertainty in the inertia parameters.

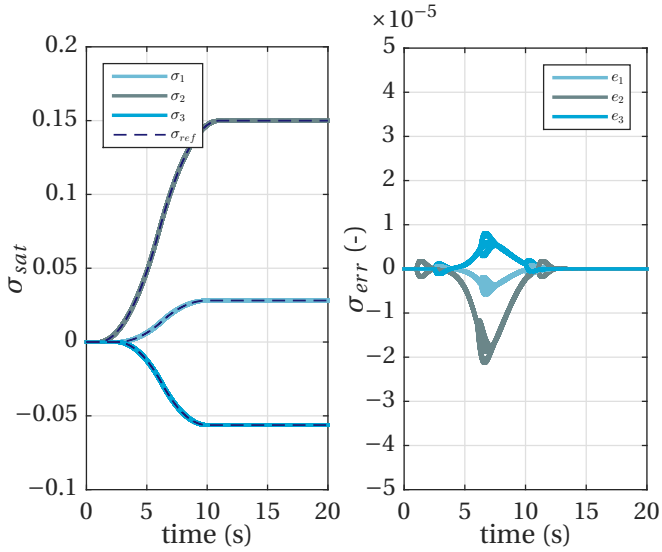


Figure 9.1: INDI control: MRP attitude tracking and tracking error during a fast slew maneuver under uncertainty.

However, the nonlinear control laws perform differently in terms of robust performance and stability according to the metric in Eq. (9.52). This result is shown in Figure 2 for both INDI and TDC/PID. At this stage it becomes evident that the self–scheduling property of the INDI controller as compared to the TDC/PID controllers makes the attitude control system to guarantee a better stability margin as compared to TDC/PID; in the latter case, their static control effectiveness hinders the stability margin as it is proportional to both the maneuver and the size of the uncertainty. In summary, simulation results verified similar nonlinear performance of agile attitude control using both INDI and TDC/PID control. The robustness and stability properties have been shown to be superior for INDI in comparison to TDC/PID control for this particular case.

9.5. CONCLUSIONS

In this chapter an agile and robust nonlinear spacecraft attitude controller is developed based on the recent incremental nonlinear dynamic inversion (INDI) reformulation. This controller is an improvement over the previously INDI approach for spacecraft attitude control in that it considers a non-cascaded dynamic inversion control where scheduling

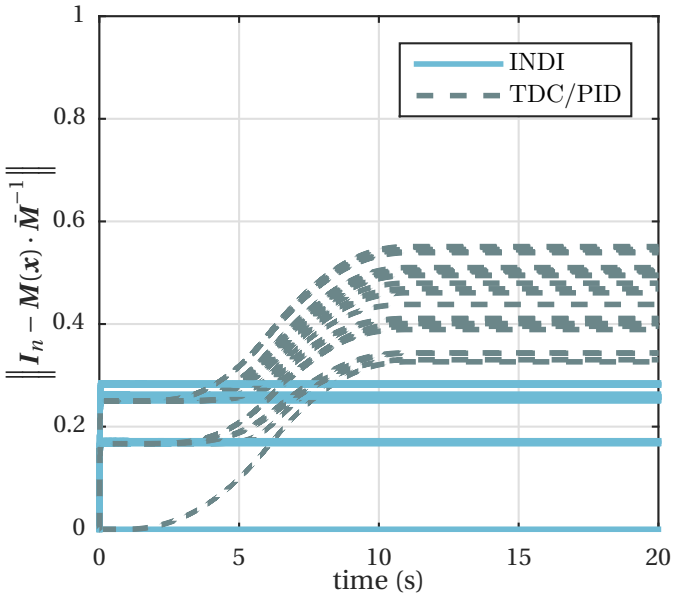


Figure 9.2: INDI and TDC/PID criterion for closed-loop stability under uncertainty.

of the time-varying control effectiveness is done analytically. This results in a nonlinear controller scheduled only by kinematic (fully known) and parametric terms, making it robust to model uncertainties. Finally, a relationship between INDI, time-delay control, and nonlinear-PID control is established. The systematic gain tuning and self scheduling property of our INDI controller can be scaled and readily applied to attitude control of rigid spacecraft for agile maneuvers that do not saturate the actuators; this issue will be addressed in future research. Simulations results shows the effectiveness of our method.

10

A Sampled-Data Form of Incremental Nonlinear Dynamic Inversion for Spacecraft Attitude Control

Abstract

This paper presents a sampled–data form of the recently reformulated incremental nonlinear dynamic inversion (INDI) applied for robust spacecraft attitude control. INDI is a combined model– and sensor–based approach mostly applied for attitude control that only requires an accurate control effectiveness model and measurements of the state and some of its derivatives. This results in a reduced dependency on exact knowledge of system dynamics which is known as a major disadvantage of model–based nonlinear dynamic inversion controllers. However, most of the INDI derivations proposed in the literature assume a very high sampling rate of the system and its controller while also not explicitly considering the available sampling time of the digital control computer. Neglecting the sampling time and its effect in the controller derivations can lead to stability and performance issues of the resulting closed–loop nonlinear system. Therefore, our objective is to bridge this gap between continuous–time and highly sampled INDI formulations and their discrete and lowly sampled counterparts in the context of spacecraft attitude control where low sampling rates are common. Our sampled–data reformulation allows explicit consideration of the sampling time via an approximate sampled–data model in normal form widely known in the literature. The resulting sampled–data INDI control is still robust up to a certain sampling time since it remains only sensitive to parametric uncertainties. Simulation experiments for this particular problem demonstrate this bridge between INDI formulations in continuous and discrete (sampled) time which allows for low sampling control rates.

Publication

Paul Acquatella B., Erik-Jan van Kampen, Qi Ping Chu: *A Sampled-Data Form of Incremental Nonlinear Dynamic Inversion for Spacecraft Attitude Control, to be submitted*.

10.1. INTRODUCTION

FUTURE small satellite systems are expected to be more performant not only for fine pointing capabilities in data acquisition but also in terms of high agility for maneuverability, e. g., for high dynamic slewing capability to command the platform for fast and flexible data acquisition [25–33]. This emerging field of '*agile Earth Observation*' motivated the development of a high-agility attitude control system [193] for the the satellite platform BIROS (*Bispectral InfraRed Optical System*) [27] while actuated with a redundant array of three orthogonal *'High-Torque-Wheels' (HTW)* [25, 26]. However, for agile reorientation, a challenge arises from the fact that time-optimal slew maneuvers are, in general, not of the *Euler-axis* rotation [136, 137] type; especially whenever the actuators are constrained independently [131]. The topic of optimal spacecraft rotational maneuvers is quite extensive and has been studied for many decades [127–133]. Some of the agile attitude control solutions have been experimentally validated for imaging satellites in-orbit [138]. However, most of the work reported in literature relies on optimization and some form of path planning and trajectory optimization, which might be difficult to implement on-board and in practice. In this chapter, we are motivated to find an agile attitude control solution in closed-loop feedback form. This is challenging because of the many nonlinearities involved in reorientation of small satellites as shown in [138, 193] which calls for a robust nonlinear control approach.

Several nonlinear control methodologies have arisen in the past decades; in part to overcome shortcomings of conventional linear techniques, but also to overcome model or parametric uncertainties that can damage the closed-loop stability and convergence of the system. Among the most popular of these control methodologies are *feedback linearization* (FBL), treated extensively in [61–63] and initially based on early papers of Krener and Brockett in the seventies [185, 186], *adaptive control* [64], and *backstepping* [65, 66]. *Nonlinear dynamic inversion* (NDI), which is how feedback linearization is more commonly known in the aerospace literature [60, 67–70], uses an accurate model of the system to entirely or partly cancel its nonlinearities by means of feedback and exact state transformations. This transforms the nonlinear system dynamics into a linear form over a desired region of interest. For the obtained system, conventional linear control techniques can be applied successfully for achieving desired closed-loop dynamics, hence eliminating the need of linearizing and designing different (linear, robust) controllers for several operational points as in gain-scheduling. The motivations behind the application of NDI for flight control systems originate from difficulties with ensuring stability and performance in between operational points of widely-used gain-scheduled controllers. More advanced methods involving robustness and improvements of the method in NDI-based flight control applications have been considered, among many others, in [58, 59, 67, 69–75]. Although initially intended for flight control, NDI for aerospace applications have also found its way for spacecraft control and re-entry vehicles [49–51]. The main disadvantage for the construction of these nonlinear control laws is that accurate knowledge of the nonlinear system dynamics is required for an explicit cancellation (NDI). For this reason, NDI is considered an *explicit* control method where the desired dynamics of the closed-loop system reside in some explicit model to be followed. Therefore, this explicit aspect of NDI-based control laws is considered to be a disadvantage despite its abilities to linearize and decouple certain classes of nonlinear MIMO

systems when full knowledge of the nonlinearities is available. Moreover, this model-based aspect is also strongly influenced by modeling uncertainties. In reality, the model mismatch in the implementation of NDI control laws, together with all sensor aspects, delays and biases, can compromise the performance and stability of the controlled system. Many successful attempts have been carried to identify and reduce these aforementioned flaws of NDI-based control laws with regards to robustness. These attempts are focused in improving the robustness of the overall control architecture by means of applying linear robust control in the outer loop of the system. The works [51, 69] combine NDI with the structured singular value (μ-analysis) and $\mathscr{H}_\infty$ synthesis for reentry flight clearance, and significant benefits were found over conventional NDI. However, not all uncertainties were taken into account or they were covered by lumped uncertainties hence introducing conservatism.

Incremental nonlinear dynamic inversion (INDI) has been proposed as a promising sensor-based approach providing high performance and robust nonlinear control for aerospace vehicles without requiring a detailed model of the controlled plant. The INDI approach reduces its dependency on on-board or baseline models while making use of actuator output and angular acceleration measurement feedback. In contrast to regular NDI, this method is inherently *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed [73, 74]. Theoretical development of increments of nonlinear control action date back from the late nineties and started with activities concerning 'implicit dynamic inversion' for inversion-based flight control in the works of P. R. Smith, B. J. Bacon *et al.* [70, 73], where the architectures considered in this chapter were firstly described. Other designations for these developments found in the literature are 'modified NDI' and 'simplified NDI', but the designation 'incremental NDI' is considered to describe the methodology and nature of these type of control laws better [71, 77–80]. INDI has been elaborated and applied theoretically in the past decade for advanced flight control applications [70, 71, 73–75, 80] as well as in space applications for spacecraft attitude control [4]. More recently, this technique has been applied also in practice for quadrotors using adaptive control [84, 85], and in real flight tests [88–90], verifying its performance and robustness properties against aerodynamic model uncertainties [71, 91, 92] and disturbance rejection [84, 85, 91, 92].

INDI relies on the assumption that for small time increments and high sampling rates, the nonlinear system dynamics in its incremental form is simply approximated by the (linearized) control effectiveness evaluated at the current state. However, previous theoretical stability and robustness proofs for INDI controllers have many drawbacks and were not mathematically consistent as pointed out in [97, 98]. Most of the previous attempts to prove stability were only based on simplifying assumptions and approximated transfer functions and block diagrams [71, 84, 85]. Recently, the INDI control in the literature has been reformulated for systems with arbitrary relative degree and without recurring to cascaded-control structures, i. e., without using a time–scale separation assumption [97, 98]. This reformulation allowed to extended further the incremental nonlinear control approach for Sliding Mode Control [99]. For these new reformulations and extensions, conditions for stability and robustness analyses have been established and analyzed using Lyapunov-based methods. Another nonlinear control

method is *time–delay–control* (TDC) [140–142], more commonly known in the motion control and robotics community and pioneered in the 90's by the works of Hsia, and Youcef-Toumi, *et al.* [140]. TDC works by estimating and compensating disturbances and system uncertainties (model and parametric) by utilizing time–delayed signals of some of the system variables. In [142] it has been shown that TDC can be rendered equivalent to a PID–control under some assumptions and some discrete sampling considerations. This motivates the question to study how neglecting the sampling time and its effect in the controller derivations can lead to stability and performance issues of the resulting closed–loop nonlinear system [195, 196].

In this chapter, we present three main contributions in the context of nonlinear spacecraft attitude control system design. 1) We revisit the NDI and the reformulated INDI for the spacecraft attitude control problem and introduce a time–delay explicitly in this incremental reformulation of INDI. 2) We consider the reformulated INDI control for the spacecraft attitude control problem and introduce its sampled–data form based on the model from [196, 197] where the nonlinear dynamics are approximated by a discrete model with piece–wise constant inputs. 3) We bridge the gap between continuous–time and highly sampled (100 – 1000 Hz) INDI formulations and their discrete and lowly sampled counterparts in the context of spacecraft attitude control where low sampling rates are common (1 – 10 Hz). In that sense, our sampled–data reformulation allows explicit consideration of the sampling time via an approximate sampled–data model in normal form widely known in the literature.

The outline of this chapter is as follows. A nonlinear model of rigid spacecraft equipped with reaction wheels is presented in Sec. II. Section III presents incremental nonlinear dynamic inversion in continuous–time form while Section IV presents incremental nonlinear dynamic inversion in sampled–data form, both for the particular problem of spacecraft attitude control. Attitude control numerical simulations are presented in Sec. V. Conclusions are finally presented in Sec. VI.

10.2. SPACECRAFT MODEL

In this section we describe the nonlinear rotational dynamics model for spacecraft including a generic set of reaction wheels in arbitrary configuration which are driven by exogenous inputs provided by each wheel's powertrain [138, 193]. In this chapter we make use of the Modified Rodrigues Parameters (MRPs) [188, 189, 194] as they represent a well defined attitude parameterization for all Eigen-axis rotations in the large domain of $0^\circ \leq \theta < 360^\circ$, where θ is the principle angle rotation around the Euler-axis λ. Typically, Euler angles and quaternions are used to parameterize the attitude kinematics of rigid bodies and most attitude controllers are based on these parameterizations. The preference of using MRPs is motivated to address the problem of agile reorientation while using a minimal set of three rigid body attitude coordinates, thus avoiding redundancy of parameters (quaternions) or singularities (Euler angles). The MRP attitude kinematics parameterization and their potential advantages have been shown to be suitable for attitude stabilization and control applications [188, 189, 194].

10.2.1. KINEMATICS

Consider first an array consisting of n reaction wheels. Introducing unit vectors $\mathbf{a}_i$ which give the orientation of the spin-axis of each reaction wheel with respect to the spacecraft coordinate system collected in the configuration or alignment matrix:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}, \tag{10.1}$$

then each $\mathbf{a}_i$ can define the $i-$th reaction wheel or 'actuator' frame by taking $\mathbf{a}_i$ as the first axis and making the remaining axes constitute an orthogonal frame. In that sense, the kinematics of the $i-$th reaction wheel with respect to its corresponding actuator frame, in terms of its spin-axis angle Φ_w and angular velocity Ω_w, is simply given by:

$$\dot{\Phi}_{w,i} = \Omega_{w,i} \qquad i = 1, \ldots, n. \tag{10.2}$$

Consider now the spacecraft equipped with the n reaction wheels just introduced. The MRP vector $\boldsymbol{\sigma}$, derived from the quaternion through stereo-graphic projection, is defined in relation to the Euler-axis λ and to the principle angle rotation θ through [189, 194]:

$$\boldsymbol{\sigma} = \lambda \tan \frac{\theta}{4} \tag{10.3}$$

The MRP kinematic differential equation relating the spacecraft MRP vector $\boldsymbol{\sigma}$ (with respect to the inertial frame) with the spacecraft angular velocity $\boldsymbol{\omega} \in \mathscr{R}^3$ (with respect to to the body fixed frame) in vector form is given by [189, 194] as:

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[(1 - \boldsymbol{\sigma}^\top \boldsymbol{\sigma})\boldsymbol{I}_{3\times 3} + 2\boldsymbol{S}(\boldsymbol{\sigma}) + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top\right]\boldsymbol{\omega} = \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\,\boldsymbol{\omega} \tag{10.4}$$

The *skew map* $\boldsymbol{S}(\cdot) : \mathscr{R}^3 \mapsto \mathfrak{so}(3)$ is a linear isomorphism between $\mathscr{R}^3$ and the Lie algebra $\mathfrak{so}(3)$ of 3×3 skew-symmetric matrices and is defined such that $\boldsymbol{S}(\boldsymbol{x})\ \boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{R}^3$, or simply as:

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \quad \boldsymbol{x} \in \mathscr{R}^3. \tag{10.5}$$

Moreover, in this chapter we will also be interested in the fact that the time derivative of the MRP kinematic differential equation produces the exact relation [194]:

$$\ddot{\boldsymbol{\sigma}} = \frac{1}{4}\left[\,\dot{\boldsymbol{B}}(\boldsymbol{\sigma})\cdot\boldsymbol{\omega} + \boldsymbol{B}(\boldsymbol{\sigma})\cdot\dot{\boldsymbol{\omega}}\,\right] = \frac{1}{4}\boldsymbol{C}(\boldsymbol{\sigma},\boldsymbol{\omega},\dot{\boldsymbol{\omega}}) \tag{10.6}$$

where:

$$\dot{\boldsymbol{B}}(\boldsymbol{\sigma})\cdot\boldsymbol{\omega} = \frac{1}{2}\left[\,2\boldsymbol{\sigma}^\top\boldsymbol{\omega}(1-\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega} - (1+\boldsymbol{\sigma}^\top\boldsymbol{\sigma})\boldsymbol{\omega}^\top\boldsymbol{\omega}\boldsymbol{\sigma} - 4\boldsymbol{\sigma}^\top\boldsymbol{\omega}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{\sigma} + 4(\boldsymbol{\sigma}^\top\boldsymbol{\omega})^2\boldsymbol{\sigma}\,\right]$$

which relates the MRP "acceleration" $\ddot{\boldsymbol{\sigma}}$ to the rigid body's angular velocity $\boldsymbol{\omega}$ and angular acceleration $\dot{\boldsymbol{\omega}}$. This relationship will be key for the attitude control design as it will be shown later on.

10.2.2. DYNAMICS

Following the derivations in Karpenko *et al.* [138], we obtain the rotational dynamics model as follows. First, consider the angular momentum of the spacecraft equipped with the reaction wheel array in question:

$$\boldsymbol{H} = \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{h} \tag{10.7}$$

where, expressed in body-fixed frame, $\boldsymbol{H} \in \mathscr{R}^3$ is the total angular momentum of the system, $\boldsymbol{I} \in \mathscr{R}^{3\times 3}$ is the constant inertia matrix of the spacecraft when the reaction wheels are rotating freely, $\boldsymbol{\omega} \in \mathscr{R}^3$ is the spacecraft angular velocity, and $\boldsymbol{h} \in \mathscr{R}^3$ is the total angular momentum vector associated with the reaction wheel array. The angular momentum $\boldsymbol{h}$ can be expressed from individual actuator frames to body-fixed frame as:

$$\boldsymbol{h} = \sum_{i=1}^{n} \mathbf{a}_i h_{w,i} = \boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}, \tag{10.8}$$

where $\boldsymbol{I}_{\boldsymbol{w}}$ is a diagonal matrix of reaction wheel spin-axis inertia values:

$$\boldsymbol{I}_{\boldsymbol{w}} = \begin{bmatrix} I_{w,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & I_{w,n} \end{bmatrix}, \tag{10.9}$$

and $\boldsymbol{\Omega}$ the inertial angular rate of the reaction wheel array. Defining $\boldsymbol{\Omega}_w$ as the angular rate of the reaction wheel relative to the actuator frame [138], we have:

$$\boldsymbol{\Omega} = \boldsymbol{\Omega}_w + \boldsymbol{A}^\top \boldsymbol{\omega}. \tag{10.10}$$

where the term $\boldsymbol{A}^\top\boldsymbol{\omega}$ is the extra angular velocity of the reaction wheels due to rotation of the spacecraft. Considering the angular momentum associated with the $i-$th reaction wheel in actuator frame:

$$h_{w,i} = I_{w,i}\,(\Omega_{w,i} + \mathbf{a}_i^\top \boldsymbol{\omega}), \qquad i = 1,\ldots,n, \tag{10.11}$$

we can already obtain the differential equation describing the reaction wheel dynamics in terms of reaction wheel torques $\tau_{w,i}$, which are considered as the exogenous inputs to the system provided by the wheel's powertrain:

$$\dot{\Omega}_{w,i} = I_{w,i}^{-1}\tau_{w,i} - \mathbf{a}_i^\top \dot{\boldsymbol{\omega}}, \qquad i = 1,\ldots,n. \tag{10.12}$$

Because the angular momentum must be conserved in the absence of external perturbations, applying the transport theorem [127, 138] to Eq. (10.7), the following relation is obtained:

$$\frac{\mathrm{d}\boldsymbol{H}}{\mathrm{d}t}\bigg|_{\mathscr{I}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H} + \boldsymbol{\omega}^{\mathscr{B}/\mathscr{I}} \times \boldsymbol{H} = \mathbf{0}, \tag{10.13}$$

where $\mathscr{I}$ and $\mathscr{B}$ denotes the inertial and body frame, respectively, and we had denoted $\boldsymbol{\omega} \equiv \boldsymbol{\omega}^{\mathscr{B}/\mathscr{I}}$ overall in the chapter for notation convenience. Eq. (10.13) can be further expanded as:

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\dot{\boldsymbol{\Omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}) = \mathbf{0}. \tag{10.14}$$

Combining Eqs. (10.8), (10.12), and (10.14), the comprehensive nonlinear model for spacecraft dynamics equipped with reaction wheels [138] is given by:

$$\boldsymbol{\Gamma}\begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\Omega}_{w,1} \\ \vdots \\ \dot{\Omega}_{w,n} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}\times\left(\boldsymbol{I}\boldsymbol{\omega}+\boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}_w+\boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{A}^\top\boldsymbol{\omega}\right) \\ \tau_{w,1} \\ \vdots \\ \tau_{w,n} \end{bmatrix} \tag{10.15}$$

where:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{I}+\boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{A}^\top & \mathbf{a}_1 I_{w,1} & \cdots & \mathbf{a}_n I_{w,n} \\ I_{w,1}\mathbf{a}_1^\top & I_{w,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_{w,n}\mathbf{a}_n^\top & 0 & \cdots & I_{w,n} \end{bmatrix}$$

is an augmented inertia coupling matrix for the full system.

