POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Robust Incremental Sensor-Based Control for Flexible Aircraft

FRÉDÉRICK LALIBERTÉ
Département de génie électrique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie électrique

Mai 2025

© Frédérick Laliberté, 2025.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Robust Incremental Sensor-Based Control for Flexible Aircraft

présentée par **Frédérick LALIBERTÉ**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Guchuan ZHU, président
David SAUSSIÉ, membre et directeur de recherche
Sofiane ACHICHE, membre
Georges GHAZI, membre externe

DEDICATION

To Catherine...

ACKNOWLEDGEMENTS

I would first like to thank my research director, David Saussié, whose invaluable advice, support, and encouragement greatly contributed to the completion of this thesis. His passion, expertise in robust control of aerospace systems, and strong work ethic made him an ideal role model and mentor. He continuously found ways to foster in me similar skills and a like-minded approach. I am especially grateful for his trust in my ability to teach others and to explore the uncertainties of research long enough to uncover meaningful answers.

I thank the Control Laws team at Bombardier Aerospace for their support during our partnership, particularly Aymeric Kron, Matthew Olsthoorn, Sylvain Thérien, Mohammad Riazi, Steve Hébert, and Martin Krenn. I also acknowledge the MITACS organization for facilitating such collaborations through its scholarship programs. The experience gained in implementing aircraft models and control laws was valuable, and the challenges encountered provided new perspectives that contributed to the success of this research.

I also thank the jury members, Prof. Guchuan Zhu, Prof. Sofiane Achiche, and Prof. Georges Ghazi, for their time and thoughtful evaluation.

I would like to thank my friends Jonathan, Jordan, and Joël, as well as my family, for their unwavering support during difficult times and for always sharing their joy. Times may change, but it is reassuring to see that true friendships endure.

Finally, my deepest thanks go to my partner, Catherine, for being herself and for standing by my side all these years. I dedicate this thesis to her.

RÉSUMÉ

La recherche d'une meilleure efficacité énergétique incite l'industrie aéronautique à utiliser des matériaux plus légers, mais plus flexibles. Cela accentue les interactions dynamiques entre les déformations élastiques des surfaces portantes et les forces aérodynamiques qu'elles génèrent, entraînant l'apparition de phénomènes aéroservoélastiques (ASE) indésirables qui perturbent les commandes de vol.

L'approche courante repose sur le séquencement de gains avec des filtres coupe-bande, ce qui nécessite des modèles aérodynamiques précis et des réglages fins, engendrant des coûts croissants pour le développement des lois de commande. Dans ce contexte, cette thèse vise à établir un processus de conception des systèmes de commande de vol pour avions flexibles, en proposant l'inversion incrémentale de la dynamique non linéaire (*Incremental Nonlinear Dynamic Inversion*, INDI) comme alternative au séquencement de gains, et en exploitant la synthèse $\mathcal{H}_\infty$ structurée pour optimiser de manière robuste et automatique les nombreux réglages requis.

La commande INDI est d'abord analysée en profondeur afin d'en raffiner ses hypothèses simplificatrices en tests quantitatifs d'applicabilité, facilitant ainsi son utilisation en pratique. Cette analyse révèle que plusieurs hypothèses ne sont que suffisantes, mais non nécessaires. En particulier, l'architecture classique de INDI peut être reformulée sous une forme plus complexe intégrant un prédicteur de Smith. À la lumière de ces investigations, des tests quantitatifs sont proposés, accompagnés de suggestions de techniques de mitigation en cas d'échec. Ces contributions renforcent la maturité de la technologie étudiée et étendent son applicabilité à une plus grande classe de systèmes.

À la suite des améliorations apportées à la commande INDI, le processus de conception proposé est appliqué à un avion rigide. Les tests quantitatifs permettent d'évaluer l'applicabilité de la commande INDI, tandis que les techniques de mitigation suggérées sont mises en œuvre pour compenser les actionneurs réalistes (non identiques et à phase non minimale) de l'appareil. Leur performance est ensuite démontrée par comparaison avec une version non modifiée de la commande INDI. La synthèse $\mathcal{H}_\infty$ structurée est employée afin d'obtenir un réglage robuste du correcteur linéaire unique de la commande INDI sur l'ensemble de l'enveloppe de vol de l'aéronef, atténuant ainsi la perte de performance induite par les mesures de mitigation. Le respect des exigences est évalué à l'aide de campagnes d'analyse de type pire-cas (*Worst-Case analysis*) et de simulations Monte-Carlo, mettant en lumière les avantages de chaque approche et la manière dont les outils d'analyse pire-cas peuvent renforcer les efforts

de vérification.

L'applicabilité du processus de conception proposé, y compris celle de la commande INDI, est ensuite étendue aux avions flexibles grâce à l'intégration des filtres ASE habituels. La synthèse $\mathcal{H}_\infty$ structurée est employée pour régler le contrôleur linéaire sur une approximation rigide de l'avion flexible, tout en prenant en compte la présence des filtres ASE afin d'en atténuer l'impact négatif sur la performance. Enfin, une campagne Monte-Carlo limitée est réalisée afin d'évaluer la viabilité de cette approche pour la commande d'avions flexibles. Les résultats obtenus montrent une performance satisfaisante sur l'ensemble de l'enveloppe de vol, confirmant ainsi la faisabilité du processus de conception proposé.

Ensemble, ces contributions et ces démonstrations constituent une approche alternative à la conception des commandes, éliminant le besoin de séquencer ou d'ajuster manuellement une multitude de gains, tout en conservant les composants matériels existants. De plus, la transition du processus traditionnel vers l'approche proposée peut s'effectuer de manière progressive. La synthèse $\mathcal{H}_\infty$ structurée étant compatible avec le séquencement de gains, elle peut être intégrée progressivement avant l'adoption de INDI.

ABSTRACT

In its quest to improve fuel efficiency, the aviation industry has increasingly turned to lightweight and flexible materials. This has led to greater dynamic interactions between the elastic deformations of the lifting surfaces and the aerodynamic forces they generate, giving rise to undesirable aeroservoelastic (ASE) phenomena that interfere with the aircraft's controls. Given the common practice of using a gain-scheduled controller with specially designed notch filters, accurate aerodynamic models and precise tuning are required, leading to ever-increasing costs for control design development. In this context, this thesis aims to formulate an alternative control design process for flexible aircraft, centered around the Incremental Nonlinear Dynamic Inversion (INDI) approach as a replacement for gain scheduling, and the use of Structured $\mathcal{H}_\infty$ Synthesis instead of manual fine-tuning.

First, the INDI control scheme itself is investigated to refine its qualitative assumptions into quantitative applicability tests, which would significantly facilitate its practical use. This analysis reveals that several assumptions are sufficient but not necessary, as the classic INDI architecture can be framed as a simplification of a more complex architecture involving a Smith Predictor. Through these investigations, quantitative tests are proposed along with suggested mitigation techniques in case of failure, contributing to the improvement of the technology's maturity and extending its applicability to a broader class of systems.

Following the improvements in INDI control, the proposed design process is applied to a rigid aircraft. Quantitative tests are conducted to assess INDI applicability, and proposed mitigation techniques are implemented to compensate for the aircraft's realistic (non-identical and non-minimum phase) actuators. Their performance is demonstrated through comparison with an unmodified version. The Structured $\mathcal{H}_\infty$ Synthesis is employed to robustly tune the unique linear controller stage of the INDI control law across the aircraft's entire flight envelope, thereby compensating for performance losses caused by the applied mitigations. Requirement fulfillment is evaluated through worst-case analysis and Monte Carlo campaigns, highlighting the benefits of both approaches and demonstrating how verification efforts could be enhanced by leveraging worst-case analysis tools.

The applicability of the proposed design process, including that of INDI control itself, is then extended to flexible aircraft through the use of commonplace ASE filters. The Structured $\mathcal{H}_\infty$ Synthesis is employed to tune the linear controller for a rigid approximation of the flexible aircraft while accounting for the presence of ASE filters to mitigate their adverse effects on performance. A limited Monte Carlo campaign is conducted to evaluate the viability of this

approach to flexible aircraft control, demonstrating reasonable performance across the flight envelope and confirming the feasibility of the proposed design process.

Taken together, these contributions and demonstrations establish an alternative control design process that eliminates the need for gain-scheduling architectures and tedious manual fine-tuning of various controllers, without requiring costly hardware modifications or upgrades. Furthermore, the transition from the traditional process to the proposed one can be made incrementally, as Structured $\mathcal{H}_\infty$ Synthesis is compatible with gain-scheduling and can be adopted before fully integrating INDI.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE OF CONTENTS ix

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF SYMBOLS AND ABBREVIATIONS xvii

LIST OF APPENDICES xviii

CHAPTER 1 INTRODUCTION 1
- 1.1 Context 1
- 1.2 Axes of research 2
  - 1.2.1 Incremental sensor-based control 3
  - 1.2.2 Robust control 4
  - 1.2.3 Control for flexible aircraft 4
- 1.3 Thesis outline 6

CHAPTER 2 LITERATURE REVIEW AND OBJECTIVES 8
- 2.1 Literature review 8
  - 2.1.1 Preliminary topics 8
  - 2.1.2 Modelling of flexible aircraft 12
  - 2.1.3 Control 20
  - 2.1.4 Verification 28
- 2.2 Objectives, methodology and contribution 30
  - 2.2.1 Objective 1: Develop a control law based on feedback-linearization that is compatible with current control design methods 31

2.2.2 Objective 2: Propose an algorithm-based controller tuning approach that is compatible with both gain-scheduling and the proposed control law based on feedback-linearization . . . 32
2.2.3 Objective 3: Demonstrate that the proposed control law and tuning method are compatible with flexible aircraft . . . 33
2.2.4 List of publications . . . 34

CHAPTER 3 ARTICLE 1: RELAXING THE APPLICABILITY CONDITIONS FOR INCREMENTAL NONLINEAR DYNAMICS INVERSION . . . 35
3.1 Introduction . . . 35
3.2 Classic Incremental Nonlinear Dynamic Inversion . . . 37
3.2.1 Linearization by Taylor series expansion . . . 38
3.2.2 INDI Control Law . . . 38
3.3 Alternative solutions to the equations of motion . . . 40
3.3.1 Open-loop . . . 40
3.3.2 Closed-loop . . . 43
3.4 Characterization of the INDI Feedback-Linearization Architecture . . . 45
3.4.1 Ideal closed-loop model . . . 46
3.4.2 Continuous delay-free closed-loop model . . . 47
3.4.3 Delayed continuous closed-loop model . . . 47
3.4.4 Discrete closed-loop model . . . 50
3.4.5 Open-loop Linearization Errors . . . 51
3.5 Application example: Rigid Aircraft . . . 52
3.5.1 Model presentation . . . 52
3.5.2 Actuator . . . 54
3.5.3 Plant . . . 56
3.5.4 Delay . . . 56
3.5.5 Sampling . . . 57
3.5.6 Mitigation - Non-identical actuators . . . 60
3.6 Discussion: Applicability conditions . . . 64
3.7 Conclusion . . . 66

CHAPTER 4 ARTICLE 2: ROBUST TUNING OF THE MODIFIED INCREMENTAL NONLINEAR DYNAMICS INVERSION FOR A RIGID AIRCRAFT . . . 67
4.1 Introduction . . . 68
4.2 Model . . . 70
4.2.1 Aircraft rigid model . . . 70

4.2.2 Hardware models 72
4.3 Control 73
4.3.1 Requirements 74
4.3.2 INDI control law 75
4.3.3 INDI applicability and mitigation - Feedback-Linearization stage . . . 78
4.3.4 Tunable controller architecture - Linear-Controller stage 81
4.3.5 Robust gain tuning - Structured $\mathcal{H}_\infty$ Synthesis 83
4.4 Verification 86
4.4.1 Worst-Case Analysis Toolbox 87
4.4.2 Nonlinear requirement assessment 89
4.4.3 Results 90
4.5 Conclusion 96

CHAPTER 5 ARTICLE 3: ROBUST CONTROL FOR FLEXIBLE AIRCRAFT USING INCREMENTAL NONLINEAR DYNAMICS INVERSION WITH NOTCH FILTERS 97
5.1 Introduction 99
5.2 Technical Preliminaries 100
5.2.1 Incremental Nonlinear Dynamic Inversion 100
5.2.2 Aeroservoelastic filtering 101
5.2.3 Structured $\mathcal{H}_\infty$ Synthesis 102
5.3 Model 104
5.3.1 Flexible aircraft 104
5.3.2 Unsteady aerodynamics - Modified strip theory 107
5.3.3 Hardware models 109
5.3.4 Flight Envelope 109
5.4 Control 110
5.4.1 Feedback-Linearization stage - Angular acceleration loop 111
5.4.2 Linear-controller stage - Angular rate loop 113
5.4.3 Robust tuning - Structured $\mathcal{H}_\infty$ Synthesis 114
5.5 Verification 117
5.5.1 Nonlinear responses 118
5.5.2 Monte Carlo simulation campaigns 119
5.6 Conclusion 120

CHAPTER 6 GENERAL DISCUSSION 121
6.1 Summary 121

6.2 Synthesis 123

CHAPTER 7 CONCLUSION 127
7.1 Summary of Works 127
7.2 Limitations 127
7.3 Future Research 128

REFERENCES 129

APPENDICES 140

LIST OF TABLES

Table 2.1 Parameters of the Benchmark Active Control Technology (BACT) diagram of Fig.2.3 14
Table 3.1 Aerodynamic coefficient values 54
Table 3.2 Actuator systems considered as candidate 55
Table 3.3 Tests for INDI applicability and proposed mitigation 65
Table 4.2 Constant coefficient values 71
Table 4.3 Flight conditions 72
Table 4.4 Tests for INDI applicability and proposed mitigation 79
Table 4.5 Tests results for INDI applicability 82
Table 4.6 Normal flight operation limits and configurations yielding maximal Lipschitz constants 82
Table 4.7 Synthesis results - Costs 87
Table 4.8 Synthesis results - Gain values 87
Table 4.9 Monte Carlo Analysis results 90
Table 4.10 Worst Case Analysis results 91
Table 5.2 Flight conditions considered for design 110
Table 5.3 Requirements - ASE filter tuning 115
Table 5.4 Synthesis results - ASE filter tuning 115
Table 5.5 Requirements - LC gains tuning 116
Table 5.6 Synthesis results - LC Gains 117

LIST OF FIGURES

Figure 1.1 Total Scheduled Traffic, 2012-2021 1
Figure 2.1 Wing section of the BACT (from [1]) 13
Figure 2.2 Pitch And Plunge Apparatus of the BACT (from [1]) 13
Figure 2.3 Diagram of the BACT (from [1]) 13
Figure 2.4 Discretization of a complex surface (from [2]) using the Doublet Lattice Method 16
Figure 2.5 Concept of aeroservoelastic (ASE) compensation through notch filtering (from [3]) 21
Figure 2.6 Closed-loop system (standard framework) 22
Figure 2.7 Architecture of a gain-scheduled controller 24
Figure 2.8 Architecture of the Nonlinear Dynamic Inversion (NDI) control law . 25
Figure 2.9 Architecture of the Incremental Nonlinear Dynamic Inversion (INDI) control laws 26
Figure 3.1 INDI Control architecture 40
Figure 3.2 Idealized Feedback-Linearisation stage: a) Unsimplified and b) simplified actuator loop 46
Figure 3.3 Continuous delay-free closed-loop 47
Figure 3.4 Continuous closed-loop 48
Figure 3.5 Simplified continuous closed-loop using a) $\mathbf{K}(s)$, and b) $\mathbf{H}_{act}(s)$ . . . 49
Figure 3.6 Discrete closed-loop 51
Figure 3.7 Comparison of ideal closed-loop time responses for various actuators . 55
Figure 3.8 Comparison of delay-free closed-loop frequency responses for various actuators 56
Figure 3.9 Comparison of delayed ($0.05s$) closed-loop frequency responses for various actuators 57
Figure 3.10 Comparison of delayed closed-loop frequency responses for various (synchronized) delays (using actuator # 1) 58
Figure 3.11 Comparison of idealized feedback-linearization closed-loop (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) - Frequency responses 59
Figure 3.12 Comparison of idealized feedback-linearization closed-loop (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) - Time responses 59

Figure 3.13 Simulations comparing the true nonlinear time response with its measured signal, linearized model approximation, and discrete transfer function approximation ($T_s = 0.05s$ chosen to highlight the intersample ripple effect) 60
Figure 3.14 Simulations comparing the true nonlinear time response with its measured signal, linearized model approximation, and discrete transfer function approximation ($T_s = 0.002s$ chosen to suppress the intersample ripple effect) 61
Figure 3.15 Modified INDI (not implementable) 62
Figure 3.16 Modified INDI (implementable) 62
Figure 3.17 Unmodified response (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) 63
Figure 3.18 Modified INDI (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) 63
Figure 4.1 INDI Control Law 76
Figure 4.2 Modified INDI Feedback-Linearization Stage 81
Figure 4.3 Considered INDI Linear-Controller Stage ($\mathbf{C}(z)$) 83
Figure 4.4 Balanced disk margin assessment diagram 85
Figure 4.5 Linear Controller Stage response: a) Classic and b) Modified architectures 87
Figure 4.6 Nonlinear tracking assessment - Cost values: 91
Figure 4.7 Nonlinear stability margin assessment - Cost values: 92
Figure 4.8 Nonlinear tracking assessment - Parameter space: 92
Figure 4.9 Nonlinear stability margin assessment - Parameter space: 93
Figure 4.10 Nonlinear tracking assessment - Flight Envelope: 94
Figure 4.11 Nonlinear stability margin assessment - Flight Envelope: 95
Figure 5.1 Standard framework 103
Figure 5.2 Reference frames [4] 104
Figure 5.3 Right wing section 107
Figure 5.4 Flight envelope 110
Figure 5.5 INDI Control Law 111
Figure 5.6 Frequency responses of the unfiltered closed-loop FL stage (flexible aircraft linear models) 113
Figure 5.7 Balanced disk margin assessment diagram 114
Figure 5.8 Filtered closed-loop FL stage (flexible aircraft linear models): 116
Figure 5.9 Flexible aircraft linear responses at design flight conditions: 118
Figure 5.10 Flexible aircraft nonlinear time responses at design flight conditions . 118
Figure 5.11 Monte Carlo campaigns result: a) Tracking, and b) Stability Margins 119

Figure 6.1 Diagrams of the gradual transition from the common practice to the proposed solution to control design process 125

LIST OF SYMBOLS AND ABBREVIATIONS

ASE	Aeroservoelastic
NDI	Nonlinear Dynamic Inversion
INDI	Incremental Nonlinear Dynamic Inversion
ICAO	International Civil Aviation Organization
SAS	Stability Augmentation System
CAS	Control Augmentation System
LCO	Limit-Cycle Oscillation
NASA	National Aeronautics and Space Administration
BACT	Benchmark for Active Control Technology
PDE	Partial Derivative Equation
VLM	Vortex Lattice Method
DLM	Doublet Lattice Method
FEM	Finite-Element Model
LTI	Linear Time-Invariant
LTV	Linear Time-Variant
PID	Proportional-Integral-Derivative
LQR	Linear-Quadratic Regulator
LQG	Linear-Quadratic Gaussian
LFT	Linear Fractional Transformation
UAV	Unmanned Aerial Vehicle
ZOH	Zero-Order Hold
ESA	European Space Agency
PCH	Pseudo-Control Hedging
SISO	Single Input Single Output
MIMO	Multi-Input Multi-Output
WCAT	Worst-Case Analysis Toolbox
DE	Differential Evolution
FFT	Fast-Fourier Transform

LIST OF APPENDICES

Appendix A User guide: Structured $\mathcal{H}_\infty$ Synthesis 140
Appendix B Applicability tests and mitigations for INDI 156

CHAPTER 1 INTRODUCTION

1.1 Context

In its 2021 annual report, the International Civil Aviation Organization (ICAO) highlights that while the industry has suffered from the consequences of COVID-19 pandemic restrictions, air travel demand is recovering and continues its upward trend of the past decades. Figure 1.1 presents the yearly trend of scheduled traffic in both passenger-kilometres and freight tonne-kilometres. In the face of the financial pressure caused by these economic challenges, the aviation industry as a whole is expected to seek cost reductions like never before.

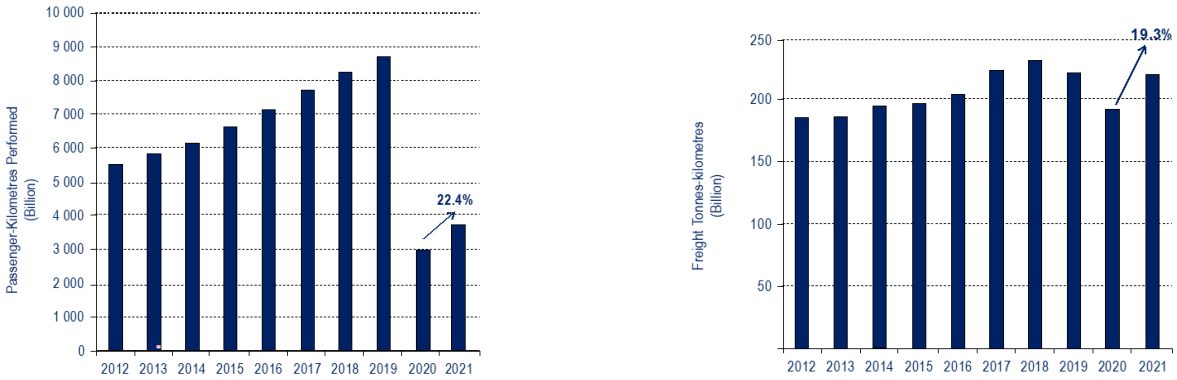

Figure 1.1 Total Scheduled Traffic, 2012-2021

Left: Passenger-Kilometres Performed; Right: Freight Tonnes-Kilometres
Source: ICAO's Annual Report 2021

One area where planemakers have contributed to cost reduction is improving the fuel efficiency of their aircraft, as fuel consumption represents a major operating cost. In their pursuit of greater fuel efficiency, manufacturers have modified the materials used in aircraft structures, resulting in lighter designs. Unfortunately, these lighter materials are also less rigid, allowing the structure to elastically deform under stress, which can disrupt the aircraft's control performance. As a result, planemakers must conduct extensive testing to update and refine their models to account for dynamic interactions between flexible structures and the aerodynamics acting on the aircraft, thereby increasing development costs.

Traditional control law design, which develops the Stability Augmentation System (SAS) and Control Augmentation System (CAS) that assist human pilots and track commands

from navigational autopilots, is based on a rigid aircraft simplification that becomes invalid as the natural harmonic frequency of the aircraft structure begins to overlap with the control system's bandwidth. While specially designed filters have provided sufficient mitigation in the past, the frequent model updates resulting from extensive testing have significantly increased the workload of control designers, who must update the controller gains each time. This challenge is further compounded by the use of a gain-scheduling control architecture, which requires a large set of gains tuned for various flight conditions, a smooth scheduling function to interpolate gains as flight conditions change, and extensive performance verification through a Monte Carlo campaign, consisting of thousands of randomized simulation trials.

In this context, the present thesis aims to propose an alternative approach to control law design for modern aircraft, reducing development costs by replacing the gain-scheduling method and introducing greater automation in the control design process.

1.2 Axes of research

Before considering any improvements to the industry's traditional approach to control system design, it is essential to examine why it has remained dominant. Gain-scheduling is a nonlinear control method constructed from a collection of linear controllers, allowing the use of well-established tools developed for linear control design. As a result, it is simple, effective, and versatile. However, tuning a gain-scheduled controller can be tedious due to the large number of linear models that must be considered and the fine-tuning required to ensure smooth interpolation of gains by the scheduling function. This challenge is further exacerbated for flexible aircraft, as the models are subject to frequent updates. These considerations motivate the research activities carried out in this thesis, which are structured along three main research axes, as explained below.