10.2.3. FULL NONLINEAR SPACECRAFT MODEL

The augmentation of the nonlinear spacecraft dynamics model together with the MRP kinematics can be rewritten as a full model in the generic form of affine n-dimensional multivariable nonlinear system with m inputs u_i and p outputs y_i as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{10.16a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{10.16b}$$

where $\boldsymbol{x}\in\mathscr{R}^n$, $\boldsymbol{u}\in\mathscr{R}^m$, and $\boldsymbol{y}\in\mathscr{R}^p$. The functions $\boldsymbol{f}(\boldsymbol{x}) = [\,f_1(\boldsymbol{x}) \quad \cdots \quad f_n(\boldsymbol{x})\,]^\top$, $\boldsymbol{g}(\boldsymbol{x}) = [\,\boldsymbol{g}_1(\boldsymbol{x}) \quad \ldots \quad \boldsymbol{g}_m(\boldsymbol{x})\,]^\top \in \mathscr{R}^{n\times m}$, and $\boldsymbol{h}(\boldsymbol{x}) = [\,h_1(\boldsymbol{x}) \quad \cdots \quad h_p(\boldsymbol{x})\,]^\top$ are assumed as smooth vector fields continuously differentiable on $\mathscr{R}^n$. In this chapter we consider the output MRP as control variables $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\sigma}$ and assume to have three reaction wheels ($n_w = 3$) as actuators, hence $\boldsymbol{u} = \boldsymbol{\tau}_w = [\,\tau_{w,1} \quad \tau_{w,2} \quad \tau_{w,3}\,]^\top$, and $p = m = 3$. In the usual where $p < m$, meaning that there are more control inputs than control variables, the inversion required for input–output linearization is not direct and some form of control allocation is required. Else, when $p > m$, the system is said to be underactuated and therefore the input–output linearization is underdetermined and possibly not feasible. These two cases are however out of the scope of this chapter since $p = m$. Considering the vector $\boldsymbol{x} = [\,\boldsymbol{\sigma}^\top \quad \boldsymbol{\omega}^\top \quad \boldsymbol{\Omega}^\top\,]^\top$ with $\boldsymbol{\sigma} = [\,\sigma_1 \quad \sigma_2 \quad \sigma_3\,]^\top$, $\boldsymbol{\omega} = [\,\omega_x \quad \omega_y \quad \omega_z\,]^\top$, and $\boldsymbol{\Omega}_w = [\,\Omega_{w,1} \quad \Omega_{w,2} \quad \Omega_{w,3}\,]^\top$, the full nonlinear system dynamics in Eqs. (10.16a)–(10.16b) results in $n = 9$ and is given by:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\,\boldsymbol{\omega} \\ \boldsymbol{\Gamma}^{-1}\begin{bmatrix} -\boldsymbol{\omega}\times\left(\boldsymbol{I}\boldsymbol{\omega}+\boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{\Omega}_w+\boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{A}^\top\boldsymbol{\omega}\right) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \end{bmatrix}, \tag{10.17a}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \boldsymbol{\Gamma}^{-1}\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{1}_{3\times 3} \end{bmatrix} \end{bmatrix} = \boldsymbol{G}. \tag{10.17b}$$

10.3. INCREMENTAL NONLINEAR DYNAMIC INVERSION IN CONTINUOUS–TIME FORM

In this section we revisit incremental nonlinear dynamic inversion in the context of input–output feedback linearization described in (companion) normal form, and we apply this transformation to the attitude control problem. Moreover, we consider the continuous–time descriptions already widely described in the literature [4, 71, 80, 97, 98].

10.3.1. NONLINEAR DYNAMIC INVERSION PRELIMINARIES

Finding an explicit relationship between the input u and the output y is generally not straightforward because they are not directly related. Consider again the generic model of affine n-dimensional multivariable nonlinear systems described in Eqs. (10.16a)–(10.16b). Collecting all differentiated outputs y_i results in m equations in the form of:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}, \tag{10.18}$$

where:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \begin{bmatrix} y_1^{(\rho_1)} \\ y_2^{(\rho_2)} \\ \vdots \\ y_m^{(\rho_m)} \end{bmatrix}, \qquad \boldsymbol{l}(\boldsymbol{x}) = \begin{bmatrix} \mathscr{L}_f^{\rho_1} h_1(\boldsymbol{x}) \\ \mathscr{L}_f^{\rho_2} h_2(\boldsymbol{x}) \\ \vdots \\ \mathscr{L}_f^{\rho_m} h_m(\boldsymbol{x}) \end{bmatrix}, \tag{10.19a}$$

$$\boldsymbol{M}(\boldsymbol{x}) = \begin{bmatrix} \mathscr{L}_{g_1}\mathscr{L}_f^{\rho_1-1} h_1(\boldsymbol{x}) & \mathscr{L}_{g_2}\mathscr{L}_f^{\rho_1-1} h_1(\boldsymbol{x}) & \cdots & \mathscr{L}_{g_m}\mathscr{L}_f^{\rho_1-1} h_1(\boldsymbol{x}) \\ \mathscr{L}_{g_1}\mathscr{L}_f^{\rho_2-1} h_2(\boldsymbol{x}) & \mathscr{L}_{g_2}\mathscr{L}_f^{\rho_2-1} h_2(\boldsymbol{x}) & \cdots & \mathscr{L}_{g_m}\mathscr{L}_f^{\rho_2-1} h_2(\boldsymbol{x}) \\ \vdots & \vdots & & \vdots \\ \mathscr{L}_{g_1}\mathscr{L}_f^{\rho_m-1} h_m(\boldsymbol{x}) & \mathscr{L}_{g_2}\mathscr{L}_f^{\rho_m-1} h_m(\boldsymbol{x}) & \cdots & \mathscr{L}_{g_m}\mathscr{L}_f^{\rho_m-1} h_m(\boldsymbol{x}) \end{bmatrix}, \tag{10.19b}$$

and $\mathscr{L}_f^{\rho_j} h_j$, $\mathscr{L}_{g_i}\mathscr{L}_f^{\rho_j-1} h_j$ are the Lie derivatives [61, 76] of the scalar functions h_j with respect to the vector fields $\boldsymbol{f}$ and $\boldsymbol{g}_i$, where $j,\ i = 1,2,\ldots,m$, respectively. Moreover, the system is said to have a vector of relative degree $\boldsymbol{\rho} = [\,\rho_1 \quad \ldots \quad \rho_p\,]^\top$ at some point $\bar{\boldsymbol{x}} \in \mathscr{R}^n$ of the state–space when there exists a region of interest $\mathscr{D}_0 \subset \mathscr{R}^n$ around $\bar{\boldsymbol{x}}$ such that for all $\boldsymbol{x} \in \mathscr{D}_0$, $\boldsymbol{M}(\boldsymbol{x})$ is nonsingular (i.e., invertible) and:

$$\mathscr{L}_{g_i}\mathscr{L}_f^k h_j(\boldsymbol{x}) = 0, \quad 0 \leq k \leq \rho_j - 1, \quad 1 \leq i,\ j \leq m.$$

In other words, the vector of relative degree represents, for each output $y_i,\ i = 1,\ldots,p$, the number of output differentiations needed for the input to appear [61, 76]. Moreover, the vector of relative degree of the continuous–time nonlinear system satisfies:

$$\rho = \|\boldsymbol{\rho}\|_1 = \sum_{i=1}^{m} \rho_i \leq n \tag{10.20}$$

where ρ is henceforth called the total relative degree of the system. Furthermore, the system is said to have uniform relative degree when $\rho_1 = \rho_2 = \cdots = \rho_m$. Denoting the m

outputs y_j and their derivatives up to the $(\rho_j - 1)$ order as new states $\boldsymbol{\xi} = [\,\boldsymbol{\xi}_1 \quad \dots \quad \boldsymbol{\xi}_m\,]^\top$, where $\boldsymbol{\xi}_i = [\,\xi_1^i \quad \dots \quad \xi_{\rho_i}^i\,]^\top$, $i = 1, \dots, m$, and defined as:

$$\begin{aligned}
\xi_1^1 &:= h_1(\boldsymbol{x}), \quad & \xi_2^1 &:= \mathscr{L}_f h_1(\boldsymbol{x}), \quad & \cdots, \quad & \xi_{\rho_1}^1 := \mathscr{L}_f^{\rho_1 - 1} h_1(\boldsymbol{x}), \\
\xi_1^2 &:= h_2(\boldsymbol{x}), \quad & \xi_2^2 &:= \mathscr{L}_f h_2(\boldsymbol{x}), \quad & \cdots, \quad & \xi_{\rho_2}^2 := \mathscr{L}_f^{\rho_2 - 1} h_2(\boldsymbol{x}), \\
&\vdots & &\vdots & & \vdots \\
\xi_1^m &:= h_m(\boldsymbol{x}), \quad & \xi_2^m &:= \mathscr{L}_f h_m(\boldsymbol{x}), \quad & \cdots, \quad & \xi_{\rho_m}^m := \mathscr{L}_f^{\rho_m - 1} h_m(\boldsymbol{x}),
\end{aligned} \tag{10.21}$$

if the total relative degree is equal to the order of the system ($\rho = n$), the ρ–coordinates ξ_i^j, $j = 1, 2, \dots, m$; $i = 1, 2, \dots, \rho_j$, describe fully the nonlinear behavior of the original system, and moreover, the system is said to be full–state feedback linearizable. Otherwise, whenever the total relative degree is strictly less than the order of the system ($\rho < n$), a part of the system dynamics would become unobservable via input–output linearization using the new set of the ρ–coordinates ξ_i^j, and therefore, these coordinates do not fully describe the original system. In such case, the input–output linearization decomposes the dynamics of the nonlinear system into an external part (input–output), described by the ρ–coordinates, and an internal part (unobservable), described by a new set of $(n - \rho)$–coordinates and therefore called the internal dynamics of the system. The unobservable states, usually denoted as $\boldsymbol{\eta} = [\,\eta_1 \quad \dots \quad \eta_{n-\rho}\,]^\top$, are defined via smooth functions $\boldsymbol{\phi}(\boldsymbol{x}) = [\,\phi_1 \quad \dots \quad \phi_{n-\rho}\,]^\top$ in the neighborhood $\mathscr{D}_0$ of $\boldsymbol{x}$ as:

$$\boldsymbol{\eta} = \boldsymbol{\phi}(\boldsymbol{x}) \tag{10.22}$$

such that:

$$\frac{\partial \phi_i}{\partial \boldsymbol{x}} \boldsymbol{g}_j(x) = 0, \quad \text{for } 1 \le i \le n - \rho, \quad \text{for } 1 \le j \le m, \quad \forall \boldsymbol{x} \in \mathscr{D}_0. \tag{10.23}$$

Considering a new coordinate–system $\boldsymbol{z}$ defined as:

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\psi}(\boldsymbol{x}) \\ \boldsymbol{\phi}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix}, \tag{10.24}$$

where $\boldsymbol{T}$ represents a diffeomorphism on the domain $\mathscr{D}_0$, then the original nonlinear system can be transformed into the normal form [61, 62, 76, 97] as:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_c \boldsymbol{\xi} + \boldsymbol{B}_c \left[\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x}) \boldsymbol{u} \right] \Big|_{\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})} \tag{10.25a}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{f}_c(\boldsymbol{\xi}, \boldsymbol{\eta}) = \mathscr{L}_f \boldsymbol{\phi}(\boldsymbol{x}) \Big|_{\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \Big|_{\boldsymbol{x} = \boldsymbol{T}^{-1}(\boldsymbol{z})} \tag{10.25b}$$

$$\boldsymbol{y} = \boldsymbol{C}_c \boldsymbol{\xi} \tag{10.25c}$$

where the triplet $(\boldsymbol{A}_c, \boldsymbol{B}_c, \boldsymbol{C}_c)$ is in Brunovsky block canonical form, i. e., $\boldsymbol{A}_c = \text{diag}\{\boldsymbol{A}_o^i\}$, $\boldsymbol{B}_c = \text{diag}\{\boldsymbol{B}_o^i\}$, $\boldsymbol{C}_c = \text{diag}\{\boldsymbol{C}_o^i\}$, $i = 1, \dots, m$, where $(\boldsymbol{A}_o^i, \boldsymbol{B}_o^i, \boldsymbol{C}_o^i)$ is a canonical form repre-

sentation of a chain of ρ_i integrators:

$$\boldsymbol{A}_o^i := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & \cdots & 0 & 1 \\ 0 & 0 & & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{B}_o^i := \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{C}_o^i := \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad i = 1, \ldots, m. \tag{10.26}$$

The transformation $\boldsymbol{T}$ is required to be a diffeomorphism by the necessity of $\boldsymbol{T}$ being invertible (at least locally in $\mathscr{D}_0$), i.e., $\boldsymbol{T}^{-1}(\boldsymbol{T}(\boldsymbol{x})) = \boldsymbol{x}$, $\forall \boldsymbol{x} \in \mathscr{D}_0$, in order to recover the original state vector from the new coordinate $\boldsymbol{z}$, together with the necessity of $\boldsymbol{T}$ and its inverse $\boldsymbol{T}^{-1}$ to be smooth mappings in $\mathscr{R}^n$ guaranteeing that the description of the nonlinear system in the new coordinates is still a smooth one. Since it is generally difficult to find a diffeomorphism defined for all $\boldsymbol{x} \in \mathscr{R}^n$, the requirement of having a diffeomorphism well defined for all $\boldsymbol{x} \in \mathscr{D}_0 \subset \mathscr{R}^n$ makes it a local one. Defining the vector $\boldsymbol{\varphi}(\boldsymbol{x})$ and matrix $\boldsymbol{\vartheta}(\boldsymbol{x})$ as:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = -\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{l}(\boldsymbol{x}) \tag{10.27a}$$

$$\boldsymbol{\vartheta}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x}) \tag{10.27b}$$

and denoting $\boldsymbol{\nu}$ as a virtual control input, the state feedback control law $\boldsymbol{u}$ defined as:

$$\boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{x}) + \boldsymbol{\vartheta}(\boldsymbol{x})\boldsymbol{\nu} = \boldsymbol{M}^{-1}(\boldsymbol{x})\left[\boldsymbol{\nu} - \boldsymbol{l}(\boldsymbol{x})\right] \tag{10.28}$$

cancels all nonlinearities in closed-loop in absence of external disturbances and model uncertainties, resulting in the system:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_c\boldsymbol{\xi} + \boldsymbol{B}_c\boldsymbol{\nu} \tag{10.29a}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{f}_c(\boldsymbol{\xi}, \boldsymbol{\eta}) \tag{10.29b}$$

$$\boldsymbol{y} = \boldsymbol{C}_c\boldsymbol{\xi} \tag{10.29c}$$

which is still described in normal form and decomposed into an external (input–output) part and an internal (unobservable) part. This resulting system is now driven by the virtual control input and entirely described in the newly defined $\boldsymbol{z}$–coordinates $(\boldsymbol{\eta}, \boldsymbol{\xi})$. The equation $\dot{\boldsymbol{\eta}} = \boldsymbol{f}_c(\boldsymbol{0}, \boldsymbol{\eta})$ defines the zero–dynamics of the system which is defined as the internal dynamics that appear in the system when the input and the initial conditions are chosen such that the output is made or kept identically to zero for all $t \geq 0$. Furthermore, the system is said to be minimum phase if the zero–dynamics have an asymptotically stable equilibrium point in the domain of interest [76]. Referring back to the y_i differentiated outputs in Eq. (10.18), application of the control input in (10.28) results in m equations in the form of $y_i^{(\rho_i)} = \nu_i$, $i = \{1, \ldots, m\}$, or, with a slight abuse of notation, more compactly as:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{\nu} \tag{10.30}$$

and the sought linear input–output relationship between the new input $\boldsymbol{\nu}$ and the output $\boldsymbol{y}$ is obtained as long as $\boldsymbol{\vartheta} = \boldsymbol{M}^{-1}(\boldsymbol{x})$ is nonsingular. Apart from being linear, an interesting result from this relationship is that it is also decoupled since the input ν_i only affects

the differentiated output $y_i^{(\rho_i)}$. From this fact, the input transformation (10.28) is called a *decoupling control law*, and the linear system (10.30) results in an *integrator–chain*. The integrator–chain (10.30) is sought to be rendered exponentially stable with the proper design of $\boldsymbol{\nu}$, for instance such as:

$$\boldsymbol{\nu} = -k_0\boldsymbol{x} - k_1\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} - k_2\frac{\mathrm{d}^2\boldsymbol{x}}{\mathrm{d}t^2} - \cdots - k_{\boldsymbol{\rho}-1}\frac{\mathrm{d}^{(\boldsymbol{\rho}-1)}\boldsymbol{x}}{\mathrm{d}t^{(\boldsymbol{\rho}-1)}} \tag{10.31}$$

From this typical control problem it can be seen that the entire system will have two control loops [71, 78, 79]: the inner linearization loop (10.28), and the outer control loop (10.31). This resulting NDI control law depends on accurate knowledge of the model and its parameters, hence it is susceptible to model and parametric uncertainties contained in both $\boldsymbol{l}(\boldsymbol{x})$ and $\boldsymbol{M}(\boldsymbol{x})$.

Now we bring back the discussion into spacecraft attitude control for rigid spacecraft models as described in Eqs. (10.17a)–(10.17b), where in this case $n = 9$. Recall the output of the system to be the MRP vector $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\sigma} = [\sigma_1 \quad \sigma_2 \quad \sigma_3]^\top$, meaning that $p = 3$, and therefore the system has a vector of relative degree $\boldsymbol{\rho} = [\rho_1 \quad \rho_2 \quad \rho_3]^\top = [2 \quad 2 \quad 2]^\top$ and total relative degree $\rho = 6$. Since $\rho < n$, the input–output linearization will consist of both an external and an internal part, where the internal part is comprised of $n - \rho$ unobservable states. For the design of an attitude controller as in Eq. (10.28), first we obtain $\boldsymbol{l}(\boldsymbol{x})$ and $\boldsymbol{M}(\boldsymbol{x})$ as follows. First consider:

$$\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}, \tag{10.32}$$

then, $\boldsymbol{l}(\boldsymbol{x})$, which represents the Jacobian of the MRP kinematics, is given by:

$$\boldsymbol{l}(\boldsymbol{x}) = \mathscr{L}_f^2\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial\left[\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x})\right]}{\partial\boldsymbol{x}} = \frac{1}{4}\frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{x}} \tag{10.33}$$

$$= \frac{1}{4}\begin{bmatrix} \frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{\sigma}} & \frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{\omega}} & \frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{\Omega}} \end{bmatrix} \tag{10.34}$$

$$= \underbrace{\begin{bmatrix} \frac{1}{4}\frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{\sigma}} & \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma}) & \boldsymbol{0}_{3\times3} \end{bmatrix}}_{\text{purely kinematic and fully known}} \tag{10.35}$$

where:

$$\frac{1}{4}\frac{\partial\left[\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]}{\partial\boldsymbol{\sigma}} = \frac{1}{2}\begin{bmatrix} \omega_x\sigma_1 + \omega_y\sigma_2 + \omega_z\sigma_3 & \omega_z - \omega_x\sigma_2 + \omega_y\sigma_1 & \omega_z\sigma_1 - \omega_x\sigma_3 - \omega_y \\ \omega_x\sigma_2 - \omega_z - \omega_y\sigma_1 & \omega_x\sigma_1 + \omega_y\sigma_2 + \omega_z\sigma_3 & \omega_x - \omega_y\sigma_3 + \omega_z\sigma_2 \\ \omega_y + \omega_x\sigma_3 - \omega_z\sigma_1 & \omega_y\sigma_3 - \omega_x - \omega_z\sigma_2 & \omega_x\sigma_1 + \omega_y\sigma_2 + \omega_z\sigma_3 \end{bmatrix} \tag{10.36}$$

and therefore:

$$\boldsymbol{M}(\boldsymbol{x}) = \mathscr{L}_g\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial\left[\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x})\right]}{\partial\boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{l}(\boldsymbol{x})\cdot\boldsymbol{G}. \tag{10.37}$$

The attitude control design begins by first considering the new coordinate–system given by the local diffeomorphism (10.24) where the external states $\boldsymbol{\xi} = [\boldsymbol{\xi}_1 \quad \boldsymbol{\xi}_2 \quad \boldsymbol{\xi}_3]^\top$ are

given by $\boldsymbol{\xi}_i = [\xi_i^1 \quad \xi_i^2]^\top = [\sigma_i \quad \dot{\sigma}_i]^\top$, $i = 1,2,3$, and the choice for the remaining internal states $\boldsymbol{\eta} = \boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1 \quad \phi_2 \quad \phi_3]^\top$ are written in compact form as $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{H}_{GG} \cdot [\boldsymbol{\omega} \quad \boldsymbol{\Omega}]^\top$ where:

$$\boldsymbol{H}_{GG} = \begin{bmatrix} \boldsymbol{G}_{12}^{-1} & \boldsymbol{G}_{13}^{-1} \end{bmatrix}, \tag{10.38a}$$

$$\boldsymbol{G}_{12} = -\boldsymbol{A}\boldsymbol{I}^{-1}, \tag{10.38b}$$

$$\boldsymbol{G}_{13} = -(\boldsymbol{I} \cdot \boldsymbol{I}_{\boldsymbol{w}})^{-1}(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{I}_{\boldsymbol{w}}\boldsymbol{A}^\top), \tag{10.38c}$$

and furthermore, the choice of $\boldsymbol{H}_{GG}$ has been found such that the following condition:

$$\boldsymbol{\phi}(\boldsymbol{0}) = \boldsymbol{0}, \quad \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x}) = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{G} = \boldsymbol{0}, \tag{10.39}$$

is fulfilled according to the definition of $\boldsymbol{\phi}(\boldsymbol{x})$. Then the original nonlinear system can be transformed into the normal form (10.25) and the state feedback nonlinear dynamic inversion (NDI) control law $\boldsymbol{u}$ is obtained as in (10.28):

$$\boldsymbol{u} = [\boldsymbol{l}(\boldsymbol{x}) \cdot \boldsymbol{G}]^{-1} [\boldsymbol{\nu} - \boldsymbol{l}(\boldsymbol{x})] \tag{10.40}$$

while denoting $\boldsymbol{\nu}$ as a virtual control input to be considered later. With this NDI control law, and in absence of external perturbations and model uncertainties, the nonlinearity is canceled resulting in the nominal closed-loop system as:

$$\dot{\boldsymbol{\xi}}_{[6]} = \boldsymbol{A}_{c[6\times6]} \boldsymbol{\xi}_{[6]} + \boldsymbol{B}_{c[6\times3]} \boldsymbol{\nu}_{[3]} \tag{10.41a}$$

$$\dot{\boldsymbol{\eta}}_{[3]} = \boldsymbol{f}_c(\boldsymbol{\xi}_{[6]}, \boldsymbol{\eta}_{[3]}) \tag{10.41b}$$

$$\boldsymbol{y}_{[3]} = \boldsymbol{C}_{c[3\times6]} \boldsymbol{\xi}_{[6]} \tag{10.41c}$$

where the subscript indexes indicate the dimensions of the vectors and matrices, and $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$ are in Brunovsky block canonical form as in Eqs. (10.26). Furthermore, the internal dynamics characterized in the $\boldsymbol{x}$-coordinates are given by:

$$\boldsymbol{f}_c(\boldsymbol{\xi}, \boldsymbol{\eta})\Big|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} = \boldsymbol{G}_{12}^{-1}\dot{\boldsymbol{\omega}} - \boldsymbol{G}_{13}^{-1}\dot{\boldsymbol{\Omega}}_w$$

where the zero–dynamics are be given by:

$$\boldsymbol{f}_c(\boldsymbol{0}, \boldsymbol{\eta}) = \boldsymbol{0} \tag{10.42}$$

which makes the system marginally stable at the origin and around the small neighborhood $\mathscr{D}_0$ in consideration, and therefore the spacecraft attitude control system is non-minimum phase. Zero–dynamics defines the internal dynamics of the system when the input and the initial conditions are chosen such that the output is maintained identically to zero at all times. Therefore, whenever $\boldsymbol{\xi} = \boldsymbol{0}$ (which implies $\boldsymbol{\omega} = \boldsymbol{0}$) we obtain from definition that $\dot{\boldsymbol{\omega}} = \boldsymbol{0}$ and $\dot{\boldsymbol{\Omega}}_w = \boldsymbol{0}$. Note that nothing is said about the corresponding $\boldsymbol{\Omega}_w$ which shall remain constant and therefore responsible to keep the angular momentum fixed inertially. Furthermore, since $[\boldsymbol{l}(\boldsymbol{x}) \cdot \boldsymbol{G}]$ is not singular, the linear input–output relationship between $\ddot{\boldsymbol{\sigma}}$ and $\boldsymbol{\nu}$ is obtained as a double–integrator:

$$\ddot{\boldsymbol{\sigma}} = \begin{bmatrix} \ddot{\sigma}_1 \\ \ddot{\sigma}_2 \\ \ddot{\sigma}_3 \end{bmatrix} = \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \end{bmatrix} = \boldsymbol{\nu} \tag{10.43}$$

We denote $\boldsymbol{e} = \boldsymbol{\sigma}_e = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$, which is a valid representation of the MRP error for small rotations [198], and therefore $\dot{\boldsymbol{e}} = \dot{\boldsymbol{\sigma}}_e = \dot{\boldsymbol{\sigma}}_d - \dot{\boldsymbol{\sigma}}$. The double-integrator (10.43) can be therefore rendered exponentially stable with:

$$\boldsymbol{\nu} = \ddot{\boldsymbol{\sigma}}_d + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} \tag{10.44}$$

where $\ddot{\boldsymbol{\sigma}}_d$ is the feedforward reference term for tracking tasks, and $\mathbf{k}_D$ and $\mathbf{k}_P$ being 3×3 constant diagonal matrices whose $i-$th diagonal elements $\mathbf{k}_{D_i}$ and $\mathbf{k}_{P_i}$, respectively, are chosen so that the polynomials:

$$s^2 + \mathbf{k}_{D_i} s + \mathbf{k}_{P_i} \; i = 1, \ldots, n = 3 \tag{10.45}$$

may become Hurtwiz. This results in the exponentially stable and decoupled error dynamics:

$$\ddot{\boldsymbol{e}} + \mathbf{k}_D \dot{\boldsymbol{e}} + \mathbf{k}_P \boldsymbol{e} = \mathbf{0} \tag{10.46}$$

which implies that $\boldsymbol{e}(t) \to \mathbf{0}$ as $t \to \infty$.