While various nonlinear control approaches have been explored in the literature, one significant obstacle to their adoption by the industry is the need for authorities to assess their stability and performance in a deterministic manner to grant certifications. This highlights the advantages of leveraging linear methods during control design, as their simplicity and ease of implementation also make them easier to certify. As such, the most promising alternative to gain-scheduling would be another nonlinear method that builds upon a linear controller architecture, such as feedback-linearization control, like Nonlinear Dynamic Inversion (NDI). Of particular note is the novel sensor-based variant, called Incremental Nonlinear Dynamic Inversion (INDI), which is claimed to be robust to modelling errors and uses a unique linear controller for the entire flight envelope of the aircraft, drastically reducing the amount of tuning needed.

Concerning the introduction of greater automation in the tuning process, [5] has already demonstrated the versatility and effectiveness of the structured $\mathcal{H}_\infty$ synthesis when applied to a gain-scheduled controller for flexible aircraft. While the advantages of using a robust tuning approach are numerous, they become less convincing when applied to an already robust controller, as INDI is claimed to be. This apparent excess of caution explains the lack of application examples in the literature concerning the combination of robust tuning and INDI, but it might prove appropriate given that [6] casts doubt on how robust INDI proves to be in practice (especially for flexible aircraft with problematic natural harmonic frequencies causing significant perturbations).

In addition to these two axes of research, control for flexible aircraft also requires special attention. Not only does this axis include the modelling of aeroservoelastic (ASE) phenomena inherent to flexible aircraft, but also novel nonlinear control methods that leverage advances in sensor and actuator designs, in addition to improvements in the traditional reliance on notch filtering as a way to mask these effects from the perspective of the linear controller designed for the rigid-body approximation of the aircraft's dynamics. Unfortunately, these novel designs remain at the experimental stage, as the certification process proves prohibitive for the industry to adopt them. As such, any new controller design proposed should provide support for future changes as a way to "future-proof" the suggested alternative to the traditional control design process.

1.2.1 Incremental sensor-based control

A recently developed family of control techniques, incremental sensor-based control, offers a reduced dependency on plant models and thus presents good potential for reducing developmental costs related to flight tests intended to collect aerodynamic data. This is because this novel approach to control trades off the need for accurate plant models in favor of higher performance requirements on the hardware, which has been made viable thanks to recent advances in computing and sensor designs. Of particular note is INDI, which applies an incremental, sensor-based approach to the feedback-linearization method of nonlinear control and yields a more robust closed-loop that behaves as an ideal linear model, which is then trivial to control with a single unique controller. This is in stark contrast to the gain-scheduling approach, making INDI control a potential alternative. Furthermore, the INDI control law is generic, as it is the exact same regardless of the plant. All particularities of the plant are captured within the Control Effectiveness Matrix, which is far simpler to characterize than an aerodynamics model and could further reduce development costs for new aerospace vehicles.

Unfortunately, the qualitative nature of the simplifying assumptions underlying the INDI control law makes it difficult in practice to assess whether the control law will actually stabilize the system with a given hardware setup; the practitioner is expected to modify the hardware upon failure of the control law, until it finally behaves as intended. This trial-and-error method of assessing the applicability of the control technique is sufficient in simulation settings, where hardware parameters can be modified with a simple click, but would lead to wasteful inefficiencies if attempted in the industry and potentially result in budget issues, as the advanced hardware required to fulfill these qualitative assumptions becomes prohibitively costly for flexible aircraft. As such, more research is required to help this promising technology mature, such as identifying quantifiable criteria of applicability that would allow an *a priori* assessment before any implementation has occurred.

1.2.2 Robust control

Advances in robust control have led to the development of the structured $\mathcal{H}_\infty$ synthesis, which offers an automated tuning method for fixed-architecture controllers. It has been shown to support multiple models, multiple optimization goals, and even self-scheduling tuning during a single optimization run. While this promising tuning method is versatile, it remains a linear control method, as it relies on linear models. Fortunately, it can nonetheless be applied to nonlinear systems through the use of a nonlinear control method that leverages linear control methods, such as Gain Scheduling or feedback linearization. Furthermore, since the main purpose of robust control is to account for various uncertainties and enforce tuning goals even in worst-case scenarios, it has the potential to mitigate some of the performance loss caused by ASE perturbations on flexible aircraft or non-ideal hardware.

Additionally, while the INDI control law has been described as "robust," its robustness is limited to plant model uncertainties and is conditional on its qualitative assumption of near-infinite bandwidth, which is unrealistic in a practical context. As such, its embedded linear controller will be robustly tuned to enhance its robustness against other types of uncertainties, such as those inherent to flexible aircraft.

1.2.3 Control for flexible aircraft

In order to design a controller for a given system, one must first model the system. Research into ASE phenomena has led to the proposal of various approaches to modelling a flexible aircraft, each with its share of advantages and drawbacks. The most popular method appears to be the augmented approach, where the ASE dynamics are modeled as a perturbation on the rigid-body dynamics of the aircraft. This method allows the use of various Finite-Element

Models (FEMs) to accurately model the structural and aerodynamic components of the ASE dynamics, which can then be used to identify problematic harmonic frequencies that may interact with the controls in an undesirable manner.

As such, a popular control technique for flexible aircraft consists of using notch filters to remove these problematic frequencies from measurement signals in order to prevent the controller from stimulating the vibration modes of the structure. However, they tend to reduce performance as the problematic frequencies vary with flight conditions while the filters remain fixed. Furthermore, the increased flexibility of modern aircraft is further encroaching on control bandwidth, which leads to an increasing number of problematic frequencies to filter out and thus additional performance loss due to the added filters required. These issues warrant the consideration of alternatives to flexible aircraft modelling and control in order to alleviate the workload induced by frequent updates to the model.

One drawback of the augmented approach to modelling is the reliance on accurate aerodynamic data, which needs to be acquired through expensive flight testing that cannot be performed before the first prototype is built. As such, analytic approximations are used to obtain a first estimate of the expected aerodynamic properties of the aircraft in order to provide a controller for the first flight test. While this first iteration fulfills its role, it is worth noting that another approach to flexible aircraft modelling, called the integrated approach, appears better suited to the analytic nature of this preliminary model and remains compatible with FEMs usually produced during the augmented approach. Unfortunately, this approach is also more computationally intensive than the augmented approach, so its usefulness would be limited to early iterations of the design or for research purposes when data is insufficient for the augmented approach.

Furthermore, research into distributed actuation (and sensing) has shown promise in improving the achievable performance for flexible aircraft, at the cost of increased hardware and control law complexity. Fortunately, INDI has been shown to leverage such distributed sensor and actuator suites by incorporating flexible states into its feedback-linearization loop, with the drawback of significantly increasing the required bandwidth for the technique's underlying assumptions to hold. As this prohibitively costly hardware architecture could realistically become affordable in the future as the associated technologies mature, any planemaker already familiar with the use of INDI control on flexible aircraft will potentially have an advantage over their competition.

1.3 Thesis outline

The remainder of this thesis is divided into six chapters, in addition to this introduction, and complemented with two appendices.

Chapter 2 is dedicated to the literature review of relevant topics related to the control of flexible aircraft and the presentation of research objectives. The ASE phenomena arising from the interaction between structure and aerodynamics are described first, followed by a presentation of modelling approaches for flexible aircraft that aim to capture these ASE effects in order to allow their regulation during control design. The control section introduces relevant linear and nonlinear control methods, covering topics such as ASE filters, robust control, gain scheduling, and feedback linearization. A short overview of performance verification methods, namely Monte Carlo campaigns and Worst-Case analysis, is then presented. This chapter concludes with the objectives considered for this thesis, the methodology employed to achieve them, and the resulting contributions.

Following this, Chapters 3 to 5 are reproductions of articles submitted to scientific journals.

Chapter 3 is centered on the topic of clarifying and relaxing the conditions of applicability for INDI to make it suitable for industrial use. Assumptions are pushed to their ideal limit to create the ideal baseline, and then the relevant parameters of each assumption are swept until instability. This results in several non-ideal models that are compared to the baseline in order to identify the effects of each assumption, and how they interact with one another, resulting in quantifiable thresholds and equivalent alternative architectures that enable new potential modifications not relying on the same assumptions. Applicability tests and mitigation techniques are proposed based on these investigations into the simplifying assumptions required by INDI.

Chapter 4 presents a demonstration of the proposed INDI alternative design approach on a realistic rigid aircraft while using the suggested robust tuning method. The tests and mitigation techniques from Chapter 3 are applied, the controller is robustly tuned over the entire flight envelope of the aircraft using the structured $\mathcal{H}_\infty$ synthesis, and a worst-case analysis approach is used to enhance performance verification. This application example serves as a proof of concept for combining robust control with the proposed modified INDI controller as an effective approach to the control design of rigid aircraft.

Chapter 5 tackles the control of a flexible aircraft using notch filters and the proposed robust INDI scheme. A flexible aircraft is modeled using an integrated approach, notch filters are designed to prevent ASE effects from disturbing the control loop, and the linear controller design process for a rigid aircraft presented in Chapter 4 is used with the added presence of

the ASE filters. This chapter thus provides a proof of concept that the control design process presented in Chapter 4, with the addition of common ASE filters, is compatible with flexible aircraft.

Then, Chapter 6 offers a general discussion on how the works presented in the previous chapters collectively provide an improved design process for flexible aircraft control. Furthermore, it explains how this proposed process can be adopted progressively in a modular fashion, with each intermediate process bringing improvements over the previous ones. This modularity, in addition to reference guides supplied in Appendices A and B, will further facilitate its adoption by the industry.

Finally, Chapter 7 concludes this thesis, offering a summary of the works presented along with the limitations of the results obtained and an outlook for potential future research expanding on the topics explored.

CHAPTER 2 LITERATURE REVIEW AND OBJECTIVES

This chapter will first present the literature review, providing the necessary technical context for this thesis, followed by the research objectives, associated methodology, and contributions.

2.1 Literature review

This literature review covers relevant topics and expands on those mentioned in the introduction to provide further context for the subsequent chapters.

First, preliminary topics are presented, beginning with the common modelling of a rigid aircraft, which serves as the starting point for developing a new aircraft control law, followed by the relevant ASE phenomenon that this simplified model neglects. Next, various flexible modelling approaches aimed at capturing these effects in simulations are detailed. Then, control-related topics are explored, including notch filtering, robust control, gain scheduling, and feedback linearization, with particular attention given to INDI. Finally, performance verification methods, such as Monte Carlo and worst-case analysis, are introduced to complete this review.

2.1.1 Preliminary topics

Preliminary topics include the modelling of a rigid aircraft and the definition of ASE phenomena.

2.1.1.1 Rigid aircraft model

Aircraft modelling for control design is typically performed under the following assumptions:

- Fuel consumption is negligible.
- Earth's rotation is negligible.
- Earth's surface is considered flat.
- The aircraft is a rigid body (all deformations are neglected).

These simplifications are justified given the time scale of flight control systems, though they would be unsuitable for navigation purposes. They also allow the use of the well-known equations of motion that describe the six degrees-of-freedom dynamics of a rigid body.

To properly represent the state variables, two reference frames are considered: an inertial frame $\mathcal{O}$ fixed to the ground (following the North-East-Down convention) and the body frame $\mathcal{B}$, which is centred at the centre of mass, with the x-axis pointing forward and the z-axis pointing downward.

The following state variables[1] are considered:

- $\mathbf{p} = [p_N \;\; p_E \;\; p_D]^\top$ represents the position of the origin of $\mathcal{B}$ relative to the origin of $\mathcal{O}$.
- $\mathbf{V} = [u \;\; v \;\; w]^\top$ represents the translational velocity expressed in $\mathcal{B}$.
- $\mathbf{\Phi} = (\phi, \;\; \theta, \;\; \psi)$ represents the Euler angles defining the rotation between $\mathcal{O}$ and $\mathcal{B}$.
- $\boldsymbol{\omega} = [p \;\; q \;\; r]^\top$ represents the angular velocity of $\mathcal{B}$ relative to $\mathcal{O}$.