10.3.2. INCREMENTAL NONLINEAR DYNAMIC INVERSION

The concept of incremental nonlinear dynamic inversion (INDI) amounts to the application of NDI to a system expressed in an incremental form. This improves the robustness of the closed-loop system as compared with conventional NDI since dependency on the accurate knowledge of the plant dynamics is reduced. Unlike NDI, this control design technique is *implicit* in the sense that desired closed-loop dynamics do not reside in some explicit model to be followed but result when the feedback loops are closed [73, 74]. To begin the discussion, we introduce a sufficiently small time–delay λ and define the following deviation variables:

$$\dot{\boldsymbol{x}}_0 := \dot{\boldsymbol{x}}(t - \lambda) \tag{10.47a}$$

$$\boldsymbol{x}_0 := \boldsymbol{x}(t - \lambda) \tag{10.47b}$$

$$\boldsymbol{u}_0 := \boldsymbol{u}(t - \lambda) \tag{10.47c}$$

which are λ–time–delayed signals of the current state derivative $\dot{\boldsymbol{x}}(t)$, state $\boldsymbol{x}(t)$, and control $\boldsymbol{u}(t)$, respectively. The *explicit* consideration of the time–delay λ in these deviation variables has not been widely considered in the literature; in reality, an infinitesimal time increment is not practically feasible and because of digital implementation of control systems, the lowest possible delay admissible by these assumptions is given by the actual sampling rate of the on-board digital computer. For highly sampled applications of INDI (100 – 1000 Hz), an associated (approximate) discrete–time model of the plant provides implicit consideration of the sampling time but most often this is done for the angular rate dynamics with a Taylor series approximation and not for the entire plant (with relative degree > 1).

Moreover, we will denote:

$$\Delta\dot{\boldsymbol{x}} := \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0 \tag{10.48a}$$

$$\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_0 \tag{10.48b}$$

$$\Delta\boldsymbol{u} := \boldsymbol{u} - \boldsymbol{u}_0 \tag{10.48c}$$

as the incremental state derivative, the incremental state, and the so–called incremental control input, respectively. To obtain an incremental form of system dynamics [4, 70, 71, 73–75, 80, 95], we consider the first–order Taylor series expansion of $\boldsymbol{y}^{(\boldsymbol{\rho})}$ as in [97, 98], not in the geometric sense, but with respect to the newly introduced $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{u}$ (functions of the time–delay λ) as:

$$\begin{aligned}\boldsymbol{y}^{(\boldsymbol{\rho})} &= \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \frac{\partial}{\partial \boldsymbol{x}}\Big[\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}\Big]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \Delta\boldsymbol{x} + \boldsymbol{M}(\boldsymbol{x}_0)\Delta\boldsymbol{u} + \mathscr{O}(\Delta\boldsymbol{x}^2)\\ &= \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{M}(\boldsymbol{x}_0)\Delta\boldsymbol{u} + \boldsymbol{N}(\Delta\boldsymbol{x}, \lambda)\end{aligned}$$

with:

$$\boldsymbol{y}_0^{(\boldsymbol{\rho})} = \boldsymbol{y}^{(\boldsymbol{\rho})}(t-\lambda) = \boldsymbol{l}(\boldsymbol{x}_0) + \boldsymbol{M}(\boldsymbol{x}_0)\boldsymbol{u}_0 \tag{10.49a}$$

$$\boldsymbol{N}(\boldsymbol{x}, \boldsymbol{u}, \lambda) = \left[\frac{\partial}{\partial \boldsymbol{x}}\Big[\boldsymbol{l}(\boldsymbol{x}) + \boldsymbol{M}(\boldsymbol{x})\boldsymbol{u}\Big]\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \Delta\boldsymbol{x} + \mathscr{O}(\Delta\boldsymbol{x}^2)\right]\Big|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} \tag{10.49b}$$

where $\boldsymbol{N}(\boldsymbol{x}, \boldsymbol{u}, \lambda)$ has the property [97, 98] of:

$$\lim_{\lambda \to 0} \|\boldsymbol{N}(\boldsymbol{x}, \boldsymbol{u}, \lambda)\| \to 0, \ \forall \boldsymbol{x} \in \mathscr{R}^n, \forall \boldsymbol{u} \in \mathscr{R}^m \tag{10.50}$$

This assumption of having high sampling rates such that (10.68) holds has been widely considered as valid in the incremental nonlinear dynamic inversion and incremental backstepping control literature. From now on, we shall refer to $\boldsymbol{M}(\boldsymbol{x}_0)$ as the *instantaneous control effectiveness (ICE)* matrix, meaning that this model–based term is sampled at each incremental instant. This leaves us with an approximate linearization about the $\lambda-$delayed signals that is performed *incrementally*, and not with respect to a particular equilibrium or operational point of interest. We now introduce a formal assumption to continue with the incremental nonlinear control design.

Time-scale separation (TSS) assumption: *For a sufficiently small time-delay λ and for any incremental control input, it is assumed that $\Delta\boldsymbol{x}$ does not vary significantly during λ. In other words, the input rate of change is much faster than the state rate of change, so this is seen as a time-scale separation assumption:*

$$\epsilon_{TSS}(t) \equiv \Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_0 \cong 0, \ \forall \ \Delta\boldsymbol{u} \tag{10.51}$$

which leads to:

$$\boldsymbol{N}(\boldsymbol{x}, \boldsymbol{u}, \lambda) \cong \mathscr{O}(\Delta\boldsymbol{x}^2)$$

and therefore:

$$\Delta\boldsymbol{y}^{(\boldsymbol{\rho})} \cong \boldsymbol{M}(\boldsymbol{x}_0)\,\Delta\boldsymbol{u} + \mathscr{O}(\Delta\boldsymbol{x}^2) \tag{10.52}$$

in other words:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} \cong \ \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{M}(\boldsymbol{x}_0)\,(\boldsymbol{u} - \boldsymbol{u}_0) + \mathscr{O}(\Delta\boldsymbol{x}^2). \tag{10.53}$$

This assumption shows that for small time increments (high sampling rates) the nonlinear system dynamics in its incremental form (in relative coordinates) are simply approximated by its instantaneous control effectiveness (ICE) matrix $\boldsymbol{M}(\boldsymbol{x}_0)$, i.e., the control effectiveness evaluated at the current state. Since this results in a change of coordinates (absolute to relative), the development of control laws in the original set of (absolute) coordinates implies or requires the availability of $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$ and $\boldsymbol{u}_0$ in (10.53). For the obtained approximation, ignoring the remainder, $\boldsymbol{y}^{(\boldsymbol{\rho})} \cong \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{M}(\boldsymbol{x}_0)(\boldsymbol{u}-\boldsymbol{u}_0)$, NDI is applied to obtain a relation between the incremental control input and the output of the system:

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{M}(\boldsymbol{x}_0)^{-1}(\boldsymbol{\nu} - \boldsymbol{y}_0^{(\boldsymbol{\rho})}). \tag{10.54}$$

Note that the incremental input $\boldsymbol{u}_0$ that corresponds to $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$ is measured or estimated from the output of the actuators, and it has been assumed that a commanded control is achieved sufficiently fast as to being able to neglect the effect of the actuator dynamics. The total control command along with the obtained linearizing control $\boldsymbol{u}_0 = \boldsymbol{u}(t-\lambda)$ can be rewritten as:

$$\boldsymbol{u}(t) = \boldsymbol{u}(t-\lambda) + \boldsymbol{M}(\boldsymbol{x}(t-\lambda))^{-1}\left[\boldsymbol{\nu} - \boldsymbol{y}^{(\boldsymbol{\rho})}(t-\lambda)\right] \tag{10.55}$$

and it is referred to as the incremental nonlinear dynamic inversion (*INDI*) control law. The dependency of the closed–loop system on accurate knowledge of the dynamic model in $\boldsymbol{l}(\boldsymbol{x})$ is largely decreased, improving robustness against model uncertainties contained therein. Therefore, this implicit control law design is more dependent on accurate measurements or accurate estimates of $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$, the state derivatives, and $\boldsymbol{u}_0$, the incremental control input, respectively. The canceling of all nonlinearities in closed–loop, in absence of external disturbances and model uncertainties, results in the system:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}_c\boldsymbol{\xi} + \boldsymbol{B}_c\left[\boldsymbol{\nu} + \boldsymbol{N}(\boldsymbol{x},\boldsymbol{u},\lambda)\right] \tag{10.56a}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{f}_c(\boldsymbol{\xi},\boldsymbol{\eta}) \tag{10.56b}$$

$$\boldsymbol{y} = \boldsymbol{C}_c\boldsymbol{\xi} \tag{10.56c}$$

which is still described in normal form. The stability and robustness properties of this system has been thoroughly studied in [97, 98] and the reader is referred to these references for more details. Referring back to the attitude control problem, since we will consider the dynamics in its incremental form as in (10.49a) for the control design, the application of *INDI* results in a control law that is only subject to sensor uncertainties and model uncertainties contained within the ICE matrix:

$$\boldsymbol{u}(t) = \boldsymbol{u}(t-\lambda) + \bar{\boldsymbol{M}}(\boldsymbol{x}(t-\lambda))^{-1}\left[\boldsymbol{\nu} - \boldsymbol{y}^{(\boldsymbol{\rho})}(t-\lambda)\right]. \tag{10.57}$$

However, notice that:

$$\boldsymbol{M}(\boldsymbol{x}(t-\lambda)) = \mathscr{L}_g\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x}(t-\lambda)) = \frac{\partial\left[\mathscr{L}_f^1\boldsymbol{h}(\boldsymbol{x}(t-\lambda))\right]}{\partial\boldsymbol{x}}\cdot\boldsymbol{g}(\boldsymbol{x}(t-\lambda)) \tag{10.58}$$

$$= \underbrace{\boldsymbol{l}(\boldsymbol{x}(t-\lambda))}_{\text{purely kinematic}}\cdot\underbrace{\boldsymbol{G}}_{\text{purely parametric}}. \tag{10.59}$$

This means that in the particular case of this plant, namely a rigid body spacecraft actuated with a non-redundant set of orthogonal reaction wheels and parameterized by MRPs, the incremental nonlinear dynamic inversion is robust against uncertainties. This is because the term $\boldsymbol{l}(\boldsymbol{x}(t-\lambda))$, contained in $\boldsymbol{M}(\boldsymbol{x}(t-\lambda))$ is given by a kinematic and known relationship (therefore void of uncertainty) which contains information only about the spacecraft inertia matrix and the inertias of the wheel–set in $\boldsymbol{G}$. Moreover, this purely kinematic term in the resulting control law is only subjected to the measured errors contained in $\boldsymbol{x}(t-\lambda)$. To conclude the INDI attitude control design, we have made use of the fact that:

$$\boldsymbol{y}_0^{(\boldsymbol{\rho})} = \ddot{\boldsymbol{\sigma}}_0 = \frac{1}{4}\left[\, \dot{\boldsymbol{B}}(\boldsymbol{\sigma}_0)\cdot\boldsymbol{\omega}_0 + \boldsymbol{B}(\boldsymbol{\sigma}_0)\cdot\dot{\boldsymbol{\omega}}_0 \,\right], \tag{10.60}$$

where the relationship:

$$\begin{aligned}\dot{\boldsymbol{B}}(\boldsymbol{\sigma}_0)\cdot\boldsymbol{\omega}_0 = \frac{1}{2}\big[\, & 2\boldsymbol{\sigma}_0^{\top}\boldsymbol{\omega}_0(1-\boldsymbol{\sigma}_0^{\top}\boldsymbol{\sigma}_0)\boldsymbol{\omega}_0 \\ & -(1+\boldsymbol{\sigma}_0^{\top}\boldsymbol{\sigma}_0)\boldsymbol{\omega}_0^{\top}\boldsymbol{\omega}_0\boldsymbol{\sigma}_0 - 4\boldsymbol{\sigma}_0^{\top}\boldsymbol{\omega}_0\boldsymbol{S}(\boldsymbol{\omega}_0)\,\boldsymbol{\sigma}_0 + 4(\boldsymbol{\sigma}_0^{\top}\boldsymbol{\omega}_0)^2\boldsymbol{\sigma}_0 \,\big]\end{aligned} \tag{10.61}$$

is given and therefore highly beneficial to compute $\ddot{\boldsymbol{\sigma}}_0$ analytically which is otherwise very hard to estimate with finite differences or by approximation since this would amplify the noise contained in measurements.

Remark 1: By using the measured $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$ and commanded $\boldsymbol{u}_0$ incrementally, we practically obtain a nonlinear 'self-scheduling' *NDI* control law that is robust to model and parametric uncertainties.

Remark 2: Notice, however, that this *INDI* control law is depending on the instantaneous control effectiveness (ICE) matrix reflected in $\boldsymbol{M}(\boldsymbol{x}_0)$, which in turn is only susceptible to parametric uncertainties in $\boldsymbol{G}$ that are related to inertia values of the rigid body and its reaction wheels. This remark gives a hint to one of the key differences with respect to *time-delay control* (*TDC*), where the control effectiveness is considered as a fixed–gain matrix instead.

10.4. INCREMENTAL NONLINEAR DYNAMIC INVERSION IN SAMPLED–DATA FORM

In this section we are interested in bridging the continuous–time derivation of INDI with respect to a discrete or sampled–time counterpart. In contrast to linear systems where exact sampled-data models can be obtained, for nonlinear systems a sampled–data model can only *approximate* the real dynamics up to a certain degree [195–197]. However, the accuracy of such models can be characterised in a precise way. Considering an analogous system to (10.16) but described as a sampled–data model, we will derive parallels in terms of incremental control design but recalling that these models are obtained as an approximation of the input–output mapping of the nonlinear forms already presented.

10.4.1. PRELIMINARIES

Since a sampled–data model is sought such that it closely approximates the nonlinear input–output mapping given in the previous section, we may obtain these approxima-

tions by considering that the control inputs $\boldsymbol{u}(t)$ are provided in sampled–time by a *zero–order hold* (ZOH). When assuming the input comes from a digital–to–analog converter as such zero–order hold signals, the input is hence generated as piecewise constant signals. Such piecewise constant input are provided in between sampling time intervals of amplitude λ are given by:

$$\boldsymbol{u}(t) := \boldsymbol{u}_k = \boldsymbol{u}(k\lambda), \quad \text{for} \quad t \in [k\lambda, (k+1)\lambda), \quad k \geq 0 \tag{10.62}$$

where $k \in \mathscr{Z}^+$ is the sampled or discrete–time index [196]. Furthermore, recalling the shift–operator q ($q\boldsymbol{x}_k = \boldsymbol{x}_{k+1}$), we will be interested in using the δ–operator [196, 197]:

$$\delta = \frac{q-1}{\lambda} \tag{10.63}$$

which corresponds to a reparameterization of sampled–data models that allows to explicitly include the sampling period in a discrete–time description [199]. Denoting $\boldsymbol{x}_k = [\, x_{k_1} \quad \cdots \quad x_{k_n} \,]^\top \in \mathscr{R}^n$ as the sampled–time state sequence, $\boldsymbol{y}_k = [\, y_{k_1} \quad \cdots \quad y_{k_m} \,]^\top \in \mathscr{R}^p$ as the sampled–time output sequence, and $\boldsymbol{u}_k = [\, u_{k_1} \quad \cdots \quad u_{k_m} \,]^\top \in \mathscr{R}^m$ as the piecewise continuous and sampled–time input sequence, the sampled–data (discrete–time) dynamics will result in a model of the form:

$$\delta \boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{g}_k(\boldsymbol{x}_k)\boldsymbol{u}_k \tag{10.64a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) \tag{10.64b}$$

where the functions $\boldsymbol{f}_k(\boldsymbol{x}_k) = [\, f_{k_1}(\boldsymbol{x}_k) \quad \cdots \quad f_{k_n}(\boldsymbol{x}_k) \,]^\top$, $\boldsymbol{g}_k(\boldsymbol{x}_k) = [\, \boldsymbol{g}_{k_1}(\boldsymbol{x}_k) \quad \ldots \quad \boldsymbol{g}_{k_m}(\boldsymbol{x}) \,]^\top \in \mathscr{R}^{n\times m}$, and $\boldsymbol{h}_k(\boldsymbol{x}_k) = [\, h_{k_1}(\boldsymbol{x}_k) \quad \cdots \quad h_{k_p}(\boldsymbol{x}) \,]^\top$ are assumed to be analytical approximations of the original nonlinear model, and therefore smooth vector fields continuously differentiable on $\mathscr{R}^n$.

10.4.2. A SAMPLED–DATA MODEL FOR DETERMINISTIC NONLINEAR SYSTEMS

Yuz and Goodwin [196, 197] presented a sampled–data model as in Eqs. (10.64a)-(10.64b) for deterministic nonlinear systems. The model results in a truncation error of order $[\, \lambda^{\rho_1+1} \quad \ldots \quad \lambda^{\rho_p+1} \,]^\top$ between the sampled–data model output $\boldsymbol{y}_k$ and the continuous–time output $\boldsymbol{y}(t)$ of the original system (10.16a)–(10.16b) at sampling instants $t = k\lambda$ when the inputs $\boldsymbol{u}(t)$ are generated from $\boldsymbol{u}_k$ as sampled–time and piece–wise constant (ZOH) control inputs. The fact that this model is close in a well defined sense to the continuous–time output helps to bridge the connection between continuous–time incremental nonlinear control methods and their discrete or sampled–time counterparts in appropriate fashion. Moreover, the key fact of using models described as in (10.64) is that these are models already given in *incremental form*, but in the discrete sense, i.e., as state transitions represented as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda \cdot \delta \boldsymbol{x}_k \tag{10.65}$$

where also the sampling time λ is considered *explicitly*. Denoting $\boldsymbol{\xi}_k = [\, z_{k_1} \quad \cdots \quad z_{k_\rho} \,]^\top$ and $\boldsymbol{\eta}_k = [\, z_{k_{\rho+1}} \quad \cdots \quad z_{k_n} \,]^\top$, we can consider the new coordinate–system defined as:

$$\boldsymbol{z}_k = \boldsymbol{T}_s(\boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{\psi}_s(\boldsymbol{x}_k) \\ \boldsymbol{\phi}_s(\boldsymbol{x}_k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\xi}_k \\ \boldsymbol{\eta}_k \end{bmatrix} \tag{10.66}$$

where clearly $\boldsymbol{z}_k = [\, z_{k_1} \quad \cdots \quad z_{k_n} \,]^\top$ and $\boldsymbol{T}_s$ defines the diffeomorphism on the domain $\mathscr{D}_0$, and then the original continuous-time nonlinear system (10.16a)–(10.16b) system can be approximated as the following discrete normal form [196, 197]:

$$\delta\boldsymbol{\xi}_k = \boldsymbol{A}_s\boldsymbol{\xi}_k + \boldsymbol{B}_s\left[\boldsymbol{l}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) + \boldsymbol{M}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\boldsymbol{u}_k\right] \tag{10.67a}$$

$$\delta\boldsymbol{\eta}_k = \boldsymbol{f}_s(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) \tag{10.67b}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_s\boldsymbol{\xi}_k \tag{10.67c}$$

where the triplet $(\boldsymbol{A}_s, \boldsymbol{B}_s, \boldsymbol{C}_s)$ is in Brunovsky block canonical form, i. e., $\boldsymbol{A}_s = \text{diag}\{\boldsymbol{A}_k^i\}$, $\boldsymbol{B}_s = \text{diag}\{\boldsymbol{B}_k^i\}$, $\boldsymbol{C}_s = \text{diag}\{\boldsymbol{C}_k^i\}$, $i = 1,\ldots,m$, and $(\boldsymbol{A}_k^i, \boldsymbol{B}_k^i, \boldsymbol{C}_k^i)$ are given as the following matrices [196, 197]:

$$\boldsymbol{A}_k^i := \begin{bmatrix} 0 & 1 & \lambda/2 & \cdots & \frac{\lambda^{r-2}}{(r-1)!} \\ 0 & 0 & 1 & \cdots & \frac{\lambda^{r-3}}{(r-2)!} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & \cdots & 0 & 1 \\ 0 & 0 & \cdots & & 0 \end{bmatrix}, \quad \boldsymbol{B}_k^i := \begin{bmatrix} \frac{\lambda^{r-2}}{(r-1)!} \\ \frac{\lambda^{r-3}}{(r-2)!} \\ \vdots \\ \lambda/2 \\ 1 \end{bmatrix}$$

$$\boldsymbol{C}_k^i := \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad i = 1,\ldots,m.$$

and therefore, the model is explicitly defined in terms of the approximated analytical functions $\boldsymbol{l}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)$, $\boldsymbol{M}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)$, $\boldsymbol{f}_s(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)$, and the system sampling time λ, which are based on the original continuous–time nonlinear model. Note that:

$$\lim_{\lambda\to 0}\boldsymbol{A}_k^i \to \boldsymbol{A}_o^i, \quad \lim_{\lambda\to 0}\boldsymbol{B}_k^i \to \boldsymbol{B}_o^i, \tag{10.68}$$

which recovers the original continuous time normal form for small sampling times [199]. Furthermore, the local truncation error between the output $\boldsymbol{y}_k = \boldsymbol{C}_c\boldsymbol{\xi}_k$ of the discrete-time nonlinear model and the true system output $\boldsymbol{y}(t)$ is of order λ^{r+1}. This results in a well approximated sampled–data system for sufficiently small sampling times and low relative degrees (1 or 2). Considering the sampled vector $\boldsymbol{\varphi}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)$ and sampled matrix $\boldsymbol{\vartheta}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)$ as:

$$\boldsymbol{\varphi}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) = \left[-\boldsymbol{M}^{-1}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\boldsymbol{l}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\right] \tag{10.69a}$$

$$\boldsymbol{\vartheta}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) = \boldsymbol{M}^{-1}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) \tag{10.69b}$$

and denoting $\boldsymbol{\nu}_k$ as the sampled–time virtual control input, the state feedback control law $\boldsymbol{u}_k$ defined as:

$$\boldsymbol{u}_k = \boldsymbol{\varphi}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) + \boldsymbol{\vartheta}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\boldsymbol{\nu}_k = \boldsymbol{M}^{-1}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\left[\boldsymbol{\nu}_k - \boldsymbol{l}(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k)\right] \tag{10.70}$$

cancels all nonlinearities in closed-loop in absence of external disturbances and model uncertainties, resulting in the system represented in the following approximated discrete normal form:

$$\delta\boldsymbol{\xi}_k = \boldsymbol{A}_s\boldsymbol{\xi}_k + \boldsymbol{B}_s\boldsymbol{\nu}_k \tag{10.71a}$$

$$\delta\boldsymbol{\eta}_k = \boldsymbol{f}_s(\boldsymbol{\xi}_k,\boldsymbol{\eta}_k) \tag{10.71b}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_c\boldsymbol{\xi}_k \tag{10.71c}$$

which is still described in normal form and decomposed into an external (input–output) part and an internal (unobservable) part. In this case, this sampled–data model is driven by the sampled–time virtual control input and approximated by the corresponding sampled $\boldsymbol{z}_k$–coordinates $(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)$. The sampled–data model (10.67) results in having a vector of relative degree $\boldsymbol{\rho}_k = [\rho_{k_1} \quad \dots \quad \rho_{k_p}]^\top = [1 \quad \dots \quad 1]^\top$ with respect to the output $\boldsymbol{y}_k = \boldsymbol{z}_{k_1}$ [196, 197] and furthermore, the discrete-time zero dynamics are given by two subsystems, namely the sampled counterpart of the continuous–time zero dynamics $\delta\boldsymbol{\eta}_k = \boldsymbol{f}_s(\boldsymbol{0}, \boldsymbol{\eta}_k)$ and a linear subsystem of dimension denoted as $\bar{\boldsymbol{\rho}} = [\rho_1 - 1 \quad \dots \quad \rho_p - 1]^\top$. From the approximated model (10.67) it is clear that we can collect the ρ shifted outputs with a slight abuse of notation as:

$$\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} = \boldsymbol{l}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k) + \boldsymbol{M}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)\boldsymbol{u}_k = \boldsymbol{\nu}_k \tag{10.72a}$$

and note that the relationship between the continuous–time $\boldsymbol{\rho}$ differentiated outputs and the shifted state variables is given by the forward Euler method also obtained by a truncated Taylor series expansion, i.e., as:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} \cong \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} = \frac{\boldsymbol{z}_{(\boldsymbol{\rho})(k+1)} - \boldsymbol{z}_{(\boldsymbol{\rho})(k)}}{\lambda} \tag{10.73}$$

where again, a linear input–output relationship between the $\boldsymbol{\nu}_k$ and the output $\boldsymbol{y}_k$ is precluded by the condition of $\boldsymbol{M}^{-1}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)$ being nonsingular.

10.4.3. INCREMENTAL NONLINEAR DYNAMIC INVERSION IN SAMPLED–DATA FORM

Notice that the control law uses the approximated models:

$$\boldsymbol{u}_k = \bar{\boldsymbol{M}}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)^{-1} \left[\boldsymbol{\nu}_k - \bar{\boldsymbol{l}}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)\right] \tag{10.74}$$

which causes a problem in terms of uncertainties since:

$$\delta\boldsymbol{\xi}_k = \boldsymbol{A}_s\boldsymbol{\xi}_k + \boldsymbol{B}_s\left\{\boldsymbol{l}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k) + \boldsymbol{M}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)\bar{\boldsymbol{M}}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)^{-1}\left[\boldsymbol{\nu}_k - \bar{\boldsymbol{l}}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k)\right]\right\} \tag{10.75}$$

resulting in:

$$\delta\boldsymbol{\xi}_k = (\boldsymbol{A}_s - \boldsymbol{K}\boldsymbol{B}_s)\boldsymbol{\xi}_k + \boldsymbol{\epsilon}_{NDI} \tag{10.76}$$

which is not linearizing because of the extra term $\boldsymbol{\epsilon}_{NDI}$ containing uncertain nonlinear terms [97, 98]. Because of such uncertainties, the *NDI* control law presented is actually not linearizing anymore. This major flaw of NDI–based control systems is well known, and also previously demonstrated by [71, 160, 162] for instance. In order to partially tackle this flaw of NDI, we are now interested in obtaining an INDI control but in a sampled–data framework. To design a sampled–data form incremental nonlinear dynamic inversion, first we introduce the following assumption.

Small time delay (STD) assumption: *For a sufficiently small time-delay λ so that $\boldsymbol{l}(\boldsymbol{\eta}^S, \boldsymbol{\xi}^S)$ and $\boldsymbol{M}(\boldsymbol{\eta}^S, \boldsymbol{\xi}^S)$ does not vary significantly during λ, we assume the following approximations to hold*:

$$\epsilon_{STD_l}(t) \equiv \boldsymbol{l}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k) - \boldsymbol{l}(\boldsymbol{\eta}_{k-1}, \boldsymbol{\xi}_{k-1}) \cong \boldsymbol{0}, \tag{10.77}$$

$$\epsilon_{STD_M}(t) \equiv \boldsymbol{M}(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k) - \boldsymbol{M}(\boldsymbol{\eta}_{k-1}, \boldsymbol{\xi}_{k-1}) \cong \boldsymbol{0}, \tag{10.78}$$

this implies that:

$$\boldsymbol{l}(\boldsymbol{\eta}_k,\boldsymbol{\xi}_k) \cong \boldsymbol{l}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1}) := \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)} - \boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1})\boldsymbol{u}_{k-1} \tag{10.79}$$

which, by defining the following deviation variables:

$$\Delta\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} := \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} - \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)}, \tag{10.80a}$$

$$\Delta \boldsymbol{u}_k := \boldsymbol{u}_k - \boldsymbol{u}_{k-1}, \tag{10.80b}$$

and in virtue of (10.78) *(as in the TSS assumption),* (10.79) *leads to*:

$$\Delta\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} \cong \boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1})\Delta \boldsymbol{u}_k \tag{10.81}$$

in other words:

$$\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k)} \cong \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)} + \boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1})\,(\boldsymbol{u}_k - \boldsymbol{u}_{k-1}) + \mathscr{O}(\lambda^{\boldsymbol{\rho}+1}). \tag{10.82}$$

Recalling the control law in Eq. (10.74), and considering our given STD assumption, we may now use:

$$\bar{\boldsymbol{l}}(\boldsymbol{\eta}_k,\boldsymbol{\xi}_k) = \boldsymbol{l}(\boldsymbol{\eta}^S_{k-1},\boldsymbol{\xi}^S_{k-1}) \tag{10.83}$$

which is given by Eq. (10.79) and by applying nonlinear dynamic inversion results into the following sampled–time INDI controller:

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + \bar{\boldsymbol{M}}^{-1}(\boldsymbol{\eta}_k,\boldsymbol{\xi}_k)\left[\boldsymbol{\nu}_k - \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)}\right] \tag{10.84}$$

where:

$$\bar{\boldsymbol{M}}(\boldsymbol{\eta}_k,\boldsymbol{\xi}_k) := \boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1}) \tag{10.85}$$

and:

$$\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)} = \frac{\boldsymbol{z}_{(\boldsymbol{\rho})(k-1)} - \boldsymbol{z}_{(\boldsymbol{\rho})(k-2)}}{\lambda} \tag{10.86}$$

Referring back to the attitude control problem, since we will depart from the (approximated) **discrete normal form** in (10.71) for the control design, the application of *INDI* results in a control law that is also depending only on the uncertainties contained within the ICE matrix:

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + \boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1})\left[\boldsymbol{\nu}_k - \delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)}\right] \tag{10.87}$$

However, notice that:

$$\boldsymbol{M}(\boldsymbol{\eta}_{k-1},\boldsymbol{\xi}_{k-1}) = \underbrace{\boldsymbol{l}(\boldsymbol{x}_{k-1})}_{\text{purely kinematic}} \cdot \underbrace{\boldsymbol{G}}_{\text{purely parametric}}. \tag{10.88}$$

The purely kinematic term in the resulting control law is only subjected to the measured errors contained in $\boldsymbol{x}^S$. For the sampled–time attitude control, we have the following sampled quantities:

$$\boldsymbol{\xi}^S = \boldsymbol{\psi}_s(\boldsymbol{x}^S) = \begin{bmatrix} z_1^S \\ z_2^S \\ z_3^S \\ z_4^S \\ z_5^S \\ z_6^S \end{bmatrix} = \begin{bmatrix} \sigma_1^S \\ \dot{\sigma}_1^S \\ \sigma_2^S \\ \dot{\sigma}_2^S \\ \sigma_3^S \\ \dot{\sigma}_3^S \end{bmatrix}, \tag{10.89}$$

and recall the system has a vector of relative degree $\boldsymbol{\rho} = [\rho_1 \quad \rho_2 \quad \rho_3]^\top = [2 \quad 2 \quad 2]^\top$. To conclude the *INDI* attitude control design, we have made use of the fact that, because of (10.86), we can rewrite:

$$\delta\dot{\boldsymbol{\sigma}}_k = \frac{\dot{\boldsymbol{\sigma}}_k - \dot{\boldsymbol{\sigma}}_{k-1}}{\lambda} \cong \ddot{\boldsymbol{\sigma}}_k \tag{10.90}$$

and therefore:

$$\delta \boldsymbol{z}_{(\boldsymbol{\rho})(k-1)} = \ddot{\boldsymbol{\sigma}}_{k-1} = \frac{1}{4}\left[\, \dot{\boldsymbol{B}}(\boldsymbol{\sigma}_{k-1})\cdot\boldsymbol{\omega}_{k-1} + \boldsymbol{B}(\boldsymbol{\sigma}_{k-1})\cdot\dot{\boldsymbol{\omega}}_{k-1} \,\right], \tag{10.91}$$

where the relationship:

$$\begin{aligned}\dot{\boldsymbol{B}}(\boldsymbol{\sigma}_{k-1})\cdot\boldsymbol{\omega}_{k-1} = \frac{1}{2}\big[\, & 2\boldsymbol{\sigma}_{k-1}^T\boldsymbol{\omega}_{k-1}(1-\boldsymbol{\sigma}_{k-1}^T\boldsymbol{\sigma}_{k-1})\boldsymbol{\omega}_{k-1} \\ & -(1+\boldsymbol{\sigma}_{k-1}^T\boldsymbol{\sigma}_{k-1})\boldsymbol{\omega}_{k-1}^T\boldsymbol{\omega}_{k-1}\boldsymbol{\sigma}_{k-1} - 4\boldsymbol{\sigma}_{k-1}^T\boldsymbol{\omega}_{k-1}\boldsymbol{S}(\boldsymbol{\omega}_{k-1})\boldsymbol{\sigma}_{k-1} + 4(\boldsymbol{\sigma}_{k-1}^T\boldsymbol{\omega}_{k-1})^2\boldsymbol{\sigma}_{k-1} \,\big]\end{aligned} \tag{10.92}$$

is given analytically. By virtue of eigenvalue assignment, we may find $\boldsymbol{\nu}_k$ such that:

$$\boldsymbol{\nu}_k = -\boldsymbol{K}\boldsymbol{\psi}(\boldsymbol{x}_k) = -\boldsymbol{K}\boldsymbol{\xi}_k \tag{10.93}$$

obtaining therefore the following closed-loop system in absence of model uncertainties and perturbations:

$$\delta\boldsymbol{\xi}_k = (\boldsymbol{A}_s - \boldsymbol{B}_s\boldsymbol{K})\,\boldsymbol{\xi}_k \tag{10.94a}$$

$$\delta\boldsymbol{\eta}_k = \boldsymbol{f}_s(\boldsymbol{\eta}_k, \boldsymbol{\xi}_k) \tag{10.94b}$$

$$\boldsymbol{y}_{k,CL} = \boldsymbol{C}_s\boldsymbol{\xi}_k \tag{10.94c}$$

where again, the local truncation error between the sampled output $\boldsymbol{y}_{k,CL}$ and $\boldsymbol{y}(t)$ is of order $[\lambda^{\rho_1+1} \quad \dots \quad \lambda^{\rho_p+1}]^\top$.

10.5. ATTITUDE CONTROL SIMULATIONS

For numerical simulations using the comprehensive analytical nonlinear model of Sec. I, a satellite is considered with an inertia matrix of

$$\boldsymbol{I} = \begin{bmatrix} 10 & 1 & 0.5 \\ 1 & 7 & 0.2 \\ 0.5 & 0.2 & 9 \end{bmatrix} \text{Kg}\cdot\text{m}^2.$$

and as main torque actuators an array of three *'High-Torque-Wheels'* (HTW) [25, 26] are considered in orthogonal configuration to demonstrate the high-agility attitude control capability of the system. Wheel characteristics for these *HTW*s are presented in Table 1, while the satellite's reaction wheel array alignment matrix in (10.1) is $\boldsymbol{A} = \text{diag}([1 \quad 1 \quad 1])$. The initial *HTW* wheel speeds are set to zero since we will consider these wheels only for agile reorientation; normally during operation, wheel speeds have some initial conditions that represents the angular momentum stored in the satellite platform. The nominal values presented in Table 10.1 are considered as the actuator limits in order to restrict

the demands of the atittude maneuver as to avoid the case when wheels must be saturated by their respective control commands.

Because the focus of this chapter was to introduce a sampled–data formulation for INDI controllers, we shall focus first on the numerical analysis without actuator and sensor dynamics in order to test this formulation in practice. It is well known from the literature [84, 85, 97, 200] that the actuator dynamics have to be "sufficiently" fast and also synchronized with the sensor and the filters in order to avoid undesired interaction between the controller and the plant due to the time–delayed sensor measurements, estimates, and incremental control actions. Not having synchronized signals between the time–delayed sensor signal and the time–delayed actuator signal might cause instability issues in INDI [84, 87]. In that sense, and actually corroborated in practice, the reaction–wheel "inner–loop" which is commanded by a desired torque provides the output at much higher rates from the ones to be simulated (100 Hz) and therefore, we neglect these actuator dynamics.

The MRP tracking reference commands are first illustrated in Fig. 1, which are smoothly up to a second order an obtained with a simple reference trajectory generator. The second derivative of these reference commands will result in piece–wise continuous functions that will act as feedforward acceleration commands. This is important because the resulting feedforward acceleration command will produce a moment for the maneuvers that corresponds closely to the path specified, hence why we shall also approximate the attitude error by the algebraic expression presented. This rest-to-rest maneuver is designed to achieve a desired final attitude by performing full three–axis control without an optimizer to design the resulting profile. It is important to mention that this trajectory is a time–dependent attitude path for the MRP specified a-priori, giving the satellite's on board computer the capability to perform agile maneuvers on demand with having to request an optimization routine to find a time optimal path dynamically. For all simulations we consider the virtual controller $\boldsymbol{\nu} = \ddot{\boldsymbol{y}}_d + \mathbf{k}_D\dot{\boldsymbol{e}} + \mathbf{k}_P\boldsymbol{e}$ so that the error dynamics are equivalent across the different scenarios considered

$$\ddot{\boldsymbol{e}} + \mathbf{k}_D\dot{\boldsymbol{e}} + \mathbf{k}_P\boldsymbol{e} = \mathbf{0} \tag{10.95}$$

Table 10.1: Wheel characteristics

Performance	*HTW*
Nominal speed [rpm]	1825
Max. speed [rpm]	3000
Nominal torque [Nm]	0.21
Max. torque [Nm]	0.23
Nominal ang. momentum [Nms]	0.9556
Max. ang. momentum [Nms]	1.5708
Mechanics	
Number of wheel units	3
Moment of inertia [Kg·m^2]	5×10^{-3}

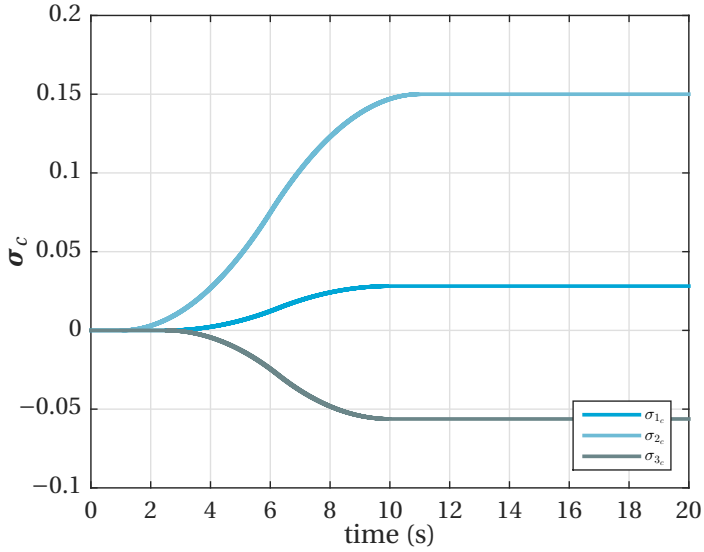


Figure 10.1: MRP reference tracking commands

This is a classical second order dynamics where considering a natural frequency $\omega_n = 3$ rad/s and damping coefficient of $\zeta = 0.707$ we can obtain the gains

$$\mathbf{k}_{D_i} = 2 \cdot \zeta \cdot \omega_n = 4.242, \; i = 1,2,3 \tag{10.96a}$$

$$\mathbf{k}_{P_i} = \omega_n^2 = 9, \; i = 1,2,3 \tag{10.96b}$$

The simulation results are as follows. Figure 2 presents the nominal performance of the INDI attitude control by illustrating the attitude tracking for the MRP reference maneuver commanded and its respective attitude tracking error. This first simulation is done at a sampling of 100 Hz which is the highest considered and it demonstrates the maneuver in this ideal case. For the attitude control motion obtained, Fig. 3 presents the resulting spacecraft angular velocity and its corresponding *HTW* input control commands using INDI control, showcasing that the limits have been avoided and that the platform is commanded with high-agility. A second simulation is performed where the sampling time has been set to 10 Hz. Figures 4 and 5 shows the resulting attitude control performance and the corresponding angular velocity and commanded control commands. From these figures it can be seen that the attitude control tracking error has been degraded but nevertheless remain quite small. By only changing the sampling time but leaving the rest of the controller structure as it was, it is seen that a down-scaling of the sampling time is feasible in this case. Of course, there will be a limit on how low can the sampling time be. For this case, we have also demonstrated that the platform can again be commanded with high-agility.

A third simulation is carried out, this time the sampling time has been set to 5 Hz. The simulation results, shown in Figures 6 and 7, demonstrates that still the attitude control is performing well at the expense of higher attitude tracking errors during the transient

phases. This nominal condition verifies that INDI control in sampled time still performs as expected. The fourth and final simulation presented in this chapter is performed by considering a sampling time of 2 Hz. The simulation results, shown in Figures 8 and

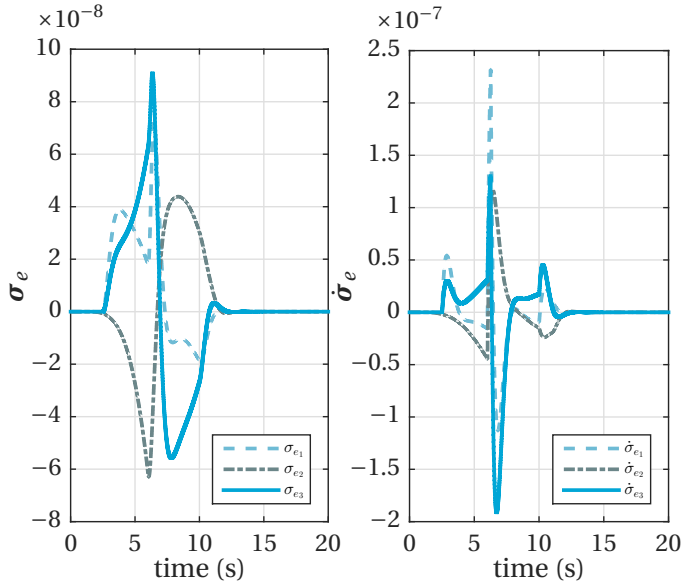


Figure 10.2: INDI control at 100 Hz: nominal MRP tracking errors during the fast slew maneuver.

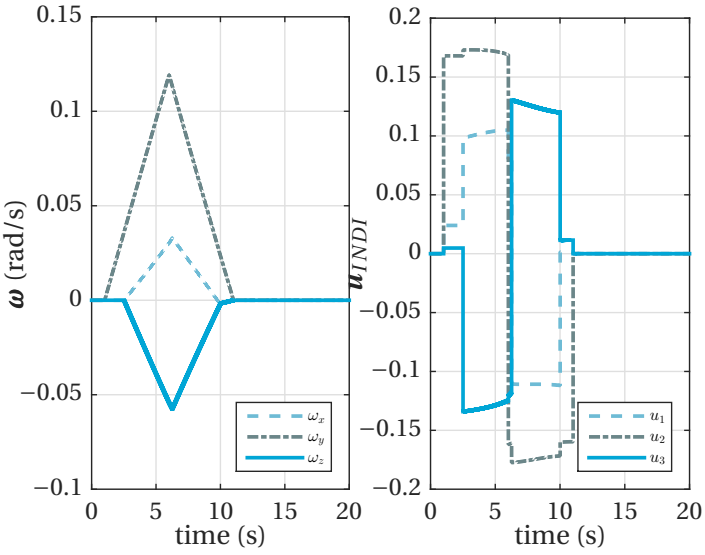


Figure 10.3: INDI control at 100 Hz: angular velocity (left) and commanded control input (wheel torques, right) during the fast slew maneuver.