With these state variables, the six degrees-of-freedom rigid body dynamics can be described as follows:

$$\dot{\mathbf{p}} = \mathbf{R}_{\mathcal{B}/\mathcal{O}}(\mathbf{\Phi})\mathbf{V} \tag{2.1a}$$

$$\dot{\mathbf{\Phi}} = \mathbf{H}(\mathbf{\Phi})\boldsymbol{\omega} \tag{2.1b}$$

$$\dot{\mathbf{V}} = \frac{1}{m}\sum \mathbf{F} + \mathbf{R}_{\mathcal{O}/\mathcal{B}}\mathbf{g} - \boldsymbol{\omega} \times \mathbf{V} \tag{2.1c}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}_b^{-1}\Big(\sum \mathbf{T} - \boldsymbol{\omega} \times \mathbf{J}_b\boldsymbol{\omega}\Big) \tag{2.1d}$$

where

- $\mathbf{R}_{\mathcal{B}/\mathcal{O}}$ is the rotation matrix from $\mathcal{B}$ to $\mathcal{O}$ and is given by

$$\mathbf{R}_{\mathcal{B}/\mathcal{O}} = \mathbf{R}_{\mathcal{O}/\mathcal{B}}^\top = \mathbf{R}_x^\top(\phi)\mathbf{R}_y^\top(\theta)\mathbf{R}_z^\top(\psi)$$

$$= \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.2}$$

- $\mathbf{H}$ is the transformation matrix for angular velocities and is given by

$$\mathbf{H}(\mathbf{\Phi}) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.3}$$

[1] The vector $\mathbf{p}$ is expressed in $\mathcal{O}$, while the velocities ω and $\mathbf{V}$ are expressed in $\mathcal{B}$.

- $\mathbf{g} = [0, 0, g]^\top$ is the gravitational acceleration;
- m is the total mass;
- $\mathbf{F}$ and $\mathbf{T}$ are the external forces and torques applied to the rigid body (excluding gravity);
- $\mathbf{J}_b$ is the aircraft inertia matrix;
- $\times$ denotes the cross-product.

The external forces and torques ($\mathbf{F}$, $\mathbf{T}$) can be categorized into aerodynamic forces and torques, as well as thrust generated by the engines. The aerodynamic forces and torques (F_i, T_i) are challenging to characterize and are typically represented using equations such as

$$F_i = \bar{q}SC_i, \quad T_i = \bar{q}SkC_i \tag{2.4}$$

where C_i are empirically determined aerodynamic coefficients, $\bar{q}$ is the dynamic pressure, S is a reference surface, and k is a characteristic distance (such as the wingspan or mean aerodynamic chord, depending on the torque considered). The aerodynamic coefficients vary over time and depend on several parameters that change with flight conditions, including altitude, angle of attack, sideslip angle, control surface positions, Reynolds number, Mach number, etc. Consequently, approximations and interpolations between data points, obtained through wind tunnel testing or flight tests, are employed to develop a more comprehensive model.

It should be noted that aerodynamic coefficients can also be estimated using thin airfoil theory, which assumes the wing has no thickness and an infinite span, or strip theory, which divides the wing into spanwise strips that may experience different flow speeds or angles of attack due to angular rates ω or geometric features such as the dihedral angle or the wing sweep angle.

The nonlinear rigid aircraft model can be linearized around different equilibrium conditions to obtain a linear state-space model:

$$\dot{\mathbf{x}}_R = \mathbf{A}_R\mathbf{x}_R + \mathbf{B}_R\mathbf{u} \tag{2.5a}$$

$$\mathbf{y} = \mathbf{C}_R\mathbf{x}_R + \mathbf{D}_R\mathbf{u} \tag{2.5b}$$

where $\mathbf{x}_R$ is the state vector for the rigid model (also referred to as the vector of rigid modes), $\mathbf{u}$ and $\mathbf{y}$ are the input (commands) and output (measurements) vectors respec-

tively, and $\mathbf{A}_R, , \mathbf{B}_R, , \mathbf{C}_R, , \mathbf{D}_R$ are the matrices describing the system. These matrices depend on several parameters, including the aerodynamic coefficients, and thus on the flight condition considered. It is important to note that the states of this linear model are typically chosen to represent the deviations from equilibrium of the degrees of freedom, such as $\mathbf{x}_R = [\Delta x \ \Delta y \ \Delta z \ \Delta\phi \ \Delta\theta \ \Delta\psi \ \Delta u \ \Delta v \ \Delta w \ \Delta p \ \Delta q \ \Delta r]^\top$.

2.1.1.2 Aeroservoelastic phenomenons

ASE phenomena emerge from the interaction between an aircraft's structure, the aerodynamic forces acting on it, and the flight control system. These phenomena include flutter, buffeting, divergence, and reversal. They were first observed in aviation's early history, with divergence being suspected as the cause of the crash of Langley's flying machine [7], and they remain of critical importance to this day.

Description

- **Flutter**: Dynamic instability where the interaction between a flexible structure and a fluid generates vibrations. In a linear model, the poles associated with the flexible structure move further to the right on the complex plane as airspeed increases. Flutter occurs when these poles reach the imaginary axis. The speed at which harmonic motion begins is called the "flutter speed" and varies with altitude. The linear model becomes unstable when the airspeed surpasses the flutter speed, as the structural poles cross into the right half-plane. However, a nonlinear model shows that the flutter phenomenon persists beyond the flutter speed and becomes a limit-cycle oscillation (LCO). While stable in the Lyapunov sense, these LCOs are problematic in practice due to the premature fatigue of the oscillating structure. Compared to other ASE phenomena, these oscillations are low-frequency and of higher amplitude, making them the most concerning from a control perspective.

- **Buffeting**: High-frequency oscillations that occur when the airflow around the wing detaches, causing turbulence that can stimulate the flexible modes of the structure.

- **Divergence**: Static instability related to wing torsion. For a given airspeed, the forces acting on an airfoil are in equilibrium for a given torsion angle. Divergence occurs when the torsion angle at equilibrium tends to infinity.

- **Reversal**: Control reversal is a phenomenon where the wing deformation (due to torsion) is substantial enough to alter, or even reverse, the effectiveness of the control

surface. As with divergence, the speed at which reversal due to wing torsion occurs can be determined analytically.

Characterisation of flutter

The first model for flutter characterization was created by Theodorsen [8], a pioneer in unsteady aerodynamics studies. Following his work, several researchers (including [9]) determined the flutter speeds and the natural frequencies of flutter oscillations for their aircraft, allowing them to avoid the phenomenon altogether by restricting the flight envelope's allowable speeds. Other researchers [10–20] investigated LCOs and eventually designed control strategies capable of breaking the cycles and returning to stability (exponential or asymptotic, instead of the weaker Lyapunov stability). Additional work [21–29] focused on store-induced LCOs, where a significant external mass (a store) is suspended under the wings and alters the aircraft's structural dynamics, thus influencing the resulting LCO.

These identification and characterization efforts have contributed to the development of flexible aircraft models and controllers capable of extending the flight envelope by reducing the oscillations caused by flutter.

2.1.2 Modelling of flexible aircraft

The first step in control design consists of obtaining a representative model of the system under study. As advances were made to identify and characterize the ASE phenomena, improvements were proposed in flexible aircraft modeling to capture these effects.

This section will first present NASA's flexible wing model, which will serve as a reference point for the subsequent subsections. Then, relevant finite-element modeling approaches are described, as they are useful for capturing complex geometries. Finally, two flexible aircraft modeling approaches are presented: the augmented approach, where the rigid aircraft is augmented with aeroservoelastic dynamics (obtained either through a flexible wing model or a FEM), and the integrated approach, a modular approach where each component is modeled separately before being integrated together.

2.1.2.1 Flexible wing model

The emergence of ASE phenomena has motivated National Aeronautics and Space Administration (NASA) to initiate the Benchmark Aeroelastic Models program [30] to gather experimental data for the analysis of these phenomena. As part of this program, the Benchmark Active Control Technology (BACT) project provided an experimental setup aimed at aiding

the analysis and design of controllers capable of mitigating flutter. An analytical model of the BACT was presented by [1] to replace the previously used data-driven models.

The BACT is a rigid rectangular wing (NACA 0012 airfoil) mounted on a device consisting of rods and pulleys, known as the Pitch and Plunge Apparatus, which allows the rigid wing to move vertically (plunge) and rotate around its longitudinal axis (pitch). Figures 2.1 and 2.2 present the setup. The BACT is equipped with a trailing-edge control surface and two spoilers.

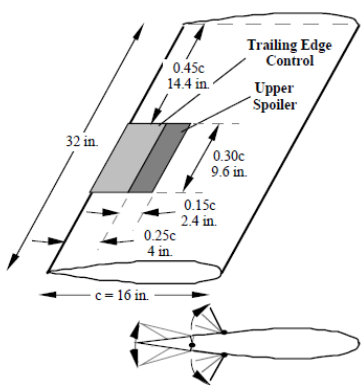

Figure 2.1 Wing section of the BACT (from [1])

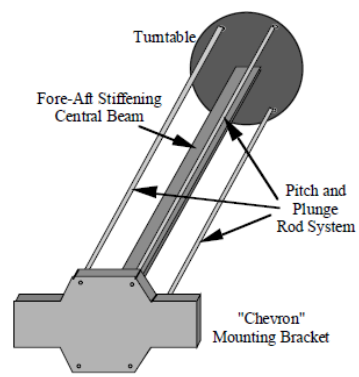

Figure 2.2 Pitch And Plunge Apparatus of the BACT (from [1])

The analytical model developed by [1] uses an approach based on Lagrange's equations and the principle of virtual work. Figure 2.3 presents the diagram used for modelling, with Table 2.1 defining the parameters used.

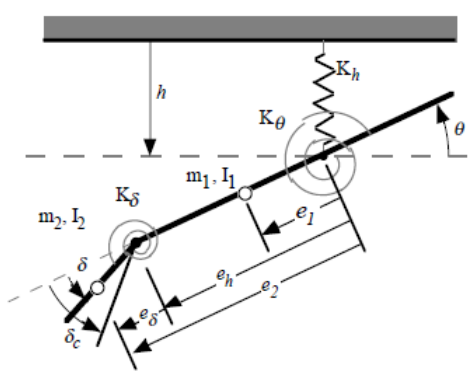

Figure 2.3 Diagram of the BACT (from [1])

Table 2.1 Parameters of the BACT diagram of Fig.2.3

Symbol	Definition
h	vertical displacement
θ	pitch angle
δ	control surface position
δ_c	control surface commanded position
m_1	mass of the wing
m_2	mass of the control surface
I_1	inertia of the wing
I_2	inertia of the control surface
K_h	stiffness of the vertical motion (virtual)
K_θ	stiffness of the pitching motion (virtual)
K_δ	stiffness of the control surface hinge
e_1	position of the center of mass for the wing
e_2	position of the center of mass for the control surface
e_h	position of the control surface hinge
e_δ	position of the center of mass for the control surface relative to its hinge

The equations of motion for the wing are then given by:

$$(\mathbf{M}_s - \mathbf{M}_a)\ddot{\mathbf{q}} + (\mathbf{D}_s - \mathbf{D}_a)\dot{\mathbf{q}} + (\mathbf{K}_s - \mathbf{K}_a)\mathbf{q} = \mathbf{Q}_0^e + \mathbf{Q}_T\theta_T + \mathbf{M}_g g\cos\theta_T + \mathbf{B}_2\ddot{\delta} + \mathbf{B}_1\dot{\delta} + \mathbf{B}_0\delta + \mathbf{E}\mathbf{w} \tag{2.6}$$

where

- $\mathbf{q} = [h,\, \theta]^\top$ are generalized coordinates
- $\mathbf{w}$ is the vector of disturbances
- θ_T is the angle of the device (simulating the angle of attack)
- $\mathbf{M}$ is the inertial matrix
- $\mathbf{D}$ is the damping matrix
- $\mathbf{K}$ is the stiffness matrix
- $\mathbf{Q}$ is the matrix of generalized forces
- $\mathbf{B}$ is the controlled input matrix
- $\mathbf{E}$ is the uncontrolled input (disturbance) matrix
- $(\cdot)_s$ denotes terms related to the structure

- $(\cdot)_a$ denotes terms related to aerodynamics
- $(\cdot)_g$ denotes terms related to gravity
- $(\cdot)^e$ denotes terms of external sources (aerodyanmics)

Regrouping the matrices of inertia, damping, and stiffness in this manner highlights the aerodynamic effects on the structure.

Additional work carried out on the BACT gathered experimental data characterizing the conditions under which flutter occurs [31, 32]; the natural frequencies of the oscillations are between 3 and 6 Hz for the considered airfoil. While the BACT model deserves its share of praise, it remains a linear model incapable of capturing nonlinear effects such as limit cycles.