9, again demonstrates that still the attitude control is still possible at the same expense of higher attitude tracking errors during the transient phase of the maneuver. One important aspect is to also study the performance under uncertainty. For that, we apply

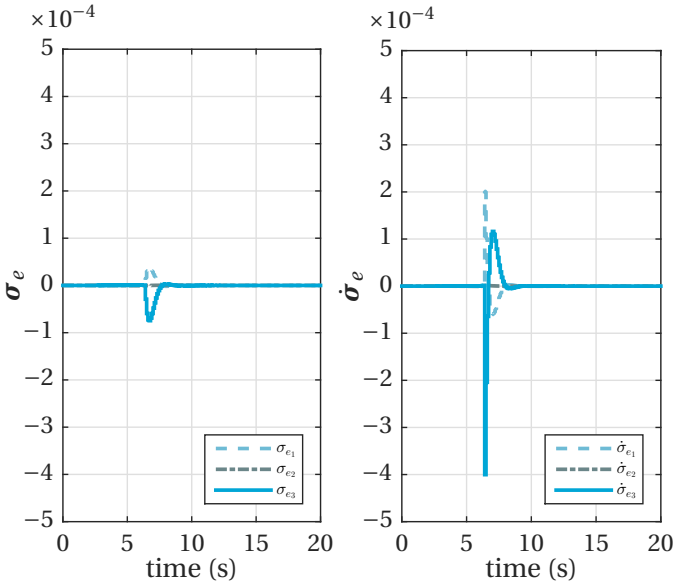


Figure 10.4: INDI control at 10 Hz: nominal MRP tracking errors during the fast slew maneuver.

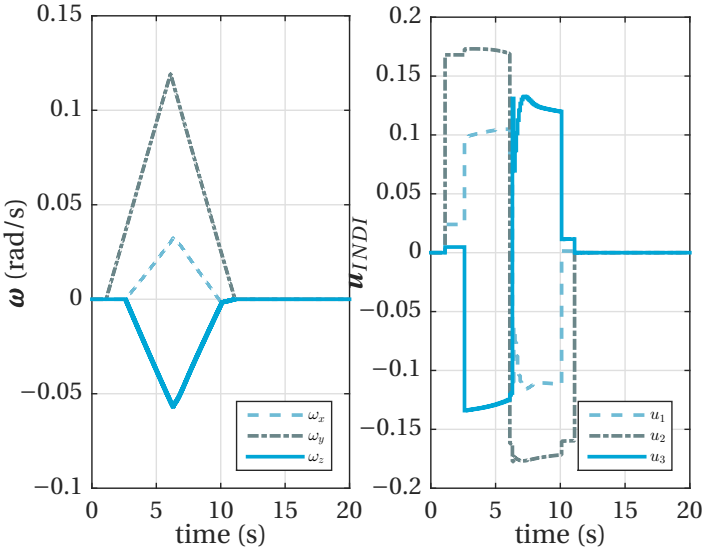


Figure 10.5: INDI control at 10 Hz: angular velocity (left) and commanded control input (wheel torques, right) during the fast slew maneuver.

uncertainty in the inertia matrix of the satellite platform and perform Monte-Carlo simulations with such parametric uncertainties. For the sampling rate set at 10 Hz it was found that indeed the controller remains robust against these uncertainties as the attitude tracking error was not severely degraded. This is also of course since the uncertainties of the inertia matrix are bounded and not overall damaging the stability properties of the closed loop system. The resulting tracking errors for one of the tests performed is shown in Fig. 10.

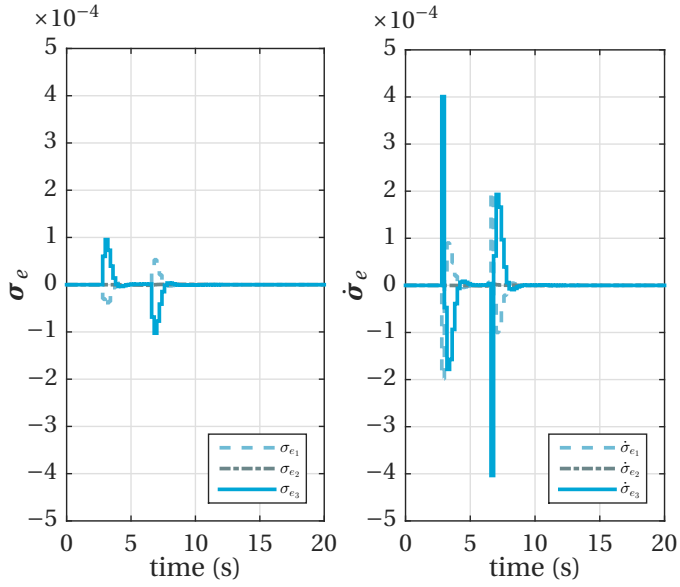


Figure 10.6: INDI control at 5 Hz: nominal MRP tracking errors during the fast slew maneuver.

In summary, simulation results shows that an agile attitude control system using INDI is promising because of the robustness properties as well for the capability to track agile maneuvers effectively. Having used a sampled–data approach also allows to consider the sampling–time explicitly in the formulations and also to study what happens while decreasing the achievable sampling time of the control computer. In [96, 201] it was shown a relationship between INDI and TDC/PID control where the influences of the parametric uncertainty on the robust performance and stability have been shown to be superior for INDI in comparison to TDC/PID control. This clearly suggests, that although they may have similar performance, the INDI control laws possess better robustness and stability properties. The systematic gain tuning and self scheduling property of our INDI controller have been shown to be scaled and readily applied to attitude control of rigid spacecraft for agile maneuvers that do not saturate the actuators; this issue will be addressed in future research.

10.6. CONCLUSIONS

In this chapter a sampled–data form of the recently reformulated incremental nonlinear dynamic inversion (INDI) is proposed and applied in the context of spacecraft attitude

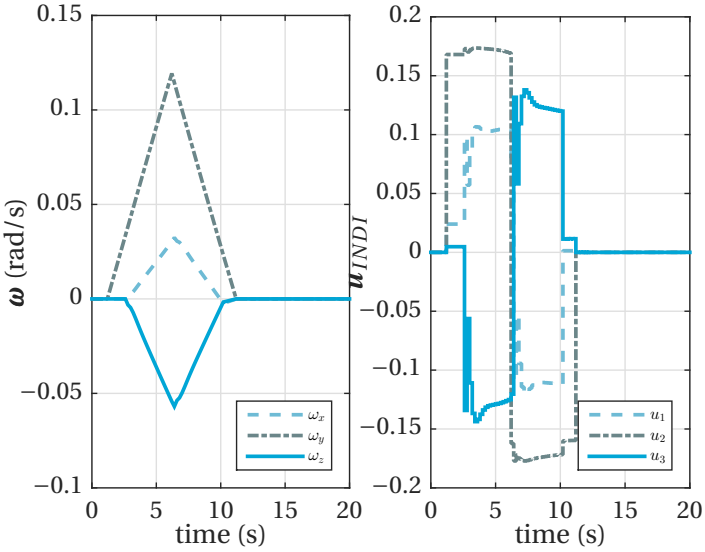


Figure 10.7: INDI control at 5 Hz: angular velocity (left) and commanded control input (wheel torques, right) during the fast slew maneuver.

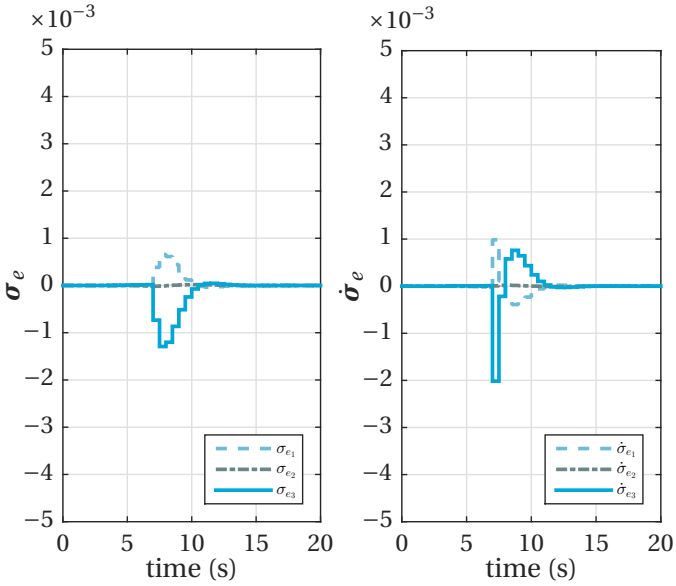


Figure 10.8: INDI control at 2 Hz: nominal MRP tracking errors during the fast slew maneuver.

control. The objective was to bridge the gap between highly sampled INDI formulations (100 – 1000 Hz) and their lowly sampled counterparts in the context of spacecraft attitude control where low sampling rates are common (1 – 10 Hz) . This was done by introducing

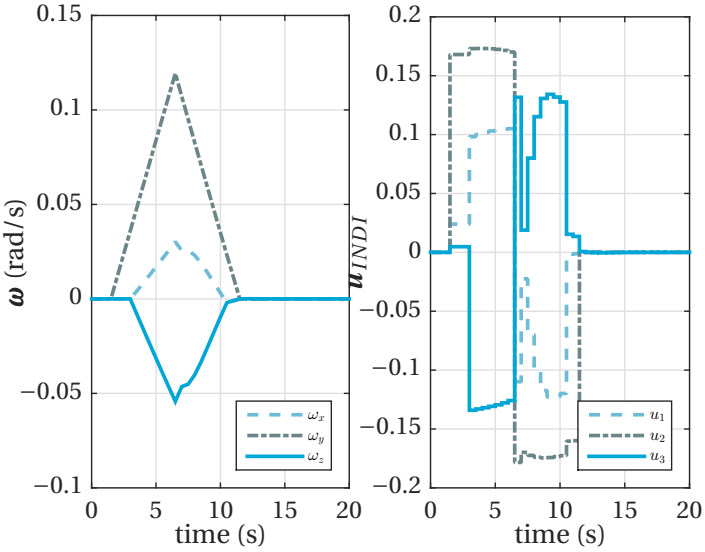


Figure 10.9: INDI control at 2 Hz: angular velocity (left) and commanded control input (wheel torques, right) during the fast slew maneuver.

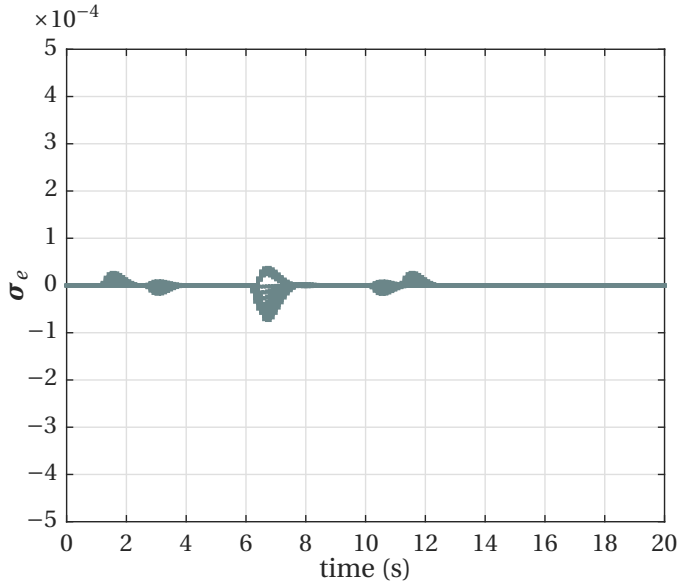


Figure 10.10: INDI control at 10 Hz: nominal MRP tracking errors during the fast slew maneuver.

a sampled–data reformulation of INDI that allows explicit consideration of the sampling time via an approximate sampled–data model in normal form available in the literature. Having an explicit consideration of the sampling time in the plant and in the controller

is important because there is a limit where the stability and performance of the resulting closed–loop nonlinear system can be compromised. Moreover, this sampled–data model is better suited for the INDI controller since it can be applied for plants with higher relative degree than one.

Regarding the attitude control, it was done in terms of *Modified Rodrigues Parameters* (MRPs) where the scheduling of the time–varying control effectiveness is achieved with the Jacobian of the MRP kinematics. This is an improvement over similar control strategies previously developed for rigid body spacecraft since it results in an architecture without a cascaded inner–loop based on a time–scale separation assumption for the rate loop which is commonly done. This results in the control effectiveness of the control loop composed entirely of kinematic (fully known) and parametric terms, making it useful as a scheduling term for the robust nonlinear controller. This resulting non–cascaded inversion–based architecture only requires an accurate control effectiveness model and measurements of the state and some of its derivatives resulting in a combined model– and sensor–based control approach. The resulting sampled–data INDI control is still robust up to a certain sampling time and it remains only sensitive to parametric uncertainties.

The systematic gain tuning and self scheduling property of this INDI controller can be scaled and readily applied to the robust and nonlinear attitude control of rigid spacecraft for any agile maneuvers that does not saturate the actuators. The effect of actuator dynamics and the synchronization of discrete signals at low sampling times for this type of controllers is to be considered in future studies. Simulations experiments for several sampling times shows the effectiveness of this method.

11

CONCLUSIONS AND RECOMMENDATIONS

This thesis considered methods for dynamics modeling, simulation, and control of aerospace vehicles. The methods proposed are geared towards preliminary design studies of space launchers and preliminary aspects in guidance and control (G&C) design. Further focus was given to studying incremental nonlinear control methods in the context of robust attitude control systems design. This chapter presents the main conclusions of the research conducted, where new challenges are identified and are formulated as recommendations for future research.

11.1. CONCLUSIONS

THIS THESIS was motivated on designing robust nonlinear attitude control laws for aerospace applications and departed from the fundamental question:

> *How can the incremental nonlinear control approach be applied to improve agility and robustness of aerospace vehicles' attitude control systems?*

In order to investigate potential applications of incremental nonlinear control, the focus was given first on the dynamics modeling for preliminary design studies, and then shifted to guidance and control aspects; first on a broad level, then on a more specific level. To answer the above question in detail, the thesis was divided in three parts. Each part is discussed below in terms of their individual research question where an overview of the contributions and main findings is therein presented.

PART I: DYNAMICS MODELING FOR PRELIMINARY DESIGN STUDIES

The first part of the thesis considered the question:

Research Question 1

How can an integrated, acausal, and multidisciplinary approach for modeling and simulation support preliminary design studies of space launch vehicles?

This question was answered in a broad sense by **Chapters 2** and **3**, where the main objective was to investigate an alternative approach to the methodologies found in recent literature [1, 116–120, 124–126] by considering an *acausal* or *declarative* modeling approach with the MODELICA language as introduced in *Section 1.2*. In fact, the starting point of this modeling approach was upon extension of the *DLR Space Systems Library* [111] by considering physics–based modeling of subsystems and components related to launch vehicle system dynamics. As mentioned in *Section 1.2*, such modeling frameworks were already introduced in many other fields [105, 107–110, 149, 150].

Chapter 2 showed that such an acausal and multidisciplinary modeling framework implemented with the MODELICA modeling language [37, 38] can enable engineers and scientists to assist preliminary design efforts in launch vehicle design [161]. This is thanks to the object–oriented, equation–based, and acausal modeling features of the MODELICA. To demonstrate benefits of this approach, a launch vehicle multibody dynamics model was described and implemented for a simplified application example. The potential of this modeling and simulation framework not only spans preliminary design phases, but could also support activities concerning more detailed system design, software and component verification and validation, i. e., to support several use cases across the whole launch vehicle program life cycle. As an example of the contribution of this modeling framework, a preliminary design study for a *reaction control system* (RCS), in the context of critical analyses performed at DLR concerning the next generation of Ariane 6 configurations, was carried out using the methods presented here [202].

Chapter 3 considered the study of stage separation dynamics modeling as a critical capability for launch vehicle design studies [152]. In fact, the development of stage separation dynamics allows performing end–to–end launch vehicle trajectory simulations by profiting from the object–oriented and equation–based acausal modeling properties of the MODELICA modeling language. It is shown that the acausal modeling features of MODELICA allow an easy implementation of the *Constraint Force Equation* (CFE) methodology [1, 117, 153], where the internal joint loads of a multi–stage space launcher can be obtained *automatically* while complying with constraints related to composite flight dynamics or during stage separation. This is a complex problem since such automatic joint load computation can be seen as a redundant set of multi–body constraints that are, in general, not easily solvable. This capability to study separation dynamics in the development of next generation space launchers was, moreover, easily integrable in the overall framework introduced in Chapter 2. As an example application, part of the work developed in this chapter contributed to the ESA study *Upper Stage Attitude Control Design*

Framework (USACDF), led by Astrium GmbH as part of Europe's Future Launchers Preparatory Program (FLPP). Moreover, a fairing separation dynamics modeling and analysis of the VLM–1 launch vehicle fairing separation process was performed at DLR in order to determine possible collision scenarios with its payload [203]. Clearance regions were obtained for given sets of initial angular velocities that the launch vehicle should maintain during the fairing separation phase in order to avoid a collision between the fairing and its payload.

Main findings of Chapters 2 and 3

- The object–oriented, equation–based, acausal modeling features of MODELICA strongly support preliminary studies in launch vehicle design with an integrated and multidisciplinary modeling framework.
- MODELICA allows for an easy implementation of the *Constraint Force Equation* (CFE) methodology, where the internal joint loads of a multi–stage space launcher are obtained automatically while complying with different constraints (composite flight or separation dynamics).

The first research question considered and presented the first building blocks towards a framework that enables physical modeling of conventional and non–conventional launch vehicles, and facilitates early developments regarding preliminary vehicle design. However, the efforts outlined here were limited in their scope and capabilities; for instance, they had not considered in much detail the aerodynamics and environment modeling [204] and they were not easily integrable with optimization tools. It also lacked of a consistent kinematic parameterization and many other features like component reusability, objected–orientation and easiness of use. For those reasons, a more advanced modeling and optimization framework has been developed in [3, 159] where substantial improvements were made. Such improvements have allowed to consider the multidisciplinary modeling capability in combination with multi–objective optimization, and as a result, this was considered in the next research question.

PART II: AEROSPACE GUIDANCE AND CONTROL (G&C)

The second part of the thesis considered the question:

Research Question 2

How can model–based nonlinear control and multi–objective optimization be combined for the study of preliminary guidance and control (G&C) aspects of reusable launch vehicles and spacecraft slew maneuvers?

This question was answered by **Chapters 4** and **5**, where the main objective was to in-

vestigate how model–based nonlinear control design and multi–objective optimization could be useful and considered at early design stages of G&C activities in aerospace. Multi–objective optimization, broadly speaking, delivers the best possible compromise between commonly existing conflicting goals, while providing reference trajectories or guidance commands for subsequent inner–loop attitude control systems. This approach is in fact widely used in the aeronautics community [52–59] and the potential of such integrated model–based approach was also shown in [58–60].

In **Chapter 4**, a *guidance and control* (G&C) architecture was presented for the early controllability study of reusable launch vehicles (RLVs) [193]. The architecture combines optimal guidance commands together with inner–loop attitude control obtained via *nonlinear dynamic inversion* (NDI). The reference trajectory and the optimal guidance commands were obtained with the modeling improvements in [3, 159] that are implemented in combination with the trajectory optimization package *'trajOpt'* [2] of the optimization tool *MOPS* ('Multi-Objective Parameter Synthesis') [52–54]. In the context of aerospace applications, NDI control is usually derived for high performant control systems design [60, 67–70]. In this chapter, however, another benefit of NDI was presented; namely, that NDI in combination with trajectory optimization can provide an early assessment of a vehicle's controllability. This is possible since NDI 'cancels the nonlinearities' (feedback linearization) in the nonlinear system so that the closed-loop dynamics are rendered into a linear form; therefore, NDI provides a direct link to the required angular impulse across the vehicle's trajectory. With that knowledge, it can be checked whether the plant can be controlled along the designed trajectory or drive some requirements in terms of vehicle design. For the obtained system, linear control techniques such as linear PID–control can be applied successfully for achieving desired closed-loop dynamics [49–51], hence eliminating the need of linearizing and designing different controllers for several operational points as in gain–scheduling.

To demonstrate the integrated approach, this method has been considered in the DLR projects AKIRA and X-TRAS regarding preliminary system studies and evaluation of key technologies for future reusable launch vehicles. In particular, this G&C architecture was tested on the AURORA reusable launch vehicle concept [18], where nonlinear flight simulations for the descent phase (including the re–entry) were considered. The simulations covered a wide flying envelope ranging from Mach 18 to Mach 5 and angles of attack between 50 and 9 deg. The results demonstrate the controllability of the launch vehicle as well as the potential to reduce *more than half* the impact on the angular impulse budget for the reaction control system (RCS) by combining it with aerodynamic surface controls during the re–entry phase. This could in turn translate to less propellant mass needed for the RCS, and therefore, better performance of the launcher.

Chapter 5 addressed the extensive topic of optimal reorientation in spacecraft attitude G&C [127–135], and more specifically, the main challenge that arises when discrete–time sampled inputs are required for slewing the continuous–time spacecraft dynamics in agile fashion [96]. This problem was motivated to design a high–agility attitude control system for the small satellite BIROS [27, 28, 96] which is actuated in sampled–time by a redundant array of *'High-Torque-Wheels'* [25, 26]. This is complex not only because of the nonlinearities involved, but also because time–optimal slew maneuvers are, in general, not of the *Euler–axis* rotation type [131, 136, 137]. By formulating the problem as

a constrained nonlinear optimal control problem, solutions can be obtained by solving multi–criteria optimization problems using a direct approach with the trajectory optimization package *'trajOpt'* [2] of the optimization tool *MOPS* ('Multi-Objective Parameter Synthesis') [52–54]. Results of this method are presented considering the sequential methodology or procedure proposed in *Section 1.2*, and are shown based on numerical simulations performed with a nonlinear spacecraft dynamics model of the small satellite BIROS.

Main findings of Chapters 4 and 5

- A preliminary G&C architecture containing NDI control can be considered for controllability assessments and as a design driver during preliminary launch vehicle design studies.
- Fast slew maneuvers can be designed for a spacecraft commanded with discrete–time sampled inputs by formulating the problem as a constrained nonlinear optimal control problem.
- Numerical solutions to this nonlinear optimal control problem can be readily obtained by solving multi–criteria optimization problems using a direct approach and trajectory optimization.

From this research question, it can be concluded that multi–objective optimization techniques, combined with model–based nonlinear control, facilitates early and preliminary guidance and control (G&C) studies very efficiently. However, several limitations were found during these studies:

- Regarding NDI control design, simplifying assumptions were considered about the plant invertibility (assuming that control derivatives are invertible in the domain of operation) and the absence of internal dynamics (the relative degree of each input–output channel was one).
- Simulations were shown for the G&C system without the consideration of model and parametric uncertainties. Aspects of robustness in nonlinear attitude control design are treated in the next research question.
- The time–optimal slew maneuvers were obtained off–line, hence, are not real–time implementable as it would be desired for an agile spacecraft. This motivates the development of an agile attitude control system that is real–time capable and implementable on board the spacecraft. The aspect concerning agile spacecraft attitude control design in closed-loop feedback form is also considered in the next research question.

PART III: ROBUST NONLINEAR ATTITUDE CONTROL

The third and final part of the thesis considered the question:

Research Question 3

How can incremental nonlinear controls be integrated with, e.g., backstepping, time–delay control (TDC), or nonlinear PID–control? And how can these incremental nonlinear control methods be applied for agile and robust nonlinear spacecraft attitude control?

The first part of this question was answered in **Chapters 6** and **7** and was motivated from a generalization of the *incremental nonlinear control approach* as introduced in *Section 1.2*, since most of the research in this area stems from an NDI control approach [4, 70, 71, 73–75, 80]. Relating to other nonlinear control design methodologies was also motivated out of curiosity.

Incremental backstepping [139] was first applied for robust nonlinear attitude control of rigid spacecraft with the motivation to combine the design of increments of control action with the recursive step-by-step procedure of the backstepping control design methodology [6, 65, 66]. In **Chapter 6**, incremental backstepping is further considered as a methodology for robust nonlinear flight control [95]. The main motivation to do this was to investigate how to deal with aerodynamic uncertainties and unmodeled dynamics that arise in flight control systems with a robust, sensor–based control approach. With such an approach, feedback control dependency on the modeled vehicle dynamics is greatly reduced, overcoming one of the major robustness flaws of conventional model–based flight control systems [60, 67–70].

Incremental nonlinear control requires information of the actuator states and the vehicle's rotational acceleration in order to reduce feedback sensitivities to an inaccurate baseline or airframe model [71, 77–80]. In that regard, in order to consider the recursive nature of the Lyapunov-based nonlinear design method backstepping, the information of a control derivatives model and the deflections of the aerodynamic control surfaces are required, together with a model structure that is in strict–feedback (cascaded) or lower triangular form [6]. This last requirement may be problematic for some configurations where the control deflections *also* enter the kinematic equations, as is the case in highly aggressive longitudinal dynamics control. This method allows to stabilize or track outer–loop control variables of multi–loop nonlinear systems *incrementally*, while accounting for model and parametric uncertainties that may rise during such aggressive maneuvers. The potential of incremental backstepping was demonstrated with a longitudinal nonlinear flight control example adapted from [6, 184], where good tracking performance was obtained while being subjected to relatively large variations in the vehicle's aerodynamic model parameters.

At this stage, some limitations were found. Namely, the incremental method did not consider details arising in highly uncertain and more advanced multivariable flight control applications and did not treat stability and robustness aspects too well. As pointed out in *Section 1.2*, these aspects have been recently treated and solved in [97–99], where incremental sliding mode control was proposed and a reformulated INDI structure was considered. However, the methodology has been further considered in the research community where many advancements of the method have been proposed. For instance, it

has been applied to active fault-tolerant control (also from a singular perturbations approach) [81–83], to adaptive flight control [205], to quadrotors [143] and to robust flight control [206] including real flight tests on small (FASER) aircraft [88] and large passenger (Cessna Citation II, PH-LAB) aircraft [90]. More recently, the method has been also extended to sliding mode fault-tolerant flight control [207].

Main findings of Chapter 6

- Augmenting incremental nonlinear control with the recursive step–by–step procedure of backstepping, *incremental backstepping* (IBKS), results in a promising methodology for robust nonlinear flight control systems.
- IBKS can exploit most of the flexibility inherent in backstepping designs, e. g., to retain stabilizing nonlinearities and to handle multiple-loops in a single and integrated incremental control law.

Chapter 7 presented an equivalence of *incremental nonlinear dynamic inversion* and *time–delay control* [140–142] when a reformulation of the plant control effectiveness is considered [96, 142]. This was motivated as a follow–up from the last Research Question 2 and also with the curiosity to study whether incremental nonlinear controls are related to other nonlinear control methods. TDC, more commonly known in the motion control and robotics community, is a nonlinear control technique that estimates and compensates disturbances and system uncertainties by utilizing time–delayed signals of some of the system variables. Moreover, Chang and Jung [142] found the relationship and equivalence between discrete formulations of TDC and *proportional-integral-derivative* (PID) for nonlinear plants of second–order controller canonical form, and in the context of a robot motion control application. This original result was then related to the found equivalence between INDI and TDC by considering *sufficiently small* time–delayed signals explicitly, the reformulation of the plant control effectiveness [142], and fixed–value gains in the PID control structure.

This brings a new interpretation of INDI that leads to a meaningful and systematic method for tuning of nonlinear PID flight control systems via INDI as it was done for robotics in [142]. This can be achieved by first imposing desired error dynamics, as usual for dynamic inversion control laws, and then, a mapping into an equivalent *incremental* nonlinear PID controller can be established with knowledge on the control derivatives. Incremental nonlinear PIDs are PIDs with state-dependent gains that are implemented in a discrete or sampled–time form, where the integral term can be replaced by considering a recursive computation of the error signals in consideration. Furthermore, their state-dependent gains might not necessarily be gain–scheduled but rather model–based. A simple nonlinear longitudinal dynamics example demonstrates this equivalence in simulation.

Main findings of Chapter 7

- For a nonlinear longitudinal flight control example, INDI control is equivalent to discrete time–delay control when considering a reformulation of the plant dynamics as originally suggested in the TDC literature [142].
- The previous finding also suggests that nonlinear PID for flight control systems can be also be obtained and tuned in a more meaningful way via INDI [96, 142].

The second part of Research Question 3 was considered in **Chapters 8, 9** and **10** and was motivated from the fact that incremental nonlinear dynamic inversion has not been widely treated for space applications in the literature. As mentioned in *Section 1.2*, INDI has been elaborated and applied theoretically in the past decade for advanced flight control applications [70, 71, 73–75, 80] and more recently for adaptive control of quadrotors [84, 85].

Chapter 8 presented an application of incremental nonlinear dynamic inversion control for robust nonlinear spacecraft attitude control [4]. The application considered the attitude tracking and disturbance rejection problem of rigid spacecraft subjected to model and parametric uncertainties. This is initially achieved with a cascaded two–loop control system, using as outer-loop control the kinematic inversion of the minimal set of attitude parameters known as *Modified Rodrigues Parameters* (MRP). Assuming a time scale separation of the attitude and rate dynamics, the rate control for the inner loop was done using incremental nonlinear dynamic inversion of the plant dynamics. As an improvement versus model–based nonlinear dynamic inversion control [60, 67, 69, 70], the INDI approach enhances robustness capabilities by reducing feedback control dependency on accurate knowledge of the system dynamics.

However, these kind of sensor–based incremental nonlinear control laws have the drawback of depending on accurate actuator output and angular acceleration measurements which may not be readily available on board or which may have to be estimated from rate measurements and state estimation. These measurements in turn may also contain noise, biases, and delays; therefore, these effects should be properly considered during control design. INDI therefore implies a trade–off between accurate knowledge of the dynamic model and accurate knowledge of the sensors and actuators of the spacecraft [4], and is more suitable than identification or model–based adaptive control architectures. Simulation results demonstrate the capabilities of the proposed INDI controller in terms of efficient tracking and external disturbances rejection capabilities by considering the combined effect of disturbances, time–delay, and parametric uncertainty.

In **Chapter 9** the recent reformulation of incremental nonlinear dynamic inversion in [97, 98] is considered to design a nonlinear and agile spacecraft attitude control system. The improvement over the INDI controller of the previous chapter is made by designing a full three–axis attitude control for a spacecraft actuated by three reaction wheels, and also in terms of the Modified Rodrigues Parameters but without the cascaded inner–loop that was based on the assumption of time–scale separation. Moreover, this reformulation of INDI for attitude control exploits an an analytic expression of the

MRP acceleration [194] which allows to decouple the nonlinear plant. This in turn results in each wheel–input MRP–output channel to be rendered as as a double integrator. It is shown that scheduling of the *instantaneous control effectiveness (ICE)*, as introduced in *Section 1.2*, can be done with the Jacobian of the MRP kinematics and is only subject to parametric uncertainty of the spacecraft augmented inertia and its wheelset alignment matrix. Moreover, as in **Chapter 7**, relationships between INDI, TDC, and nonlinear PID control were found. These relationships demonstrate that for the class of input–affine nonlinear systems considered in this thesis, INDI control can be recasted as incremental nonlinear PID control, and vice–versa. Finally, the nonlinear control law proposed can be analyzed with the stability and robustness results already obtained in [97–99, 140–142]. Simulation experiments for this particular problem demonstrate that INDI has similar nominal performance as TDC/PID control, but superior robust performance and stability.

Main findings of Chapters 8 and 9

- The INDI control approach is promising for spacecraft attitude control, in particular for agile reorientation maneuvers since it is robust against model and parametric uncertainty as well as capable to reject external disturbances very effectively.
- With the recently reformulated INDI [97, 98], a full three–axis agile attitude control system in terms of Modified Rodrigues Parameters can be derived without the classical cascaded inner–loops that are based on a time-scale separation assumption.
- Considering an analytical expression of the Modified Rodrigues Parameter attitude acceleration [194], the scheduling of the instantaneous control effectiveness can be done with the Jacobian of the MRP kinematics and is only subject to parametric uncertainty of the spacecraft augmented inertia and its wheelset alignment matrix.
- For the class of input–affine nonlinear systems considered, relationships between INDI, time–delay control, and nonlinear PID control can be found. These relationships can be useful for closed–loop gain tuning [140–142], and for stability and robustness analysis [97–99].

Chapter 10 presents a sampled–data form of the recently reformulated incremental nonlinear dynamic inversion (INDI) applied for robust spacecraft attitude control. Most of the INDI derivations proposed in the literature assume a very high sampling rate of the system and its controller while also not explicitly considering the available sampling time of the digital control computer. Neglecting the sampling time and its effect in the controller derivations can lead to stability and performance issues of the resulting closed–loop nonlinear system. In that sense, the contribution is aimed to bridge the gap between continuous–time and highly sampled INDI formulations (100 – 1000 Hz) and

their discrete and lowly sampled counterparts in the context of spacecraft attitude control where low sampling rates are common (1 – 10 Hz). This was done by introducing a sampled–data reformulation of INDI that allows explicit consideration of the sampling time via an approximate sampled–data model in normal form already known in the literature [196, 197]. The resulting sampled–data INDI control is still robust up to a certain sampling time since it remains only sensitive to parametric uncertainties. Simulation experiments for this particular problem demonstrate tha INDI attitude control is still possible for low sampling control rates.

Main findings of Chapter 10

- The INDI control approach can be formulated in the context of sampled–data nonlinear control. This is possible by considering a sampled–data model of the nonlinear dynamics of the plant in normal form available from the literature [196, 197].
- The sampled–data model considered is suited for the newly reformulated INDI controller since it can be applied for plants with higher relative degree than one.
- The systematic gain tuning and self scheduling property of this INDI controller can be scaled and readily applied to the robust and nonlinear attitude control of rigid spacecraft for any agile maneuvers that does not saturate the actuators.

Finally, to conclude on this last research question:

- Incremental nonlinear control can be integrated with backstepping, time–delay control, and nonlinear PID control;
- Incremental nonlinear control laws can be regarded as both model– and sensor–based, where 'model' refers to the scheduling of the instantaneous control effectiveness;
- Several applications and scenarios of robust nonlinear attitude control which aim to close the gap in terms of agility, robustness, and performance of future attitude control systems were considered.

11.2. RECOMMENDATIONS

The following recommendations are presented for future research.

- In terms of 'incremental control inputs' for real–time applications, special care must be taken. In particular, how the increments of the control input and how the actuator output are obtained or implemented is of high importance. Referring to Fig. 11.1 [4, 95], there will be a difference if the actuator output is directly measured or obtained from a model. Referring to Figs. 11.2(a), 11.2(b), and 11.2(c), these

kind of implementations lead to *infinite* or *high-gain* control. Meaning that special care must be taken, in particular when the closed-loop system is noisy which is common. The excitation of high frequencies with high–gain control can lead to system instability.

- More efforts should be done in finding the relationship of incremental nonlinear control with early works on time–delay control, pioneered by Hsia *et al.* [208–212] and Youcef–Toumi *et al.* [140–142]. Most of the work published in the literature on these subjects can be found in motion control systems, robotics, and nonlinear control.

- This thesis did not consider control input constraints and actuator limits. These aspects are very important, in particular for agile attitude control systems where exploiting the full capacity of the actuators might be necessary. This aspect raises several questions, such as how to design anti–windup, filtered command references, or pseudo–control hedging strategies for incremental (nonlinear) control systems?

- Another question is how to generate inverse models using the acausal methods considered in this thesis, in particular to be used in combination with model–based optimization. This can potentially help to design off–line and on–line robust guidance approaches for nonlinear flight control systems.

- State estimation was not considered in this thesis. This is usually done at another layer of a 'navigation' module of a GNC architecture. The practical aspects of having a combined guidance, navigation, and control simulation for real applications together with incremental nonlinear control should be further assessed. In some applications, the navigation module might 'block' or require some dedicated time–slot for estimation and fusion of sensor measurements, this in turn can potentially compromise the assumption of having a sufficiently fast control update rate for control.

- In this thesis it was suggested how a nonlinear PID controller for a class of input–affine nonlinear systems can be tuned via INDI. It would be interesting to find similar, meaningful and systematic ways to tune incremental nonlinear controllers, especially for the case of incremental backstepping and adaptive model–based incremental nonlinear control.

- The attitude control problem using Modified Rodrigues Parameters possesses very interesting optimality properties that were not addressed or exploited further in this thesis [188]. It is recommended to study these further, particularly for applications involving agile attitude control.

- In the literature exists a vast body of work around *incremental stability* [213] concepts. The relationship of the incremental nonlinear control approach considered in this thesis with such stability concepts should be looked into.

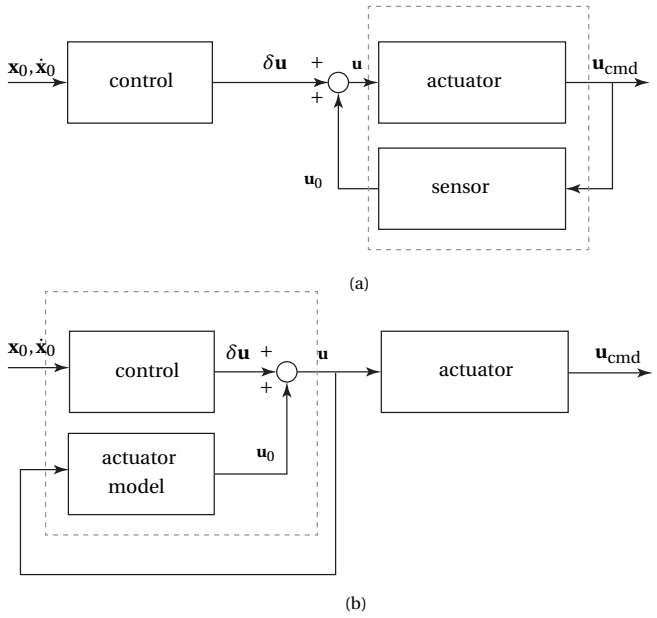
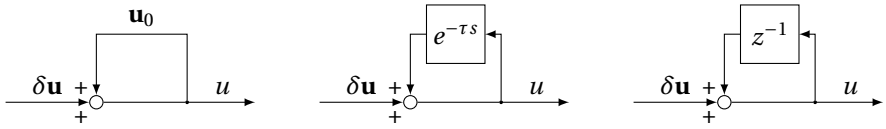


Figure 11.1: Actuator state measurement/estimation architectures for incremental backstepping: (a) sensor-dependent. (b) model-dependent.



(a) Infinite gain loop.

(b) High-gain loop (continuous).

(c) High-gain loop (discrete).

Figure 11.2: Control input gain loops.

REFERENCES

[1] M. Toniolo, P. V. Tartabini, and B. N. Pamadi, *Constraint Force Equation Methodology for Modeling Multi-Body Stage Separation Dynamics,* in *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit* (2008).

[2] K. Schnepper, *Trajektorienoptimierung in MOPS - Das Paket trajOpt Version 1.0*, Tech. Rep. (DLR German Aerospace Center, 2014).

[3] L. E. Briese, K. Schnepper, and P. Acquatella B., *Advanced Modeling and Trajectory Optimization Framework for Reusable Launch Vehicles,* in *Proceedings of the 2018 IEEE Aerospace Conference* (2018).

[4] P. Acquatella B., W. Falkena, E.-J. van Kampen, and Q. P. Chu, *Robust Nonlinear Spacecraft Attitude Control Using Incremental Nonlinear Dynamic Inversion,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference* (2012).

[5] A. Kopp, *Das AURORA-R2 RLV-Konzept,* in *Deutscher Luft- und Raumfahrtkongress* (2017).

[6] L. Sonneveldt, *Adaptive Backstepping Flight Control for Modern Fighter Aircraft* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2010).

[7] S. Stappert, J. Wilken, M. Sippel, and I. Dietlein, *Evaluation of European Reusable VTVL Booster Stages,* in *Proceedings of the AIAA Space and Astronautics Forum and Exposition* (2018).

[8] E. Dumont, S. Stappert, and J. Wilken, *Evaluation of a Future Reusable Ariane VTOL Booster,* in *Proceedings of the 68th International Astronautical Congress IAC*, IAC-17-D.2.4.3 (2017).

[9] S. Stappert and E. Dumont, *Reusability of Launcher Vehicles by the Method of SpaceX,* Tech. Rep. SARTTN007/2016 (DLR German Aerospace Center, 2016).

[10] S. Stappert, J. Wilken, M. Sippel, and E. Dumont, *Assessment of a European Reusable VTVL Booster Stage,* in *SART-TBC* (2018).

[11] M. Sippel, C. Manfletti, and H. Burkhardt, *Longterm/strategic Scenario for Reusable Booster Stages,* Acta Astronautica **58**, 209 (2006).

[12] M. Sippel, *Promising Roadmap Alternatives for the SpaceLiner,* Acta Astronautica **66**, 1652 (2010).

[13] M. Sippel, E. Dumont, and I. Dietlein, *Investigations of Future Expendable Launcher Options,* in *Proceedings of the 63rd International Astronautical Congress IAC* (2011).

[14] E. Dumont, L. Bussler, S. Karl, D. Porrmann, C. Manfletti, D. Krause, J. Klevanski, V. Clark, O. Bozic, and G. Poppe, *Analysis of the Ariane 62/64 and Vega-C Launcher Family*, Tech. Rep. DLR-IB-RY-HB-2016-63 (DLR German Aerospace Center, 2016).

[15] M. Sippel, S. Stappert, and L. Bussler, *Systematic Assessment of a Reusable First-stage Return Options,* in *proceedings of the 68th International Astronautical Congress IAC*, IAC-17-D2.4.4 (2017).

[16] M. Sippel, C. Valluchi, L. Bussler, A. Kopp, N. Garbers, S. Stappert, S. Krummen, and J. Wilken, *Spaceliner Concept as Catalyst for Advanced Hypersonic Vehicles Research,* in *proceedings of the 7th European Conference for Aeronautics and Space Sciences (EUCASS)* (2017).

[17] L. Bussler and M. Sippel, *Comparison of Return Options for Reusable First Stages,* in *21st AIAA International Space Planes and Hypersonics Technologies Conference* (2017).

[18] A. Kopp, S. Stappert, D. Mattsson, K. Olofsson, E. Marklund, G. Kurth, E. Mooij, and E. Roorda, *The Aurora Space Launcher Concept,* CEAS Space Journal **10**, 167 (2018).

[19] P. Rickmers, W. Bauer, M. Sippel, and S. Stappert, *ReFEx: Reusability Flight Experiment, A Flight Experiment to Demonstrate Controlled Aerodynamic Flight from Hypersonic to Subsonic Velocities with a Winged RLV,* in *Proceedings of the 7th European Conference for Aeronautics and Space Sciences EUCASS* (2017).

[20] W. Bauer, P. Rickmers, A. Kallenbach, S. Stappert, R. Schwarz, M. Sagliano, J. S. Haeseker, A. K. Flock, T. Thiele, A. Bierig, and J. Windelberg, *Upcoming DLR Reusability Flight Experiment,* in *Proceedings of the 68th International Astronautical Congress IAC* (2017).

[21] P. Rickmers, W. Bauer, M. Sippel, S. Stappert, R. Schwarz, M. Sagliano, G. F. Trigo, G. Wuebbels, and H. Martens, *An Update of the Upcoming DLR Reusability Flight Experiment – ReFEx,* in *Proceedings of the 69th International Astronautical Congress IAC* (2018).

[22] M. Sagliano, G. F. Trigo, and R. Schwarz, *Preliminary Guidance and Navigation Design for the Upcoming DLR Reusability Flight Experiment (ReFEx),* in *Proceedings of the 69th International Astronautical Congress IAC* (2018).

[23] C. H. Merrem, V. Wartemann, and T. Eggers, *Preliminary Aerodynamic Design of a Reusable Booster Flight Experiment,* in *Proceedings of the 2018 International Conference on High-Speed Vehicle Science Technology (HiSST)* (2018).

[24] M. Sippel, J. Klevanski, A. van Foreest, A. Gülhan, B. Esser, and M. Kuhn, *The Spaceliner Concept and its Aerothermodynamic Challenges,* in *Proceedings of the 1st ARA-Days* (2006).

[25] C. Raschke, T. Terzibaschian, and W. Halle, *High Agility Demonstration with a New Actuator System by Small Satellite BIROS,* in *Proc. of the 9th Airtec, Frankfurt/Main* (2014).

[26] C. Raschke, T. Terzibaschian, and W. Halle, *A New Actuator System for High Agility Demonstration with the Small Satellite BIROS,* in *Proc. of the 10th IAA Symposium on Small Satellites for Earth Observation, Berlin* (2015).

[27] W. Halle, T. Terzibaschian, and K.-D. Rockwitz, *The DLR-BIROS-Satellite for Fire-Detection and Technological Experiments,* in *Proc. of the 10th IAA Symposium on Small Satellites for Earth Observation, Berlin* (2015).

[28] W. Halle, T. Terzibaschian, W. Bärwald, and C. Schultz, *The DLR Small-Satellite Constellation FireBIRD,* in *Proc. of the 31st International Symposium on Space Technology and Science (ISTS), Matsuyama-Ehime* (2017).

[29] Wang, Xinwei and Han, Chao and Zhang, Rui, *Multiple Agile Earth Observation Satellites, Oversubscribed Targets Scheduling Using Complex Networks Theory,* arXiv e-prints , arXiv:1805.05053 (2018).

[30] C. Han, X. Wang, G. Song, and R. Leus, *Scheduling Multiple Agile Earth Observation Satellites with Multiple Observations,* arXiv e-prints , arXiv:1812.00203 (2018).

[31] J. Aldinger and J. Lohr, *Planning for Agile Earth Observation Satellites,* Albert-Ludwigs-Universitat Freiburg, Institut fur Informatik (2013).

[32] G. Beaumet, G. Verfaillie, and M.-C. Charmeau, *Autonomous Planning for an Agile Earth-Observing Satellite,* Albert-Ludwigs-Universitat Freiburg, Institut fur Informatik (2007).

[33] Z. Yuan, Y. Chen, and R. He, *Agile Earth Observing Satellites Mission Planning Using Genetic Algorithm Based on High Quality Initial Solutions,* in *IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China* (2014).

[34] K. Briess, W. Bärwald, E. Gill, H. Kayal, O. Montenbruck, S. Montenegro, W. Halle, W. Skrbek, H. Studemund, T. T., and H. Venus, *Technology Demonstration by the BIRD-mission,* Acta Astronautica **56**, 57 (2005).

[35] B. Zhukov, K. Briess, E. Lorenz, D. Oertel, and W. Skrbek, *Detection and Analysis of High-temperature Events in the BIRD mission,* Acta Astronautica **56**, 65 (2005).