2.1.2.2 Finite-Element Modeling

This approach can be split into the aerodynamics model and the structural model. These models can be combined to yield an incomplete flexible aircraft dynamics model, which can be used to augment a rigid aircraft model into a complete flexible aircraft model. Given the number of states involved due to the use of finite elements, model order reduction methods are mentioned.

Aerodynamics

The search for a complete model of flexible aircraft has stimulated fluid dynamics research. Several approaches have succeeded one another, each bridging the gaps left by its predecessors. These advances introduced the potential flow theory, which uses a scalar velocity potential whose gradient yields a velocity field, and eventually led to the inclusion of viscous and compressibility effects in subsequent methods. The dynamics models obtained are described using partial differential equations (PDEs), which can admit infinite-dimensional solutions. Unfortunately, the analytical solutions found are only applicable to simple wing shapes, such as thin plates. To address this issue, Prandtl proposed the lifting-line theory, based on potential flow, which allows the analysis of three-dimensional finite wings by introducing the concept of circulation and vortex shedding to account for spanwise variations in the distribution of lift. Expanding on this, the vortex lattice method (VLM) is a finite-element modeling approach that uses a lattice of horseshoe vortices (instead of just one). While it has proven suitable for complex wing geometries and capturing aerodynamic interactions between different parts of an aircraft, the VLM fails to account for unsteady aerodynamic effects and is mostly used to assess lift-induced drag. As such, the doublet lattice method (DLM) was proposed to complement the VLM by discretizing the wing further

into several chord-wise panels, each with a doublet line and a control point, as shown in Fig. 2.4. The doublet line represents a sink and a source for potential flow calculations and allows the computation of the downwash at the control point, which is induced by the vortices, using boundary conditions on the panel and numerical recursive integration over the entire wing. This approach enables the modeling of unsteady aerodynamic effects induced by flexible wings by finding the relationship between aerodynamic forces and the states (position, velocity, acceleration) of various points on the wing.

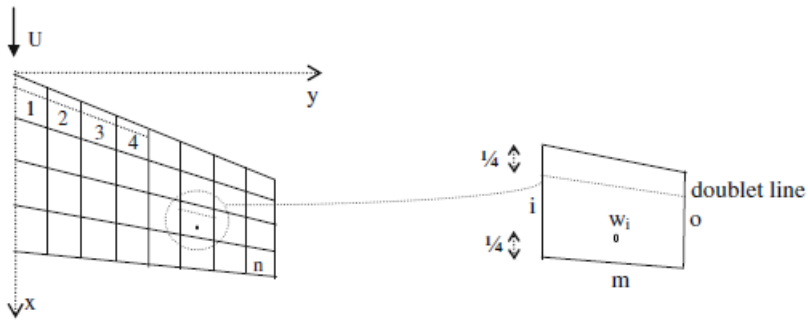

Figure 2.4 Discretization of a complex surface (from [2]) using the Doublet Lattice Method

While the Doublet Lattice Method (DLM) requires several approximations, uses linearized equations from potential flow theory, and necessitates a sufficient number of panel elements to achieve the desired accuracy, its ease of implementation (thanks to NASA's NASTRAN software) and versatile capabilities make it a popular method in the aviation industry.

Frequency-domain solutions

In order to speed up computations, frequency-domain solutions are often sought for unsteady aerodynamics problems, such as those studied using DLM. Unfortunately, these solutions cannot be used directly to simulate the aircraft, which poses challenges when verifying the designed controllers. As such, frequency-domain solutions (which take the form of a complex-valued matrix for each frequency evaluated) are approximated by transfer functions [33–36], which can then be used to generate time-domain responses through inverse Laplace transforms.

For example, the popular Padé approximant assumes that aerodynamic forces have the following shape:

$$\mathbf{Q}(s) = \mathbf{A}_0 + \mathbf{A}_1 s + \mathbf{A}_2 s^2 + \sum_{i=1}^{n} \mathbf{A}_{i+2} \frac{s}{s + b_i} \tag{2.7}$$

The first three terms represent the contributions from position, speed, and acceleration of

different points on the structure (such as the DLM's control points), while the n following terms represent contributions from various aerodynamic lags. The precision of this approximation can be improved by increasing the number of lags, though this comes at the cost of increasing the order of the resulting system.

We note that this approximation is designed to resemble (2.6) in structure (with $\mathbf{M}_a$, $\mathbf{D}_a$, $\mathbf{K}_a$, $\mathbf{Q}_0$, $\mathbf{Q}_T$, $\mathbf{B}_0$, $\mathbf{B}_1$, and $\mathbf{B}_2$ being extracted from these $\mathbf{A}$ matrices). Furthermore, using the Vortex Lattice Method (VLM) results in only the $\mathbf{A}_0$ term, as it disregards unsteady dynamics.

Structure

The structural dynamics describing the motion of a flexible structure is a key topic within the strength of materials engineering discipline. For example, equations that describe how a beam bends or how a rod twists under stress, along with the harmonic oscillations they undergo, are now standard textbook examples for aspiring mechanical engineers. While these are partial derivative equations (PDE)s yielding infinite-dimensional solutions, the solution for a complex structure can be approximated by discretizing it into finite elements, such as nodes with 6 degrees of freedom connected by beams and rods. The resulting equations of motion typically take the form of a forced mass-damper-spring oscillator in matrix format, as shown by:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{F} \tag{2.8}$$

where $\mathbf{x}$ is the vector of nodal displacements or, equivalently[2], the vector of harmonic modes [36]. We note that NASTRAN software also supports such structural analysis, which contributes to its widespread use in the aviation industry.

2.1.2.3 Augmented approach

This approach to flexible aircraft modelling involves combining both the structural and aerodynamic FEMs into a purely flexible model and using it to augment the rigid aircraft model, thereby accounting for rigid-body dynamics not captured in the FEMs. This approach models a flexible aircraft as a rigid aircraft subjected to external forces generated by interactions with the flexible structures.

[2]The vector of flexible modes is obtained from the vector of nodal displacements through an invertible transformation matrix. This transformation matrix consists of "mode shape" vectors, as it identifies a set of constant modal displacements corresponding to the unit value of a given mode.

Pure flexible model

By combining the structural dynamics with the aerodynamic forces derived from the DLM method, we obtain a flexible model[3] that captures aeroelastic effects [36]. This type of model can be represented by

$$\mathbf{M}\ddot{\eta} + \mathbf{C}\dot{\eta} + \mathbf{K}\eta = \mathbf{Q}\eta \tag{2.9}$$

where η is the vector of rigid, flexible and control surface modes, $\mathbf{M}$ is the modal inertial matrix, $\mathbf{C}$ is the modal damping matrix, $\mathbf{K}$ is the modal stiffness matrix, and $\mathbf{Q}$ is the matrix of aerodynamics coefficients (which depends on frequency). By rearranging (2.9), we get the following state-space model:

$$\dot{\mathbf{x}}_A = \mathbf{A}_F\mathbf{x}_A + \mathbf{B}_F\mathbf{u} \tag{2.10a}$$

$$\mathbf{y} = \mathbf{C}_F\mathbf{x}_A + \mathbf{D}_F\mathbf{u} \tag{2.10b}$$

where $\mathbf{x}_A = [\tilde{\mathbf{x}}_R^\top, \mathbf{x}_F^\top]^\top$ is the vector of rigid and flexible modes, and

$$\mathbf{A}_F = \begin{bmatrix} \mathbf{A}_{RR} & \mathbf{A}_{FR} \\ \mathbf{A}_{RF} & \mathbf{A}_{FF} \end{bmatrix} \tag{2.11}$$

with the subscript XY meaning "effect of the X modes on the Y modes".

Unfortunately, it was found in practice[4] that the matrix $\mathbf{A}_{RR}$ does not contain the low-frequency dynamics expected from an aircraft. This flexible model is thus deemed incomplete as it is missing the rigid aircraft dynamics.

Complete model

To obtain a complete model for a flexible aircraft, one need only augment the rigid aircraft state-space model to account for the flexible modes while preserving the rigid dynamics. This model is described as follows:

$$\begin{bmatrix} \dot{\mathbf{x}}_R \\ \dot{\mathbf{x}}_F \end{bmatrix} = \begin{bmatrix} \mathbf{A}_R & \tilde{\mathbf{A}}_{FR} \\ \tilde{\mathbf{A}}_{RF} & \mathbf{A}_{FF} \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_F \end{bmatrix} + \begin{bmatrix} \mathbf{B}_R \\ \mathbf{B}_F \end{bmatrix} \mathbf{u} \tag{2.12a}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_F \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_F \end{bmatrix} + \begin{bmatrix} \mathbf{D}_F \end{bmatrix} \mathbf{u} \tag{2.12b}$$

[3]Since the nodes of the doublet lattice may not correspond directly to the nodes of the structural mesh, an invertible transformation matrix needs to be assembled from the modeshape functions evaluated at the nodal locations. Alternatively, a matrix of spline functions can be used to interpolate the aerodynamic forces at the structural nodal positions.

[4]This conclusion comes from Bombardier, an industrial partner.

where $\mathbf{A}_R$ is the state matrix $\mathbf{A}$ from the rigid model (2.5a) and replaces the $\mathbf{A}_{RR}$ matrix from the flexible model (2.10a), and where the matrices $\mathbf{A}_{FR}$ and $\mathbf{A}_{RF}$ are modified by a transformation matrix that links the rigid modes $\tilde{\mathbf{x}}_R$ of the flexible model with the rigid states $\mathbf{x}_R$ of the rigid model. We note that this improvement to the flexible model was possible because the dynamics described by the $\mathbf{A}_{RR}$ matrix are negligible.

This complete model captures both the rigid dynamics and the ASE effects. It can be viewed as a rigid aircraft perturbed by flexible dynamics, making it convenient to implement if a rigid model is already available.

Reduction

Models generated using FEMs are generally of high order, which can be cumbersome for controller synthesis. It is therefore common practice to use model order reduction methods to find an acceptable approximation of the complete model. Among the popular methods are Singular Value Decomposition, balanced reduction [37], and proper orthogonal decomposition [38].

2.1.2.4 Integrated approach

A different approach to flexible aircraft modeling was proposed [4], where the equations of motion are derived from energy equations and expressed in terms of momenta. This allows for the combination of structural dynamics with rigid-body dynamics, where the structural dynamics are expressed as an integral over every mass element of the aircraft. As such, the structural dynamics can be defined component-wise (fuselage, wings, empennage) and spatially discretized using different techniques such as Galerkin methods, assumed modeshapes, or finite-element modeling. Furthermore, while [4] used strip theory to model the aerodynamic forces, [39] replaced it with a modified version incorporating Theodorsen's work [8] to capture unsteady aerodynamic effects. The modularity of this approach can be useful for development, as one can start with simpler analytical models and then update the model with more accurate FEMs once they are made available.

Unfortunately, this integrated approach involves the inversion of a time-dependent inertia matrix, which proves computationally intensive compared to the previously mentioned augmented approach still favoured by the industry. Despite this drawback, the integrated approach can yield more accurate results, as the augmented approach requires a linear approximation of the flexible models to compute their effect on the rigid-body dynamics. As such, we will consider this integrated approach as a potential alternative in cases where extensive data (for FEMs) from the aircraft manufacturer are not available, such as during early design

iterations.

2.1.3 Control

This section will first present well-established linear control methods, followed by a selection of nonlinear approaches that leverage the linear tools already widely used by the industry.

Linear methods

The control of Linear Time-Invariant (LTI) systems has been widely covered in the literature and includes various tuning methods for the classic Proportional-Integral-Derivative (PID) controller. The simplest methods are heuristic in nature (such as Ziegler-Nichols, Cohen-Coon, or simply trial and error), which proves impractical for aerospace applications. Popular methods include optimal control approaches (such as Linear-Quadratic Regulator (LQR) or Linear-Quadratic Gaussian (LQG)), pole placement, and eigenstructure assignment [40]. We note, however, that control for flexible aircraft requires special attention; all of these linear methods fail to reject ASE effects on their own and require the use of ASE filters to mitigate them. Furthermore, most of these methods assume that the linear model used during design is accurate and are thus not robust to modelling errors.