[36] MATLAB, *Version R2016b* (Natick, Massachusetts, United States, The Mathworks, Inc., 2016).

[37] H. Elmqvist, S. E. Mattsson, and M. Otter, *Modelica - An International Effort to Design an Object-Oriented Modeling Language,* in *Proceedings of the SCSC'98 - Summer Computer Simulation Conference* (1998).

[38] S. E. Mattson, H. Elmqvist, and M. Otter, *Physical System Modeling with Modelica,* Control Engineering Practice **6**, 501 (1998).

[39] M. Tiller, *Introduction to Physical Modeling with Modelica* (The Springer International Series in Engineering and Computer Science, 2001).

[40] M. Otter and H. Olsson, *New Features in Modelica 2.0,* in *Proceedings of the 2nd International Modelica Conference* (2002).

[41] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.* (John Wiley & Sons, 2004).

[42] L. Petzold, *A description of DASSL: A differential/algebraic system solver*, Tech. Rep. SAND82-8637, Sandia National Laboratories (NASA, 1982).

[43] K.E. Breman, S.L. Campbell, and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations* (Elsevier Science Publishing Company, 1989).

[44] F. Cellier and H. Elmqvist, *Automated Formula Manipulation Supports Object Oriented Continuous-System Modeling,* IEEE Control Systems **13**, 28 (1993).

[45] H. Elmqvist and M. Otter, *Methods for Tearing Systems of Equations in Object Oriented Modeling,* in *Proceedings of ESM'94, European Simulation Multiconference. Barcelona, Spain* (1994).

[46] C. C. Pantelides, *The Consistent Initialization of Differential-Algebraic Systems,* SIAM Journal on Scientific and Statistical Computing **9**, 213 (1998).

[47] F. Magnusson, K. Berntorp, B. Olofsson, and J. Akesson, *Symbolic Transformations of Dynamic Optimization Problems,* in *Proceedings of the 10th International Modelica Conference* (2014) pp. 1027–1036.

[48] J. Bals, G. Hofer, A. Pfeiffer, and C. Schallert, *Object-Oriented Inverse Modelling of Multi-Domain Aircraft Equipment Systems and Assessment with Modelica,* in *Proceedings of the 3rd International Modelica Conference* (2003) pp. 377–384.

[49] A. J. Roenneke and K. H. Well, *Nonlinear Flight Control for a High-lift Rentry Vehicle,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference* (1995).

[50] R. R. da Costa, Q. P. Chu, and J. A. Mulder, *Reentry Flight Controller Design Using Nonlinear Dynamic Inversion,* Journal of Spacecraft and Rockets **40**, 64 (2003).

[51] S. Juliana, *Re-entry Flight Clearance* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2006).

[52] H.-D. Joos, *A Methodology for Multi-objective Design Assessment and Flight Control Synthesis Tuning,* Aerospace Science and Technology **3**, 161 (1999).

[53] H.-D. Joos, J. Bals, G. Looye, K. Schnepper, and A. Varga, *A Multi-objective Optimisation-based Software Environment for Control Systems Design,* in *Proc. of the IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Design (CCA/CACSD), Glasgow* (2002).

[54] H.-D. Joos, *MOPS - Multi-Objective Parameter Synthesis, User's Guide V6.6*, Tech. Rep. DLR-Internal report DLR-IB-SR-OP-2016-128 (DLR German Aerospace Center, 2016).

[55] H.-D. Joos, *RCAM Design Challenge Presentation Document: Multi-Objective Parameter Synthesis (MOPS)*, Tech. Rep. GARTEUR/TP-088-16 (DLR German Aerospace Center, 1996).

[56] H.-D. Joos, A. Varga, R. Finsterwalder, and J. Bals, *Eine Integrierte Optimierungsbasierte Entwurfsumgebung für Flugregelungsaufgaben,* at-Automatisierungstechnik **47**, 239 (1999).

[57] H.-D. Joos and R. Finsterwalder, *Multi-objective Design Assessment and Control Law Synthesis Tuning for Flight Control Development,* in *Proc. of the IEEE International Symposium on Computer Aided Control System Design (CACSD), Hawaii* (1999).

[58] G. Looye and H.-D. Joos, *Design of Autoland Controller Functions with Multi-objective Optimization,* in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA* (2002).

[59] G. Looye and H.-D. Joos, *Design of Autoland Controller Functions with Multiobjective Optimization,* Journal of Guidance, Control and Dynamics **29**, 475 (2006).

[60] G. H. Looye, *An Integrated Approach to Aircraft Modelling and Flight Control Law Design*, Ph.D. thesis, Delft University of Technology, Faculty of Aerospace Engineering (2008).

[61] J. J. Slotine and W. Li, *Applied Nonlinear Control* (Prentice Hall Inc, 1990).

[62] A. Isidori, *Nonlinear Control Systems*, 3rd ed. (Springer, 1985).

[63] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems* (Springer, 1990).

[64] P. Ioannou and J. Sun, *Robust Adaptive Control* (Prentice Hall, Inc, 1996).

[65] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design* (John Wiley & Sons, 1995).

[66] P. V. Kokotović and M. Arcak, *Constructive Nonlinear Control: A Historical Perspective,* Automatica **37**, 637 (2001).

[67] G. Looye, *Design of Robust Autopilot Control Laws with Nonlinear Dynamic Inversion,* at–Automatisierungstechnik **49**, 523 (2001).

[68] T. J. Lombaerts, H. O. Huisman, Q. P. Chu, J. A. Mulder, and D. A. Joosten, *Flight Control Reconfiguration based on Online Physical Model Identification and Nonlinear Dynamic Inversion,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-2008-7435)., 2008).

[69] J. Reiner, G. J. Balas, and W. L. Garrard, *Flight Control Design Using Robust Dynamic Inversion and Time-scale Separation,* Automatica **32**, 1493 (1996).

[70] P. R. Smith, *A Simplified Approach to Nonlinear Dynamic Inversion Based Flight Control,* in *proceedings of the AIAA Atmospheric Flight Mechanics Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-98-4461), 1998) pp. 762–770.

[71] 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 **33**, 1732 (2010).

[72] P. R. Smith and A. Berry, *Flight Test Experience of a Nonlinear Dynamic Inversion Control Law on the VAAC Harrier,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-2000-3914), 2000) pp. 132–142.

[73] B. J. Bacon and A. J. Ostroff, *Reconfigurable Flight Control using Nonlinear Dynamic Inversion with a Special Accelerometer Implementation,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit* ((AIAA-2000-4565), 2000).

[74] B. J. Bacon, A. J. Ostroff, and S. M. Joshi, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Tech. Rep. (NASA Langley Research Center, 2000).

[75] B. J. Bacon, A. J. Ostroff, and S. M. Joshi, *Reconfigurable NDI Controller using Inertial Sensor Failure Detection & Isolation, AIAA Guidance, Navigation, and Control Conference and Exhibit*, IEEE Transactions on Aerospace and Electronic Systems **37**, 1373 (2001).

[76] H. K. Khalil, *Nonlinear Systems*, 3rd ed. (Prentice Hall, 2002).

[77] H. B. Chen and S. G. Zhang, *Robust Dynamic Inversion Flight Control Law Design,* in *ISSCAA 2008, 2nd International Symposium on Systems and Control in Aerospace and Astronautics* (2008).

[78] Q. P. Chu, *Spacecraft Attitude Control Systems* (Lecture notes, Delft University of Technology, Faculty of Aerospace Engineering, 2010).

[79] Q. P. Chu, *Advanced Flight Control* (Lecture notes, Delft University of Technology, Faculty of Aerospace Engineering, 2010).

[80] P. Simplício, M. Pavel, E. van Kampen, and Q. P. Chu, *An Acceleration Measurements-based Approach for Helicopter Nonlinear Flight Control using Incremental Nonlinear Dynamic Inversion,* Control Engineering Practice **21**, 1065 (2013).

[81] P. Lu, E.-J. van Kampen, and Q. P. Chu, *Robustness and Tuning of Incremental Backstepping Approach,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2015-1762), 2015).

[82] P. Lu, E.-J. van Kampen, *Active Fault-Tolerant Control System using Incremental Backstepping Approach,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2015-1312), 2015).

[83] P. Lu, *Fault Diagnosis and Fault-Tolerant Control for Aircraft Subjected to Sensor and Actuator Faults* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2016).

[84] E. J. Smeur, Q. P. Chu, and G. C. de Croon, *Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles,* Journal of Guidance, Control and Dynamics **39**, 450 (2016).

[85] E. J. Smeur, G. C. de Croon, and Q. P. Chu, *Gust Disturbance Alleviation with Incremental Nonlinear Dynamic Inversion,* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016).

[86] E.J.J. Smeur, *Incremental Control for Hybrid MAVs* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2018).

[87] C. Vlaar, *Incremental Nonlinear Dynamic Inversion Flight Control* (MSc thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2014).

[88] W. van Ekeren, G. Looye, R. O. Kuchar, Q. P. Chu, and E.-J. van Kampen, *Design, Implementation and Flight-Tests of Incremental Nonlinear Flight Control Methods,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2018-0384), 2018).

[89] F. Grondman, G. Looye, R. O. Kuchar, Q. P. Chu, and E. van Kampen, *Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2018-0385), 2018).

[90] T. Keijzer, G. Looye, Q. P. Chu, and E.-J. van Kampen, *Design and Flight Testing of Incremental Backstepping based Control Laws with Angular Accelerometer Feedback,* in *Proceedings of the AIAA Scitech 2019 Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2019-0129), 2019).

[91] X. Wang, E.-J. van Kampen, and Q. P. Chu, *Gust Load Alleviation and Ride Quality Improvement with Incremental Nonlinear Dynamic Inversion,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2017-0774), 2017).

[92] X. Wang, E.-J. van Kampen, R. De Breuker, and Q. P. Chu, *Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2018-0774), 2018).

[93] Y. Huang, *Incremental Nonlinear Control of Hydraulic Parallel Robots* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2019).

[94] M. A. Henson and D. E. Seborg, eds., *Nonlinear Process Control* (Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997).

[95] P. Acquatella B., E. van Kampen, and Q. P. Chu, *Incremental Backstepping for Robust Nonlinear Flight Control,* in *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation, and Control* (2013).

[96] P. Acquatella B., W. van Ekeren, and Q. P. Chu, *PI(D) Tuning for Flight Control Systems via Incremental Nonlinear Dynamic Inversion,* In: IFAC-PapersOnLine **50**, 8175 (2017).

[97] Wang, X., van Kampen, E., Chu, Q.P., and Lu, P., *Stability Analysis for Incremental Nonlinear Dynamic Inversion Control,* Journal of Guidance, Control, and Dynamics **42**, 1116 (2019).

[98] X. Wang, E.-J. van Kampen, Q. P. Chu, and P. Lu, *Stability Analysis for Incremental Nonlinear Dynamic Inversion Control,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2018-1115), 2018).

[99] X. Wang, E.-J. van Kampen, Q. P. Chu, and P. Lu, *Incremental Sliding-Mode Fault-Tolerant Flight Control,* Journal of Guidance, Control, and Dynamics **42**, 244 (2019).

[100] W. Falkena, E. R. van Oort, and Q. P. Chu, *Towards Certifiable Advanced Flight Control Systems, A Sensor Based Backstepping Approach,* in *AIAA Guidance, Navigation, and Control Conference* (Portland, Oregon, 2011).

[101] W. Falkena, *Investigation of Practical Flight Control Systems for Small Aircraft* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2012).

[102] L.G. Sun, C.C. de Visser, Q.P. Chu, W. Falkena, *Hybrid Sensor-Based Backstepping Control Approach with its Application to Fault-Tolerant Flight Control,* Journal of Guidance, Control, and Dynamics **37**, 59 (2014).

[103] L.G. Sun, C.C. de Visser, Q.P. Chu, J.A. Mulder, *Joint Sensor Based Backstepping For Fault-Tolerant Flight Control,* Journal of Guidance, Control, and Dynamics **38**, 62 (2015).

[104] L.G. Sun, *Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification* (PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2014).

[105] G. Looye, *The New DLR Flight Dynamics Library,* in *Proceedings of the 6th International Modelica Conference* (2008) pp. 193–202.

[106] M. Otter, H. Elmqvist, and F. E. Cellier, *Modeling of Multibody Systems with the Object-Oriented Modeling Language Dymola,* Nonlinear Dynamics **9**, 91 (1996).

[107] M. Otter, H. Elmqvist, and S. E. Mattsson, *The New Modelica MultiBody Library,* in *Proceedings of the 3rd International Modelica Conference* (2003) pp. 311–330.

[108] A. Heckmann, M. Otter, S. Dietz, and J. Lopez, *The DLR Flexible Bodies Library to model Large Motions of Beams and of Flexible Bodies exported from Finite Element Programs,* in *Proceedings of the 5th International Modelica Conference* (2006).

[109] T. Bellmann, *Interactive Simulations and advanced Visualization with Modelica,* in *Proceedings of the 7th International Modelica Conference* (2009) pp. 541–550.

[110] A. Pfeiffer, *Optimization Library for Interactive Multi-Criteria Optimization Tasks,* in *Proceedings of the 9th International Modelica Conference* (2012) pp. 669–680.

[111] M. J. Reiner and J. Bals, *Nonlinear Inverse Models for the Control of Satellites with Flexible Structures,* in *Proceedings of the 10th International Modelica Conference* (2014) pp. 577–587.

[112] J. P. Decker, *Experimental Aerodynamics and Analysis of the Stage Separation of Reusable Launch Vehicles*, Tech. Rep. SP-148 (NASA, 1967).

[113] J. P. Decker and J. Gera, *An Exploratory Study of Parallel-Stage Separation of Reusable Launch Vehicles*, Tech. Rep. TN D-4765 (NASA, 1968).

[114] J. P. Decker and A. W. Wilhite, *Technology and Methodology of Separating Two Similar Size Aerospace Vehicles within the Atmosphere,* in *AIAA 13th Aerospace Sciences Meeting*, 1975-29 (1975).

[115] G. L. Bauer, D. E. Cornick, and R. Stevenson, *Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)*, Tech. Rep. CR-2770 (NASA, 1977).

[116] B. N. Pamadi, T. A. Neirynck, N. J. Hotchko, S. W. I., K. J. Murphy, and P. F. Covell, *Simulation and Analyses of Stage Separation of Two-Stage Reusable Launch Vehicles,* Journal of Spacecraft and Rockets **44**, 66 (2007).

[117] P. V. Tartabini, C. M. Roithmayr, M. D. Toniolo, C. D. Karlgaard, and B. N. Pamadi, *Modeling Multibody Stage Separation Dynamics Using Constraint Force Equation Methodology,* Journal of Spacecraft and Rockets **48**, 573 (2011).

[118] K. J. Murphy, P. G. Buning, B. N. Pamadi, W. I. Scallion, and K. M. Jones, *Status of Stage Separation Tool Development for Next Generation Launch Vehicle Technologies,* in *AIAA Paper 2004-2595* (2004).

[119] B. N. Pamadi, T. A. Neirynck, P. F. Covell, N. J. Hotchko, and D. M. Bose, *Simulation and Analyses of Staging Maneuvers of Next Generation Reusable Launch Vehicles,* in *AIAA Paper 2004-5185* (2004).

[120] B. N. Pamadi, N. J. Hotchko, S. J. A., P. F. Covell, and P. V. Tartabini, *Simulation and Analyses of Multi-Body Separation in Launch Vehicle Staging Environment,* in *AIAA Paper 2006-8033* (2004).

[121] R. Franco M., *Enhanced Dynamics and Control Analysis Package (DCAP),* in *Proceedings of ESA Workshop on Spacecraft Guidance, Navigation and Control* (1992).

[122] R. Franco M., L. Dumontel, S. Portigliotti, and R. Venugopal, *The Dynamics and Control Analysis Package (DCAP) - A Versatile Tool for Satellite Control*, Tech. Rep. 87 (ESA Bulletin, 1996).

[123] S. Portigliotti, M. Dumontela, G. Baldesi, and D. Sciacovelli, *DCAP (Dynamics and Control Analysis Package) - an Effective Tool for Modelling and Simulating of Coupled Controlled Rigid Flexible Structure in Space Environment* (2004).

[124] G. Baldesi, D. Sciacovelli, and A. Thirkettle, *Simulation Tool for Generic Launcher Flight Dynamics-Control Interaction Analysis,* in *Proceedings of the 6th International Symposium on Launcher Technologies: Flight Environment Control for Future and Operational Launchers* (2006).

[125] G. Baldesi and M. Toso, *ESA Launcher Flight Dynamics Simulator used for System and Subsystem Level Analyses,* in *Proceedings of the 11th Intl. Workshop on Simulation & EGSE facilities for Space Programmes (SESP 2010)* (2010).

[126] G. Baldesi and M. Toso, *European Space Agency's Launcher Multibody Dynamics Simulator used for System and Subsystem Level Analyses,* CEAS Space Journal **3**, 27 (2012).

[127] J. L. Junkins and J. D. Turner, *Optimal Spacecraft Rotational Maneuvers* (Elsevier Publishing, New York, 1986).

[128] F. Li and P. M. Bainum, *Numerical Approach for Solving Rigid Spacecraft Minimum Time Attitude Maneuvers,* Journal of Guidance, Control and Dynamics **13**, 38 (1990).

[129] R. M. Byers and S. R. Vadali, *Quasi-closed-form Solution to the Time-optimal Rigid Spacecraft Reorientation Problem,* Journal of Guidance, Control and Dynamics **16**, 453 (1993).

[130] I. M. Ross, P. Sekhavat, A. Fleming, and Q. Gong, *Optimal Feedback Control: Foundations, Examples, and Experimental Results for a New Approach,* Journal of Guidance, Control and Dynamics **31**, 307 (2008).

[131] X. Bai and J. L. Junkins, *New Results for Time-optimal Three-axis Reorientation of a Rigid Spacecraft,* Journal of Guidance, Control and Dynamics **32**, 1071 (2009).

[132] A. Fleming, P. Sekhavat, and I. M. Ross, *Minimum-time Reorientation of a Rigid Body,* Journal of Guidance, Control and Dynamics **33**, 160 (2010).

[133] H. Zhou, D. Wang, B. Wu, and E. K. Poh, *Time-optimal Reorientation for Rigid Satellite with Reaction Wheels,* Intl. J. of Control **85**, 1452 (2012).

[134] T. Lee, M. Leok, and N. H. McClamroch, *Time Optimal Attitude Control for a Rigid Body,* in *Proc. of the American Control Conference (ACC), Seattle, WA* (2008) pp. 5210–5215.

[135] T. Lee, *Geometric Tracking Control of the Attitude Dynamics of a Rigid Body on SO(3),* in *Proc. of the American Control Conference (ACC), San Francisco, CA* (2011) pp. 1200–1205.

[136] K. D. Bilimoria and B. Wie, *Time-optimal Reorientation of a Rigid Axisymmetric Spacecraft,* in *Proc. of the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA* (1991).

[137] K. D. Bilimoria and B. Wie, *Time-optimal Three-axis Reorientation of a Rigid Spacecraft,* Journal of Guidance, Control and Dynamics **16**, 446 (1993).

[138] M. Karpenko, S. Bhatt, N. Bedrossian, and I. M. Ross, *Flight Implementation of Shortest-time Maneuvers for Imaging Satellites,* Journal of Guidance, Control and Dynamics **37**, 1069 (2014).

[139] P. Acquatella B., *Robust Nonlinear Spacecraft Attitude Control: an Incremental Backstepping Approach* (MSc thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2011).

[140] Youcef-Toumi, K. and Osamu Ito, *A Time Delay Controller for Systems with Unknown Dynamics,* Journal of Dynamic Systems, Measurement, and Control **112**, 133 (1990).

[141] Je Hyung Jung and Pyung-Hun Chang and Oh-Seok Kwon, *A New Stability Analysis of Time Delay Control for Input/Output Linearizable Plants,* in *Proceedings of the 2004 American Control Conference. Boston, Massachusetts* (2004).

[142] P. H. Chang and J. H. Jung, *A Systematic Method for Gain Selection of Robust PID Control for Nonlinear Plants of Second-Order Controller Canonical Form,* IEEE Transactions on Control Systems Technology **17**, 473 (2009).

[143] G. P. Falconi, V. Marvakov and F. Holzapfel, *Fault Tolerant Control for a Hexarotor System Using Incremental Backstepping,* in *Proceedings of the IEEE Multi-Conference on Systems and Control (CCA)* (IEEE, 2016).

[144] *User's Manual for TREETOPS, A Control System Simulation for Structures With a Tree Topology*, Tech. Rep. Contract NAS-36287, Marshall Space Flight Center (NASA, 1990).

[145] R. M. Yates, *TREETOPS Structural Dynamics Controls Simulation System Upgrade*, Tech. Rep. Contract NAS8-40194, Marshall Space Flight Center (NASA, 1996).

[146] J. S. Orr, *Space Launch System Flight Control,* in *Aerospace Control and Guidance Systems Committee (ACGSC) Meeting 110* (2012).

[147] F. G. Lemoine, S. C. Kenyon, J. K. Factor, R. G. Trimmer, *et al.*, *The Development of the Joint NASA GSFC and National Imagery and Mapping Agency NIMA Geopotential Model EGM96*, Tech. Rep. TP-1998-206861, NASA Goddard Space Flight Center (NASA, 1998).

[148] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin, *NRLMSISE-00 Empirical Model of the Atmosphere: Statistical Comparisons and Scientific Issues,* Journal of Geophysical Research: Space Physics **107** (2002).

[149] Modelica Association, *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification Version 3.2.* (2010).

[150] Modelica Association, *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification Version 3.3. Revision 1* (2014).

[151] F. O. Eke, *Dynamics of Variable Mass Systems*, Tech. Rep. CR-1998-208246, NASA Ames Research Center (NASA, 1999).

[152] P. Acquatella B. and M. J. Reiner, *Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations,* in *Proceedings of the 10th International Modelica Conference* (2014) pp. 589–598.

[153] B. Pamadi, P. Tartabini, M. Toniolo, C. Roithmayr, C. Karlgaard, and J. A. Samareh, *Application of Constraint Force Equation Methodology for Launch Vehicle Stage Separation,* Journal of Spacecraft and Rockets **50**, 191 (2013).

[154] J. Baumgarte, *Stabilization of Constraints and Integrals of Motion in Dynamical Systems,* Computer Methods in Applied Mechanics and Engineering **1**, 1 (1972).

[155] U. M. Ascher, H. Chin, L. R. Petzold, and S. Reich, *Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifolds,* Journal of Mechanics of Structures and Machines **23**, 135 (1994).

[156] P. V. Tartabini, D. M. Bose, J. D. McMinn, J. G. Martin, and B. K. Strovers, *Hyper-X Stage Separation Trajectory Validation Studies,* in *AIAA Paper 2003-5819* (2003).

[157] M. Källdahl, *Separation Analysis with OpenModelica,* (2007).

[158] A. Kopp, M. Sippel, S. Stappert, N. Darkow, J. Gerstmann, S. Krause, D. Stefaniak, M. Beerhorst, T. Thiele, A. Gülhan, R. Kronen, K. Schnepper, L. E. Briese, and J. Riccius, *Forschung an Systemen und Technologien für wiederverwendbare Raumtransportsysteme im DLR-Projekt AKIRA,* in *Deutscher Luft- und Raumfahrtkongress* (2017).

[159] L. E. Briese, P. Acquatella B., and K. Schnepper, *Multidisciplinary Modeling and Simulation Framework for Reusable Launch Vehicle System Dynamics and Control,* in *7th Intl. Conference on Astrodynamics Tools and Techniques* (2018).

[160] S. Juliana, Q. P. Chu, J. A. Mulder, and T. J. van Baten, *Flight Control of Atmospheric Re-entry Vehicle with Nonlinear Dynamic Inversion,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference* (2006).

[161] P. Acquatella B., *Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies,* in *6th International Conference on Astrodynamics Tools and Techniques, ICATT* (2016).

[162] S. Juliana, Q. P. Chu, J. A. Mulder, and T. J. van Baten, *The Analytical Derivation of Nonlinear Dynamic Inversion Control for Parametric Uncertain Systems,* in *proceedings of the AIAA Guidance, Navigation, and Control Conference* (2005).

[163] E. Mooij, *The Motion of a Vehicle in a Planetary Atmosphere*, Tech. Rep. LR-768 (Delft University of Technology, Faculty of Aerospace Engineering, 1997).

[164] S. Stoltz, C. Raschke, and K. Courtois, *RW-90, a Smart Reaction Wheel – Progress from BIRD to TET-1,* in *Proc. of the 8th IAA Symposium on Small Satellites for Earth Observation, Berlin* (2011).

[165] S. Löw, J. Herman, D. Schulze, and C. Raschke, *Modes and More; Finding the Right Attitude for TET-1,* in *Proc. of the 12th International Conference on Space Operations (SpaceOps)* (2012).

[166] C. Raschke, A. Nicolai, A. Deckert, and S. Stoltz, *Development, Test and Operation of the Attitude Control System of the TET-1 Satellite,* in *Proc. of the 29th International Symposium on Space Technology and Science (ISTS), Nagoya-Aichi* (2013).

[167] C. Raschke, T. Terzibaschian, Z. Yoon, S. Stoltz, A. Deckert, and A. Nicolai, *The Attitude Control System of the TET-1 Satellite – In-orbit Experiences,* in *Proc. of the 9th IAA Symposium on Small Satellites for Earth Observation, Berlin* (2013).

[168] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, *Rigid-body Attitude Control,* IEEE Control Systems **31**, 30 (2011).

[169] F. Goodarzi, D. Lee, and T. Lee, *Geometric Nonlinear PID Control of a Quadrotor UAV on SE(3),* in *Proc. of the 2013 European Control Conference (ECC), Zurich* (2013).

[170] MATLAB, *Release 14b* (The Mathworks, Inc. Natick, Massachusetts, USA. http://www.mathworks.com, 2014).

[171] R. J. Adams and S. S. Banda, *Robust Flight Control Design Using Dynamic Inversion and Structures Singular Value Synthesis,* IEEE Transactions on Control Systems Technology **1**, 80 (1993).

[172] G. J. Balas, W. L. Garrard, and J. Reiner, *Robust Dynamic Inversion Control Laws for Aircraft Control,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-92-4329), 1994).

[173] D. J. Bugajski and D. F. Enns, *Nonlinear Control Law with Application to High Angle-of-attack Flight,* Journal of Guidance, Control, and Dynamics **15**, 761 (1992).

[174] D. F. Enns, D. J. Bugajski, R. C. Hendrick, and G. Stein, *Dynamic Inversion: an Evolving Methodology for Flight Control,* International Journal of Control **59**, 71 (1994).

[175] D. F. Enns, *Robustness of Dynamic Inversion vs. mu-Synthesis: Lateral-Directional Flight Control Example,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-90-3338), 1990).

[176] M. G. Goman and E. N. Kolesnikov, *Robust Nonlinear Dynamic Inversion Method for An Aircraft Motion Control,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-98-4208), 1990).

[177] P. R. Smith, *Functional Control Law Design Using Exact Nonlinear Dynamic Inversion,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-94-3516), 1994).

[178] P. R. Smith, *Translational Motion Control of VSTOL Aircraft Using Nonlinear Dynamic Inversion,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-95-3452), 1995).

[179] S. A. Snell, D. F. Enns, and W. L. Garrard, *Nonlinear Inversion Flight Control for a Supermanoeuvrable Aircraft,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, 1990).

[180] S. A. Snell, *Nonlinear Dynamic-Inversion Flight Control of Supermaneuvrable Aircraft* (PhD thesis, University of Minnesota, Aerospace Engineering and Mechanics Department, 1991).

[181] S. F. Campbell and J. T. Kaneshige, *A Nonlinear Dynamic Inversion L1 Adaptive Controller for a Generic Transport Model,* in *Proceedings of the American Control Conference (ACC)* (2010) pp. 862–867.

[182] C. Schumacher, *Adaptive Flight Control Using Dynamic Inversion and Neural Networks,* in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (American Institute of Aeronautics and Astronautics, Inc. (AIAA-99-4086), 1990).

[183] J. T. Spooner, M. Maggiore, R. Ordóñez, and K. M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, edited by S. Haykin (John Wiley & Sons, 2002).

[184] S. H. Kim, Y. S. Kim, and C. Song, *A Robust Adaptive Nonlinear Control Approach to Missile Autopilot Design,* Control Engineering Practice **33**, 1732 (2004).

[185] A. J. Krener, *A Decomposition Theory for Differentiable Systems,* SIAM Journal on Control and Optimization **15**, 813 (1977).

[186] R. W. Brockett, *Feedback Invariants for Nonlinear Systems,* in *IFAC Congress* (Helsinki, 1978).

[187] W. F. Phillips, C. E. Hailey, and G. A. Gebert, *Review of Attitude Representations Used for Aircraft Kinematics,* Journal of Aircraft **38**, 718 (2001).

[188] P. Tsiotras, *Stabilization and Optimality Results for the Attitude Control Problem,* Journal of Guidance, Control and Dynamics **19**, 772 (1996).

[189] M. D. Shuster, *A Survey of Attitude Representations,* The Journal of the Astronautical Sciences **41**, 439 (1993).

[190] B. Wie, *Space Vehicle Dynamics and Control*, 2nd ed. (AIAA Education Series, 1998).

[191] S. Wu, G. Radice, Y. Gao, and Z. Sun, *Quaternion-based Finite Time Control for Spacecraft Attitude Tracking,* Acta Astronautica **69**, 48 (2011).

[192] B. Wie, D. Bailey, and C. Heiberg, *Rapid Multitarget Acquisition and Pointing Control of Agile Spacecraft,* Journal of Guidance, Control and Dynamics **25**, 96 (2002).

[193] P. Acquatella B., *Fast Slew Maneuvers for the High-Torque-Wheels BIROS Satellite,* In: Transactions of the Japan Society of Aeronautical and Space Sciences **61**, 79 (2018).

[194] H. Schaub, J. L. Junkins, *Analytical Mechanics of Space Systems*, 1st ed. (American Institute of Aeronautics and Astronautics – Technology & Engineering, 2003).

[195] Salvatore Monaco and Dorothée Normand-Cyrot, *Advanced Tools for Nonlinear Sampled–Data Systems,* European Journal of Control **13**, 221 (2007).

[196] Juan I. Yuz, *Sampled-Data Models for Linear and Nonlinear Systems*, Ph.D. thesis, The University of Newcastle, School of Electrical Engineering and Computer Science (2005).

[197] Juan I. Yuz and Graham C. Goodwin, *On Sampled-Data Models for Nonlinear Systems,* IEEE Transactions on Automatic Control **50**, 1477 (2005).

[198] C. Jizheng, Y. Jianping, and F. Qun, *Flight Vehicle Attitude Determination Using the Modified Rodrigues Parameters,* Chinese Journal of Aeronautics **21**, 433 (2008).

[199] Juan I. Yuz and Graham C. Goodwin, *Sampled-Data Models for Linear Nonlinear Systems* (2014).

[200] Y. C. Wang, W. S. Chen, S. X. Zhang, J. W. Zhu, and L. J. Cao, *Command-Filtered Incremental Backstepping Controller for Small Unmanned Aerial Vehicles,* Journal of Guidance, Control and Dynamics **41**, 954 (2018).

[201] P. Acquatella B. and Q. P. Chu, *Agile Spacecraft Attitude Control: an Incremental Nonlinear Dynamic Inversion Approach,* In: IFAC-PapersOnLine **53** (2020).

[202] E. Dumont, O. Bozic, S. May, O. Mierheim, D. Chrupalla, L. Beyland, S. Karl, J. Klevanski, M. Johannsson, V. Clark, M. Stief, D. Keiderling, P. Koch, P. Acquatella B., D. Saile, G. Poppe, T. Traudt, and C. Manfletti, *Ariane 6 PPH Architecture Critical Analysis: Second Iteration Loop*, Tech. Rep. DLR-IB X-TRAS TN-2015/013 (DLR German Aerospace Center, 2015).

[203] J. Wiedemann *et al.*, *VLM-1 - Vehicle Design and Analysis*, Tech. Rep. DLR-IB XTRAS-TN-VLM-20150302 (DLR German Aerospace Center, 2015).

[204] L. E. Briese, A. Klöckner, and M. Reiner, *The DLR Environment Library for Multi-Disciplinary Aerospace Applications,* in *Proceedings of the 12th International Modelica Conference* (2017).

[205] P. van Gils., *Adaptive Incremental Backstepping Flight Control applied to an F-16 Aircraft Model* (MSc thesis, Delft University of Technology, Faculty of Aerospace Engineering, 2015).

[206] Y. C. Wang, W. S. Chen, S. X. Zhang, J. W. Zhu, and L. J. Cao, *Command-Filtered Incremental Backstepping Controller for Small Unmanned Aerial Vehicles,* Journal of Guidance, Control and Dynamics **41**, 954 (2018).

[207] X. Wang and E.-J. van Kampen, *Incremental Backstepping Sliding Mode Fault-Tolerant Flight Control,* in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum* (American Institute of Aeronautics and Astronautics, Inc. (AIAA 2019-0110), 2019).

[208] T. C. Hsia, *On a Simplified Joint Controller Design for Robot Manipulators,* in *Proceedings of the 26th Conference on Decision and Control* (1987).

[209] T. C. Hsia, T. A. Lasky, and Z. Y. Guo, *Robust Independent Robot Joint Control: Design and Experimentation,* in *Proceedings of the 1988 IEEE International Conference on Robotics and Automation* (1988).

[210] T. C. Hsia, *A New Technique for Robust Control of Servo Systems,* IEEE Transactions on Industrial Electronics **36** (1989).

[211] T. C. Hsia and L. S. Gap, *Robot Manipulator Control using Decentralized Linear Time-Invariant Time-Delayed Joint Controllers,* in *Proceedings of the 1988 IEEE International Conference on Robotics and Automation* (1990).

[212] T. C. Hsia, T. A. Lasky, and Z. Y. Guo, *Robust Independent Joint Controller Design for Industrial Robot Manipulators,* IEEE Transactions on Industrial Electronics **38** (1991).

[213] D. Angeli, *A Lyapunov Approach to Incremental Stability Properties,* IEEE Transactions on Automatic Control **47** (2002).

ACKNOWLEDGEMENTS

The completion of this thesis at the TU Delft while being employed at the German Aerospace Center (DLR) in Germany has been one of the biggest challenges in my professional and personal life. I am very grateful for this opportunity and completing this thesis would have never been possible without the support, encouragement, and mentorship of many people that I wish to thank for.

First of all, I wish to thank my academic mentors prof. dr. M. (Max) Mulder, dr. Q.P. (Ping) Chu, and dr. E. (Erik-Jan) van Kampen. I felt very lucky not only for coming to Delft over ten years ago for my MSc. studies but also for being able to keep the connection at the Control & Simulation Section of the Aerospace Engineering Faculty which allowed me the opportunity to pursue this thesis as a guest doctoral candidate. Max, from you I had lots of support, guidance, and encouragement – this was true also during tough times. Many thanks for being so attentive and kind, for giving me the opportunity to continue to be part of the C&S family while being abroad, for the hospitality during my doctoral thesis, and above all for being my promotor. Ping, you were the first to bring me to the world of incremental nonlinear control when I barely knew anything about *nonlinear systems*. You also encouraged me to do research abroad and came up with the idea of completing my thesis whenever possible after my MSc. studies. Without your ideas and our discussions this thesis would not have been possible, so I sincerely thank you for that. I regret the fact of not being able to finish my thesis before your retirement and therefore not having you also as a promotor. Erik-Jan, from you I had much support, feedback, and corrections not only early on during my MSc. as my supervisor but also recently during the final phase of my doctoral thesis. I was really glad and thankful that you joined my thesis as copromotor and for taking the time to review my material. I appreciated a lot that you took the time to attend my thesis last corrections and revisions (in special the last chapter on sampled–data control) during this difficult year while also dealing with coursework and many other students.

During my years at DLR I have been exposed and involved in many projects related to dynamics and control of space systems which has made my work life very exciting and fulfilling. For that reason I am very grateful to Prof. Dr. -Ing. J. Bals, director at the DLR Institute of System Dynamics and Control, for giving me the opportunity to work in the most exciting space topics and for allowing me to combine my research work during my time at DLR into this doctoral thesis. Since my arrival at DLR I've been continuously learning, experimenting, and applying many concepts and ideas into different projects and this has been only possible thanks to the support we get from the institute. I am also indebted and grateful to dr.ir. G.H.N. (Gertjan) Looye who became my first connection TU Delft–DLR and who provided support and mentorship during my time so far at DLR. Thanks to you I was able to stay motivated as I was always encouraged to keep up the

work one step at a time. Our discussions and exchange of ideas, articles, viewpoints, and worries (all mine) all over these years were really helpful to me, so my sincere thanks for that. Moreover, interacting and collaborating with my DLR colleagues has always brought me an element of growth and motivation to succeed. My sincere thanks goes to all my colleagues from our group at the *Space Systems Dynamics* department for the joint work and exchange of ideas and opinions regarding our projects during all these years and I also wish to thank to my colleagues from other departments, namely *Aircraft Systems Dynamics* and *Transport Systems Dynamics*, because of the nice atmosphere in our institute which always lead to fruitful conversations and interesting viewpoints. Many thanks goes to the wonderful administrative staff at DLR: Monika Klauer and Barbara Jaumman, and at TU Delft: Bertine Markus and Nancy van Veen for all their help and smoothness regarding all my proceedings, paperwork, visas, credits, etc. as a foreign employee in Germany and as a guest doctoral candidate at the TU Delft.

To my closer colleagues at the institute, Mihai, Björn, Tobias, and Mehran, goes my gratitude and appreciation for their enormous help during the last years, their feedback, the interesting discussions and exchange of ideas, and for the corrections during the (many) revisions of my papers and my thesis. Special thanks goes to Mehran, my office mate for so many years and from who I got to learn a lot from, not only about robotics but also about non–work related topics like literature, philosophy, stoic mindset, et cetera. Many thanks as well for the countless letters and emails translations and corrections into proper German!

I have many friends that I wish to thank for, but the list is extensive and I don't want to leave anyone behind. Some close friends are here in Munich, but most friends are abroad all around the world. I couldn't do this thesis without the friendly support I got from you and forgive me for not mentioning all of you here, but you all know who you are. Last but not least, my special thanks goes to my beloved family. We have been apart for so many years, so thank you for your support in all aspects, for believing in me, and for offering me your patience and trust in my decisions and sacrifices as you always did.

Paul J. Acquatella B.
Munich, October 2020

CURRICULUM VITÆ

Paul José ACQUATELLA BUSTILLO

25-09-1982 Born in Caracas, Venezuela.

EDUCATION

2020 PhD., Aerospace Engineering
Faculty of Aerospace Engineering, Control & Simulation Section
Delft University of Technology, Delft, The Netherlands
Thesis: Robust Nonlinear Attitude Control of Aerospace Vehicles. An incremental nonlinear control approach
Promotor: prof. dr. ir. M. Mulder
Copromotor: dr. ir. E. van Kampen

2009–2011 MSc., Aerospace Engineering, *cum laude*
Faculty of Aerospace Engineering, Control & Simulation Section
Delft University of Technology, Delft, The Netherlands
Thesis: Robust Nonlinear Spacecraft Attitude Control: an Incremental Backstepping Approach
Supervisors: dr. Q. P. Chu, dr. ir. E. van Kampen, ir. W. Falkena

2005–2007 Master Recherche, Mechanical Engineering
Genie Mecanique Conception (GMC), Laboratoire Ampére
Institut National des Sciences Apliquées de Lyon (INSA Lyon), France

2001–2007 BSc., Mechanical Engineering
Universidad Simón Bolívar (USB), Sartenejas, Venezuela

PROFESSIONAL EXPERIENCE

2012–present *Research Scientist*, DLR Institute of System Dynamics and Control

2010 *Intern*, Airbus D&S (former EADS Astrium GmbH)

2008–2009 *Academic Assistant*, USB Control Lab

2007–2008 *R&D Engineer*, Procter & Gamble Latin America Innovation Center

SELECTED PUBLICATIONS

9. **P. Acquatella B.,** E. van Kampen, Q. P. Chu (2020), *A Sampled-Data Form of Incremental Nonlinear Dynamic Inversion for Spacecraft Attitude Control,* ***to be submitted.***

8. **P. Acquatella B.,** Q. P. Chu (2020), *Agile Spacecraft Attitude Control: an Incremental Nonlinear Dynamic Inversion Approach.* In: IFAC-PapersOnLine, Volume **53**, Issue 2, 2020.

7. **P. Acquatella B.**, L. Briese, K. Schnepper (2018), *Guidance Command Generation and Nonlinear Dynamic Inversion Control for Reusable Launch Vehicles.* In: **Acta Astronautica**, Vol. 174, pp. 334–346, 2020.

6. **P. Acquatella B.** (2018), *Fast Slew Maneuvers for the High-Torque-Wheels BIROS Satellite.* In: **Transactions of the Japan Society of Aeronautical and Space Sciences**, Vol. 61, No. 2, pp. 79–86, 2018.

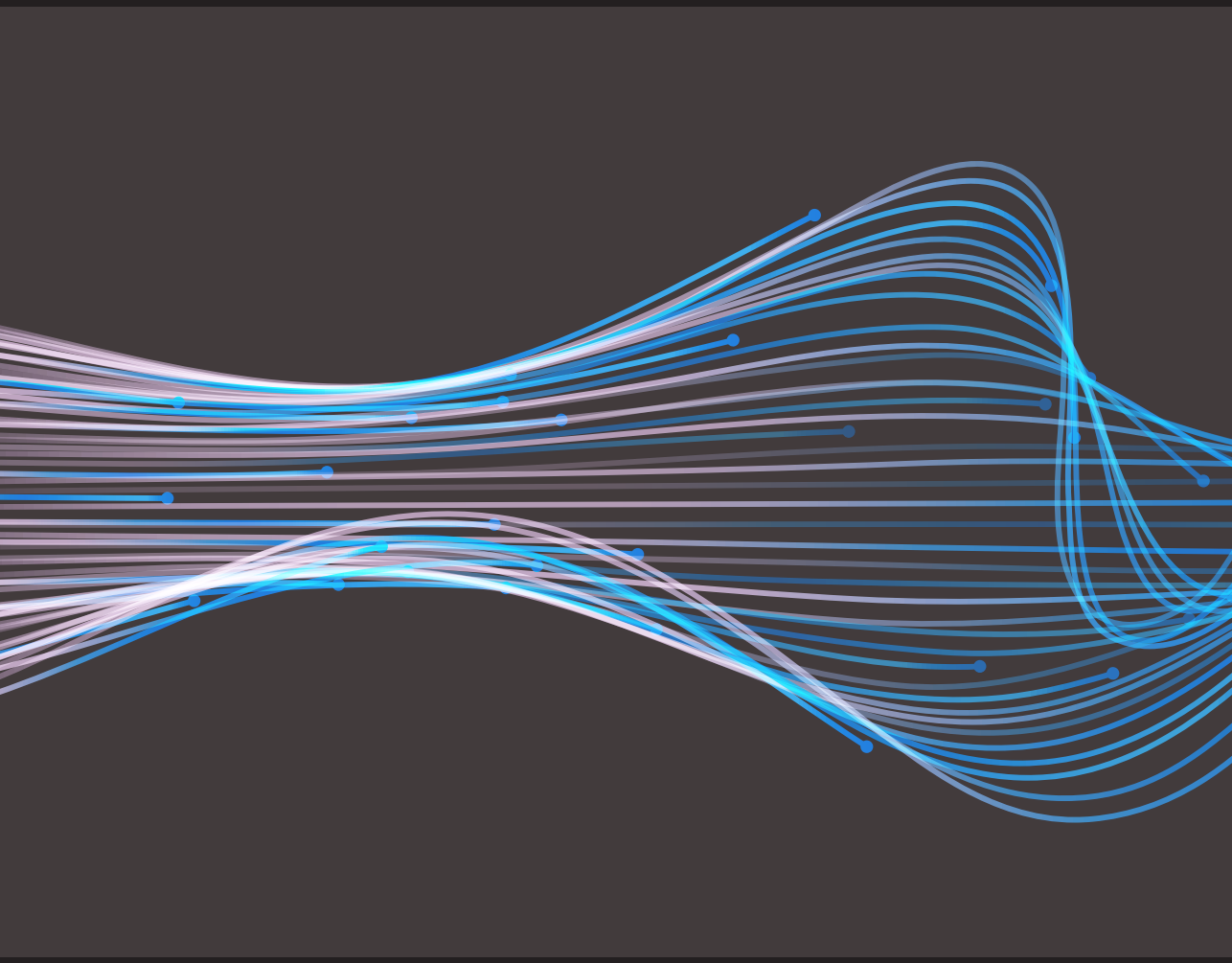
5. **P. Acquatella B.**, W. van Ekeren, Q. P. Chu (2017), *PI(D) Tuning for Flight Control Systems via Incremental Nonlinear Dynamic Inversion.* In: IFAC-PapersOnLine, Volume **50**, Issue 1, pp. 8175–8180, 2017.

4. **P. Acquatella B.** (2016), *Launch Vehicle Multibody Dynamics Modeling Framework for Preliminary Design Studies.* In: ICATT 2016, 6th ESA International Conference on Astrodynamics Tools and Techniques. Darmstadt, Germany, 2016.

3. **P. Acquatella B.**, M. J. Reiner (2014), *Modelica Stage Separation Dynamics Modeling for End-to-End Launch Vehicle Trajectory Simulations.* In: Proceedings of the 10th International Modelica Conference. Lund, Sweden, 2014.

2. **P. Acquatella B.**, E. van Kampen, Q. P. Chu (2013), *Incremental Backstepping for Robust Nonlinear Flight Control.* In: Proceedings of EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control. Delft, The Netherlands, 2013.

1. **P. Acquatella B.**, W. Falkena, E. van Kampen, Q. P. Chu (2012), *Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion.* In: Proceedings of the AIAA Guidance, Navigation and Control Conference. Minneapolis, MN, USA, 2012.



ISBN 978-94-6421-120-7