As such, this section will focus on ASE filtering and robust control linear methods, especially structured $\mathcal{H}_\infty$ synthesis.

Nonlinear methods

Given the nonlinear nature of flexible aircraft, nonlinear control methods must be used. Various methods have been proposed in the literature [41], with notable approaches such as adaptive control, sliding mode control, backstepping methods, the nonlinear version of optimal control (State-Dependent Riccati Equation), and PDE-based methods (such as boundary control [42] or distributed control). Of particular interest for this thesis are methods that do not increase the workload on designers, specifically control law approaches that allow the use of well-established linear control tools and methods. The most popular of these methods are gain-scheduling, the current standard in the aviation industry, and feedback linearization. Special attention is given to INDI, a sensor-based variant of feedback linearization that shows additional potential for the project at hand.

2.1.3.1 ASE filters

The simplest method to mitigate ASE effects is to add specially designed notch filters [3, 43]. As mentioned in Section 2.1.2.3, a flexible aircraft can be seen as a rigid aircraft perturbed

by flexible modes. This formulation can be framed as a nominal system (rigid) being perturbed by residual dynamics (flexible modes) through spillover [43]. The frequency response of a flexible aircraft (Fig.2.5) contains spikes in magnitude that correspond to the stimulated flexible modes. Designing a notch filter to cut these specific frequencies is effective at decreasing the impact of the associated flexible modes on the aircraft dynamics, thereby reducing spillover effects.

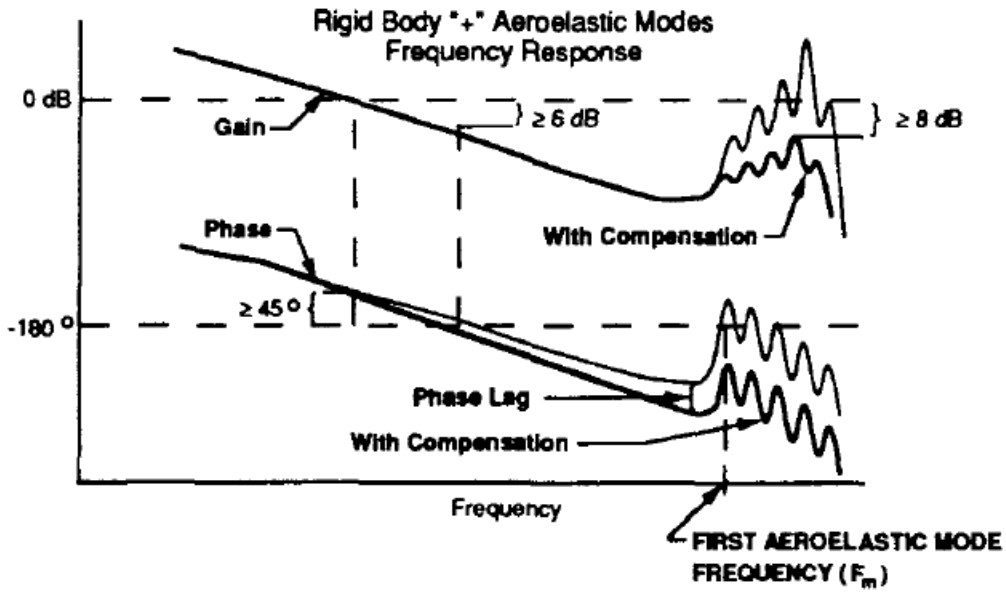

Figure 2.5 Concept of ASE compensation through notch filtering (from [3])

While this method is simple and easy to implement, the tuning process for these filters is typically done by trial and error in practice. Furthermore, the mitigation of ASE effects is usually limited, as the filters are generally fixed for all flight conditions.

2.1.3.2 Robust control

The robust control approach focuses on taking the uncertainties of a system into account during controller synthesis. The formulation of a robust control problem begins with the use of the standard framework to model the system. Figure 2.6 presents a diagram of a closed-loop system using the standard framework, where:

- **P** is the system, usually augmented with (virtual) weighting filters;
- **K** is the controller;

- $\mathbf{u}$ is the control input vector (endogenous inputs);
- $\mathbf{y}$ is the measured signals vector (endogenous outputs);
- $\mathbf{w}$ is the vector containing the control reference and disturbances (exogenous inputs);
- $\mathbf{z}$ is the vector of regulated outputs (exogenous outputs).

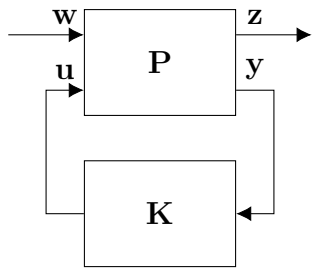

Figure 2.6 Closed-loop system (standard framework)

The weighting filters represent the requirements and are added during synthesis. They are not implemented on the real aircraft. The use of such filters is central to the $\mathcal{H}_\infty$ synthesis, the robust control method considered here.

The augmented system can be expressed as follows,

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \quad \mathbf{u} = \mathbf{K}\mathbf{y} \tag{2.13}$$

The transfer function linking $\mathbf{w}$ and $\mathbf{z}$ can then be rewritten as a lower Linear Fractional Transformation (LFT):

$$\mathbf{T}_{wz}(s) = \mathbf{F}_l(\mathbf{P}(s), \mathbf{K}(s)) = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(\mathbf{I} - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21} \tag{2.14}$$

The (optimal) $\mathcal{H}_\infty$ problem to solve is to find the stabilizing controller $\mathbf{K}$ that minimizes the $\mathcal{H}_\infty$ norm of the transfer function linking the exogenous inputs and outputs. The $\mathcal{H}_\infty$ norm is defined as

$$\| \mathbf{T}_{wz}(s) \|_\infty = \sup_\omega \bar{\sigma}(\mathbf{T}_{wz}(j\omega)) \tag{2.15}$$

where $\bar{\sigma}$ is the maximum singular value of the matrix $\mathbf{T}_{wz}(j\omega)$.

The classical solution to the $\mathcal{H}_\infty$ problem yields a controller of the same order as the system considered [44, 45], which is at least the order of the augmented system (which includes the

weighting filters). Such a controller is usually impractical to implement and thus requires model order reduction in practice, which alters its performance.

Structured $\mathcal{H}_\infty$ synthesis

To address the implementation issue and extend the usefulness of the $\mathcal{H}_\infty$ method, the structured $\mathcal{H}_\infty$ synthesis was developed to allow the design of controllers with a fixed architecture [46]. This approach uses the same principles as the classic $\mathcal{H}_\infty$ synthesis, but considers that only some parameters are actually tunable within the controller's fixed architecture. The optimization problem becomes non-smooth and is thus usually solved sub-optimally, where the cost value obtained is slightly larger than the real optimum. As such, there is no guarantee that the minimum found is the global minimum of the function being optimized. Several iterations, each using different initial conditions, are required to seek a smaller local minimum, hopefully closer to the global minimum and the optimal cost. Solving such problems has been made more convenient thanks to the implementation of the algorithm in MATLAB software [47–50] under the `systune` command.

The versatility of being able to tune fixed-architecture controllers has allowed the structured $\mathcal{H}_\infty$ synthesis to support multiple objectives and models at once, as well as self-scheduling [5, 51] and the inclusion of filter parameters (including ASE notch filters) [52].

2.1.3.3 Gain scheduling

Gain scheduling [53, 54] is a technique that can use different controllers while maintaining the same architecture. This shared architecture allows the controller's tunable parameters to vary continuously as a function of a selected flight parameter (ρ), as shown in Fig. 2.7. This method enables the design of linear controllers for different flight conditions and allows smooth transitions between tunings (through suitable interpolation of tunable parameters) during flight to match the current conditions.

While gain scheduling presents certain drawbacks, such as hidden coupling terms[5] [40, 55] and the need to smooth the scheduling (interpolating) functions to prevent sudden jumps in parameter values, its simplicity and versatility have made it a staple in the industry [56]. However, the increasing complexity of modern aircraft dynamics necessitates frequent updates to aerodynamic models during development, leading to lengthy and tedious re-tuning of gain-scheduled controllers, unless a self-scheduling approach, such as one incorporating structured

[5] These coupling terms arise due to the additional feedback loop introduced by the scheduling function, as shown in Fig. 2.7. Since this feedback is generally neglected during linearization, these terms remain "hidden" as they are absent from the models used for tuning, leading to performance degradation.

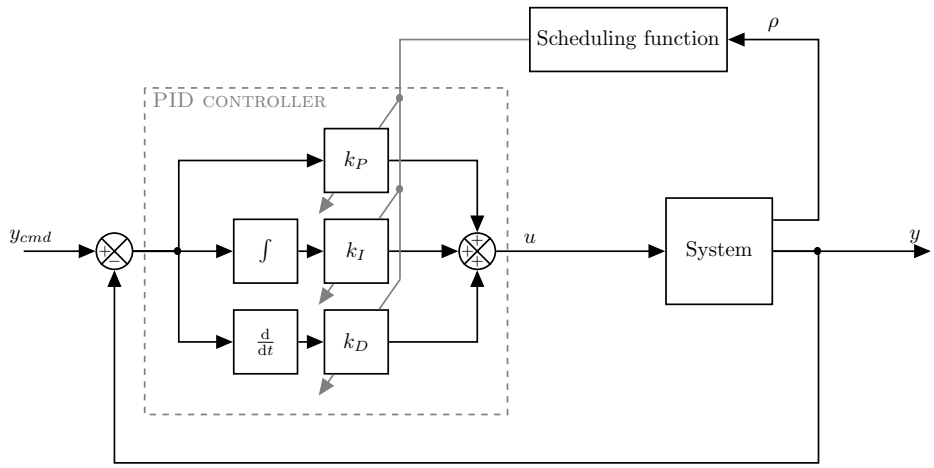

Figure 2.7 Architecture of a gain-scheduled controller

$\mathcal{H}_\infty$ synthesis [5, 51], is employed.

2.1.3.4 Feedback linearization

This approach, also known as dynamic inversion or Nonlinear Dynamic Inversion, involves designing a specific control input signal to cancel out the system's nonlinear dynamics. A linear controller can then be implemented as an outer loop to regulate the resulting linearized (closed inner-loop) system.

Figure 2.8 presents the architecture of NDI, where the system is defined as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \tag{2.16a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{2.16b}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector, $\mathbf{y} \in \mathbb{R}^m$ is the output vector meant to be controlled, $\mathbf{f}$ and $\mathbf{h}$ are smooth vector fields, and $\mathbf{G} \in \mathbb{R}^{n\times m}$ is a matrix with columns of smooth vector fields.

Differentiation of the output equation yields [57]:

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}) = \mathcal{L}_f\mathbf{h}(\mathbf{x}) + \mathcal{L}_G\mathbf{h}(\mathbf{x})\mathbf{u} = \mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u} \tag{2.17}$$

where $\mathcal{L}_f\mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})$ and $\mathcal{L}_G\mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{G}(\mathbf{x})$ are the Lie derivatives of $\mathbf{h}$ along $\mathbf{f}$ and $\mathbf{G}$ respectively. It is assumed that $\mathcal{L}_g h(\mathbf{x}) \neq \mathbf{0}$ and that it is invertible. Defining the virtual

control input ν as the desired $\dot{\mathbf{y}}$, the following control law is obtained by inverting Eq.(2.17):

$$\mathbf{u} = \mathbf{B}^{-1}(\hat{\mathbf{x}})(\nu - \mathbf{a}(\hat{\mathbf{x}})) \tag{2.18}$$

where $\hat{\mathbf{x}}$ is the estimation of the state vector. Note that the system is only stable if the relation between $\mathbf{y}$ and $\mathbf{u}$ is of minimum phase [57].

Assuming exact knowledge of $\mathbf{a}$, $\mathbf{B}$ and $\mathbf{x}$, the closed-loop input-output response reduces to a set of integrators ($\dot{\mathbf{y}} = \nu \Leftrightarrow \mathbf{y}(t) = \int_o^\tau \nu(\tau)\mathrm{d}\tau$).

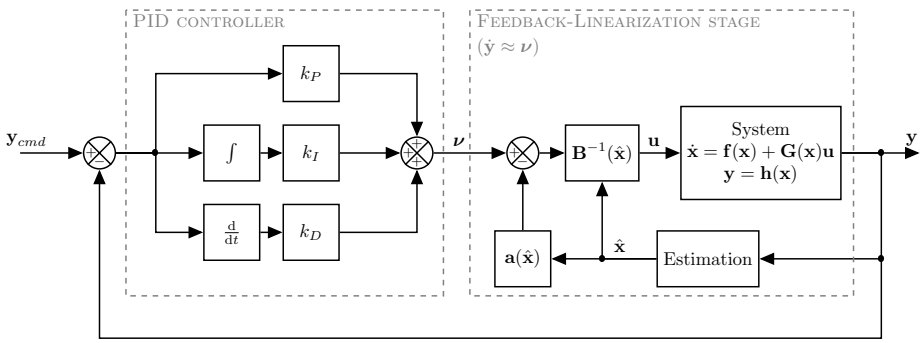

Figure 2.8 Architecture of the NDI control law

The main drawback of this approach is its sensitivity to modelling errors in the inverted nonlinear dynamics, as these errors result in only partial linearization and prevent full cancellation of the nonlinear effects. To address this issue, an adaptive component can be added to eliminate parametric uncertainties, or a "robustification" component [58] can be used to maintain a given level of performance despite uncertainties.

Incremental Nonlinear Dynamic Inversion

To improve the NDI approach, research on a sensor-based modification [59] and a more general formulation (as opposed to the previously aircraft-specific one) [60] were combined to develop the Incremental Nonlinear Dynamic Inversion (INDI) control law [61].

This alternative method updates the linearizing command incrementally and incorporates measurements of the state derivatives. It assumes that the sampling rate, along with actuator bandwidth and signal delay synchronization, is sufficiently high to permit a linear approximation of the nonlinear feedback-linearizing command and enable updates based on state derivative measurements. This approach helps mitigate the most significant errors and uncertainties.

Figure 2.9 presents the architecture of INDI control, which can be compared to the NDI

architecture in Fig. 2.8. The system dynamics of Eq.(2.16a) is approximated by truncating a Taylor series expansion around the time of the last available measurement (t_0), yielding

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \frac{\partial}{\partial \mathbf{x}}\left(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right)|_{t=t_0}(\mathbf{x} - \mathbf{x}_0) + \frac{\partial}{\partial \mathbf{u}}\left(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right)|_{t=t_0}(\mathbf{u} - \mathbf{u}_0) \tag{2.19}$$

where $\mathbf{x} = \mathbf{x}(t_0)$, $\mathbf{u}_0 = \mathbf{u}(t_0)$ and $\dot{\mathbf{x}}_0 = \dot{\mathbf{x}}(t_0)$.

It is assumed that the change in input is much faster than the change in state ($\mathbf{x} - \mathbf{x}_0 \approx \mathbf{0}$ while $\mathbf{u} - \mathbf{u}_0 \neq \mathbf{0}$) [57], which allows Eq.(2.19) to reduce to:

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \mathbf{G}(\mathbf{x}_0)(\mathbf{u} - \mathbf{u}_0) \tag{2.20}$$

It is assumed that $\mathbf{G}$ is invertible and that $\mathbf{y} = \mathbf{x}$ (see Eq.(2.16b)). Then, the pseudocommand input ν can be defined as the desired $\dot{\mathbf{x}}$ and the following control law is obtained by inverting Eq.(2.20):

$$\mathbf{u} = \hat{\mathbf{u}}_0 + \hat{\mathbf{G}}^{-1}(\hat{\mathbf{x}}_0)\left(\nu - \hat{\dot{\mathbf{x}}}_0\right) \tag{2.21}$$

where $\hat{\mathbf{u}}_0$ and $\hat{\dot{\mathbf{x}}}_0$ are the measurements (or estimations) of $\mathbf{u}$ and $\dot{\mathbf{x}}$ that must be synchronized for the same time t_0, and where $\hat{\mathbf{G}}(\hat{\mathbf{x}}_0)$ is the estimation of the control effectiveness matrix $\mathbf{G}$ at time t_0 computed using the measured (or estimated) state vector $\hat{\mathbf{x}}$.

Assuming perfect measurements of $\mathbf{u}$ and $\dot{\mathbf{x}}$, and perfect knowledge of $\mathbf{G}$, the closed-loop input-output response reduces to a set of integrators ($\dot{\mathbf{x}} = \nu \Leftrightarrow \mathbf{x}(t) = \int_o^\tau \nu(\tau)\mathrm{d}\tau$). Note that actuators are assumed to be identical and quasi-instantaneous (so their dynamics can be neglected), that the control law is assumed to be sampled at a sufficiently high frequency, and that all measurements are synchronized with sufficiently small delays.

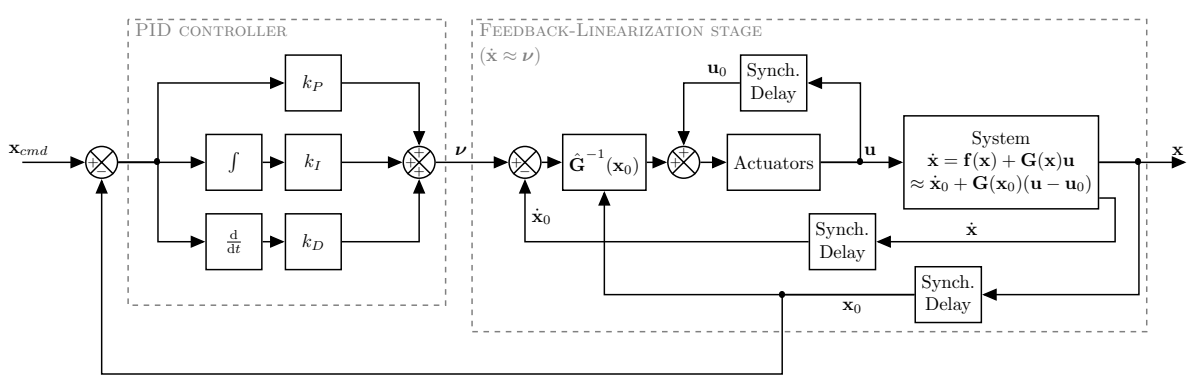

Figure 2.9 Architecture of the INDI control laws

This new control approach was first tested on the VAAC Harrier aircraft but was initially

hindered by hardware limitations, including poor angular acceleration sensors, inadequate signal synchronization due to noise filters, issues with control effectiveness, and signal drift. Despite these challenges, the initial test exceeded expectations, motivating further research to address these issues. Studies explored improvements to the control effectiveness model [62], sensor delays and synchronization [63], and the use of estimated signals in place of measured ones [64]. Additionally, a common misconception in the literature was corrected [6], offering a more rigorous approach that also relaxed applicability conditions by eliminating the need for the time-scale separation principle.

Following these refinements, experimental flight tests were conducted to address "real-world problems." Initially, these tests were primarily performed on small Unmanned Aerial Vehicle (UAV)s, commonly referred to as drones, with sampling rates ranging from 250Hz [65] to 512Hz [62], and even up to 2000Hz [66]. Notably, a flight test of a Cessna Citation in Dutch airspace [57] employed a controller operating at 100Hz while accounting for a range of real-world challenges. This provided a compelling demonstration of the control technique's effectiveness for rigid aircraft in the literature.

The INDI approach shows promise for aircraft control, as it enables the use of a single linear controller across the entire flight envelope. The feedback-linearization loop eliminates variations that would typically necessitate gain scheduling. Additionally, the linear approximation of the linearizing command simplifies implementation, allowing for the use of a basic lookup table for the control effectiveness matrix.

However, the assumptions underpinning stability guarantees remain qualitative and restrictive, often conflicting with practical constraints such as controller frequency, sensor measurement delays and sampling rates, as well as actuator servocontrol frequency and bandwidth. As a result, practitioners frequently rely on heuristic approaches, often left questioning, "How fast is fast enough?" when faced with the high sampling rate and actuator bandwidth requirements.

Various modifications and mitigation strategies have been proposed in the literature, including the weighted pseudo-inverse for control effectiveness and allocation [67], limitations related to angular accelerometers [68], and the challenges posed by finite actuator bandwidth [69]. However, the practical implementation of INDI in industry remains largely unaddressed. Moreover, its application to flexible aircraft appears prohibitively costly, as it would require additional sensors to capture flexible mode states and their derivatives [39].

Further research is needed to reduce the conservatism of these simplifying assumptions by quantifying them and examining the behaviour of marginal systems, such as those using non-identical actuators with non-minimal phase characteristics or relatively low sampling rates.

Addressing these challenges would help bridge the gap between theory and practice, allowing this technology to mature for wider adoption in industry.

2.1.4 Verification

To complete this literature review, this section introduces verification approaches used to assess nonlinear dynamic systems against performance requirements, particularly in evaluating closed-loop system performance and requirement fulfillment. Traditionally, this verification step in control design relies on Monte Carlo simulation campaigns. However, recent advancements have introduced new tools that could enhance this process, such as the worst-case analysis toolbox [70–72].

2.1.4.1 Monte Carlo simulation campaign

Traditional verification and validation are typically performed using statistical methods commonly referred to as "Monte Carlo" in the literature. This approach involves generating a large number of randomized cases, where uncertain parameters or random variables are assigned random values. Each case is then assigned a score, which can be analyzed using statistical tools such as mean and variance. This process allows for estimating the probability of failure (i.e., the score not meeting a given threshold), with the level of confidence depending on the problem at hand and the number of random cases used.

For aircraft control, Monte Carlo campaigns typically follow this straightforward procedure:

- **Sampling:** The system to be simulated is initialized, with all uncertain parameters randomly assigned a valid value based on their probabilistic distribution (uniform, normal, Gaussian, etc.). These uncertain parameters may include flight condition parameters (to cover the flight envelope), aerodynamic coefficients (to account for modeling errors), and weight and center of gravity parameters (to account for variability during normal aircraft operations). This step essentially "samples" the parameter space that defines the simulation conditions.

- **Simulation:** The randomly initialized system (the sample) is then simulated, typically following a scripted maneuver, such as coordinated turns or altitude changes. In statistical terms, this is considered a "trial" (or experiment) performed on the "sample".

- **Evaluation:** The simulation results are assessed against a specified objective (requirement) and classified as either a successful or failed trial.

This three-step procedure is repeated for each randomized trial executed during the Monte Carlo campaign. Once a large number of trials are completed, the rate and distribution of successes and failures are analyzed using statistical tools such as mean and variance. These steps can be parallelized, as modern computing allows all samples to be processed simultaneously, speeding up execution. Despite this, Monte Carlo campaigns can still be time-consuming, even though they are easy to implement and use. While useful for mean or variance analysis, this method is not ideal for identifying worst-case conditions through analysis of minimum and maximum values [71].

2.1.4.2 Worst-case analysis

Worst-case analysis is an optimization-based method that utilizes various algorithms to identify the conditions leading to the worst possible outcome. The most appropriate algorithm depends on the specific application and requirements being considered, as not all cost functions are convex and amenable to gradient-based methods. Non-gradient-based algorithms often rely on heuristics to estimate the best search direction [71]. As a result, worst-case analysis typically converges much faster towards the worst-case conditions compared to traditional Monte Carlo approaches. This speed enables a quicker identification of critical cases that warrant further attention before proceeding to the next steps in the verification and validation process. While worst-case analysis cannot fully replace Monte Carlo campaigns, it serves as a complementary tool, enhancing the overall verification process.

In contrast to the Monte Carlo campaign procedure, the worst-case analysis follows these steps:

- **Initialization:** In this step, the Monte Carlo campaign procedure is initially employed to generate a small pool of randomized trials, which serve as the first "generation" or batch of cases.
- **Alteration:** This step uses optimization algorithms to generate new samples based on the results from the previous generation. These new samples are then simulated, evaluated, and compared with the previous generation. The worst-performing cases are selected to form the next generation.
- **Termination:** The alteration step is repeated until specified termination conditions are met. These conditions can be either fixed (e.g., a set number of generations or iterations) or based on convergence metrics (e.g., a sufficiently small step size).

This optimization-based procedure leverages information from past trials to converge towards the worst-case conditions. Several optimization parameters must be fine-tuned by the practitioner to suit the specific problem at hand, such as the number of cases per generation and algorithm-specific parameters that control the extent of alteration allowed when generating new generations.

To promote these tools, a worst-case analysis toolbox for Matlab was developed by the University of Exeter with support from the European Space Agency (ESA) under grant ESTEC[6], based on prior work from the University of Leicester [72]. This toolbox includes the following algorithms:

- Sequential Quadratic Programming (SQP)
- Differential Evolution (DE)
- Dividing Rectangles (DIRECT)
- Genetic Algorithm (GA)
- Hybrid methods (HDE, HDIRECT, HGA)
- Non-dominated sorted methods (NSGA, NSDE)

For the purpose of worst-case analysis, the toolbox supports Monte Carlo analysis and more advanced techniques, such as Bernstein expansion, which fall outside the scope of this thesis.

We invite the reader to consult the WCAT-II official documentation for further details.

2.2 Objectives, methodology and contribution

With this literature review in mind, the research objectives of the thesis, along with the associated methodology and contributions, are now presented.

The main research objective is defined as follows:

Propose a control design approach for modern aircraft that would reduce development effort without increasing hardware costs when compared with the traditional approach.

To achieve this overarching goal, three specific objectives are outlined below.

[6]Contract No: 19783/06/NL/JD:4000104541

2.2.1 Objective 1: Develop a control law based on feedback-linearization that is compatible with current control design methods

As discussed in the literature review, the gain-scheduling approach introduces several inefficiencies in the control design process, and replacing it with a suitable alternative has the potential to significantly reduce the required workload. Feedback-linearization methods are considered due to their practical properties, particularly their reliance on linear controllers to complete the nonlinear control law. The sensor-based INDI control law is of particular interest, as it relies on sensors rather than on-board models, offering significant potential. However, the qualitative simplifying assumptions underlying the INDI control law may not always hold in practice, often requiring hardware modifications *a posteriori* when initial implementations fail to meet the expected performance. Since any changes to hardware at the control design stage are costly—being late in the development process—there is a clear need for an *a priori* assessment of the applicability of the INDI control law. This assessment should involve quantitative tests rather than relying on qualitative assumptions.

Methodology and contribution

The investigation into the qualitative simplifying assumptions is conducted by translating these assumptions into limits applied to hardware parameters, such as actuator bandwidth or sampling rate tending to infinity, representing the quasi-instantaneous assumptions. An ideal plant model is first created by applying all the limits simultaneously, serving as a baseline for comparison. Next, various non-ideal plant models are generated by sweeping one of the parameters instead of applying the limit. This process allows for the identification of a threshold beyond which the INDI control law fails to stabilize the system, and it also helps assess the performance loss resulting from the violation of the simplifying assumption under scrutiny.

This systematic investigation continues by sweeping two hardware parameters simultaneously to explore how they might interact. This step is crucial because, while the simplifying assumptions are known to be sufficient conditions for INDI applicability, they have not been proven to be necessary or independent (non-redundant). Such an investigation enhances understanding of the inner workings of this control method and reveals potential mitigation techniques to counter the performance loss associated with non-ideal hardware.

By the end of this investigation, a clear list of quantitative tests is generated to assess the applicability of INDI to a given system before any implementation occurs. These tests only require knowledge of the hardware parameters and open-loop plant properties. Each test is accompanied by a suggested mitigation technique, which extends the application of INDI

in cases where the available hardware may not achieve peak performance but still provides acceptable performance. This streamlined applicability assessment, along with the extension of INDI applications, is expected to contribute to increasing the technological readiness level of INDI for industrial and commercial use.

2.2.2 Objective 2: Propose an algorithm-based controller tuning approach that is compatible with both gain-scheduling and the proposed control law based on feedback-linearization

Replacing manual fine-tuning with an automated method is an obvious approach to alleviate the control design workload, especially when the controller is subject to re-tuning as the plant model is refined incrementally during development. The Structured $\mathcal{H}_\infty$ Synthesis, which supports both self-scheduling and fixed-architecture controllers, appears to be a suitable candidate for both Gain Scheduling and the proposed INDI control law. Moreover, the scope of the Structured $\mathcal{H}_\infty$ Synthesis can be easily extended by adding new tunable parameters to the architecture. This versatility allows for the integration of notch filters into the control architecture, which is essential for flexible aircraft applications.

The dual compatibility provided here is intended to facilitate industry adoption by enabling a gradual transition from the traditional design process to the new approach proposed. This alternative tuning method can be implemented at low cost initially, before considering a complete overhaul of the controller architecture in favor of INDI. This also allows control designers to become familiar with the tuning method before dealing with a novel controller architecture.

Furthermore, there is no guarantee that the claimed robustness of the INDI control law will remain intact after modifications. As highlighted by [6], the robustness properties of INDI are not absolute (although it has been shown to improve upon the notoriously non-robust NDI), but its own robustness remains in question. Therefore, the required robustness will be ensured through tuning, using it as an optimization goal, regardless of the inherent robustness (or lack thereof) of the INDI control law.

Methodology and contribution

The applicability of INDI is assessed using a rigid aircraft model, serving as an application example for the proposed criteria of applicability. Given the use of realistic hardware, several applicability tests are expected to fail, but the proposed mitigations are implemented to ensure that INDI is able to stabilize the system as intended. Once the implementation is completed, the linear controller stage of INDI is tuned using the structured $\mathcal{H}_\infty$ synthesis to

meet performance requirements and stability margins. The resulting closed-loop system is then analyzed through both Monte Carlo simulations and worst-case analysis campaigns to ensure proper performance across the entire flight envelope.

This application example demonstrates the proposed applicability tests and serves as a proof of concept for the suggested design process on a conventional aircraft. Along the way, it encounters industry-relevant issues, which helps in advancing the maturation of the applied technology. Additionally, it highlights how the INDI control law and the structured $\mathcal{H}_\infty$ synthesis can complement each other, offering a versatile control design approach that is capable of addressing a wide range of system classes, including flexible aircraft.

2.2.3 Objective 3: Demonstrate that the proposed control law and tuning method are compatible with flexible aircraft

With suitable alternatives to gain-scheduling and manual tuning identified, the next step is to confirm their viability on modern aircraft, which are increasingly characterized by flexibility. To this end, the traditional mitigation method used for flexible aircraft control will be applied to demonstrate the modularity of the proposed changes to the control design. Since notch filtering significantly impacts the phase of the signal, special attention must be given during their tuning to avoid deteriorating closed-loop performance to the extent that it fails to meet the INDI applicability criteria. Once again, the versatility of structured $\mathcal{H}_\infty$ synthesis will be demonstrated through its application to notch filter tuning, showcasing its capability to mitigate reductions in stability margins and performance losses caused by various uncertainties, such as ASE perturbations and the effects of filtering.

Methodology and contribution

An integrated approach is used to generate a flexible aircraft model from publicly available data [4, 39], which is then analyzed to identify the appropriate architecture for the notch filters. This analysis incorporates Singular Perturbation theory, also referenced in the INDI literature when describing its robustness properties [6], to facilitate the identification of natural frequencies that may prove problematic for INDI. The first stage of INDI, aimed at achieving feedback linearization, is implemented with the un-tuned notch filters added. These notch filters are then tuned using structured $\mathcal{H}_\infty$ synthesis to preserve as much stability margin and control bandwidth as possible, while maximizing rejection of ASE perturbations. Special attention is given to the dephasing introduced by filtering, as it may hinder the signal synchronization required by INDI. Following this, the second stage of INDI, the linear control stage, is implemented and robustly tuned in the presence of ASE filters, whose gains have

now been fixed.

This application example serves as both a proof of concept and a demonstration of the versatility of the proposed control design approach. It illustrates how robust tuning can compensate for performance losses caused by modifications and mitigation efforts aimed at extending the applicability of the INDI control law to a broader class of systems, including flexible aircraft with standard actuators. It also addresses the performance losses resulting from ASE phenomena and the notch filters designed to mitigate them. Furthermore, this work opens the door for future research into improved ASE filtering technologies, the development of Singular Perturbation Margin assessment tools, and more advanced mitigation techniques for signal synchronization issues in the presence of large filters.

2.2.4 List of publications

Works presented in this thesis have been submitted to scientific journals [73–75].

1. F. Laliberté and D. Saussié, "Relaxing the applicability conditions of Incremental Nonlinear Dynamics Inversion", Unpublished manuscript submitted to the Aerospace Science and Technology journal, 2025.

2. F. Laliberté and D. Saussié, "Robust tuning of the modified Incremental Nonlinear Dynamics Inversion for rigid aircraft", Unpublished manuscript submitted to the Journal of Guidance, Control and Dynamics, 2025.

3. F. Laliberté and D. Saussié, "Robust control for flexible aircraft using Incremental Nonlinear Dynamics Inversion with notch filters", Unpublished manuscript submitted to the Journal of Guidance, Control and Dynamics, 2025.

CHAPTER 3 ARTICLE 1: RELAXING THE APPLICABILITY CONDITIONS FOR INCREMENTAL NONLINEAR DYNAMICS INVERSION

Frédérick Laliberté, David Saussié
Published in: Aerospace Science and Technology
Submission date: March 30th 2025

Abstract
The Incremental Nonlinear Dynamic Inversion (INDI) technique is a promising approach for slowly varying systems equipped with fast actuators, sensors, and high-frequency sampling hardware. While numerous proofs of concept and analyses exist in the literature, they all rely on the same qualitative simplifying assumptions that define the technique's applicability. These include quasi-instantaneous actuators with identical bandwidths, synchronized delays that can be made arbitrarily small, and a sampling rate that can be made arbitrarily high. In practical applications, control designers are left with the critical question:"How fast is fast enough?" Currently, answering this requires a trial-and-error process, adjusting parameters until the desired performance is achieved, without any guarantee that the final solution will meet practical constraints. This paper introduces an alternative model of the open-loop system that characterizes the feedback linearization stage, unveiling an embedded Smith predictor and clarifying which simplifications can be disregarded to relax the technique's applicability conditions. Quantitative analyses and mitigation strategies are presented, along with an application example on an aircraft, where actuators typically exhibit non-identical dynamics.

3.1 Introduction

The Incremental Nonlinear Dynamic Inversion (INDI) control technique has gained increasing attention in recent decades, largely due to advancements in hardware that have enabled its practical implementation. As a modification of the well-known Nonlinear Dynamic Inversion (NDI), INDI aims to enhance robustness against model uncertainties by shifting reliance from accurate system modeling to precise sensor measurements.

The first work on a sensor-based modification of NDI was published by NASA in 1999 [59], where angular acceleration measurements were used to simplify the NDI control law. This simplification was based on the assumption that the transfer functions governing the rela-

tionship between angular acceleration and angular rate had a frequency content higher than the required response bandwidth, with no nonlinear saturation effects. Around the same time, [60] introduced a more generalized NDI approach, extending beyond the aircraft-specific formulation used in [59]. Combining the modification proposed in [59] with the generalized approach in [60] led to an early version of what is now known as INDI, where the bandwidth assumption was replaced by the similar time-scale separation principle.

This early sensor-based NDI approach was tested on the VAAC Harrier aircraft [61], yielding results that exceeded expectations and potentially sparking further research interest in the technique. However, hardware limitations soon became a major challenge. The Harrier's angular acceleration sensors were found to be inadequate, forcing researchers to estimate acceleration from angular rate measurements. These estimations, however, suffered from poor synchronization due to noise filtering, as well as issues related to control effectiveness and signal drift.

Years later, with improved hardware becoming available, research on INDI focused on expanding its applicability. Key topics such as the control effectiveness model [62], sensor delays and their synchronization [63], and the use of estimated signals instead of direct measurements [64] were explored, effectively addressing the minor issues identified in [61]. Additionally, [39] identified a common misconception in the literature and introduced a more rigorous approach that further relaxed applicability conditions by eliminating the need for the time-scale separation principle. Notably, a linear controller is often incorporated to enhance performance, typically including an integral action to prevent drift, which has now become a standard feature of what could be considered the "classic" INDI approach.

Building on these theoretical advancements, experimental flight tests were conducted to address real-world challenges. Most of these tests involved drones operating at various controller frequencies, such as 512 Hz [62], 250 Hz [65], and 2000 Hz [66][1]. Particularly noteworthy, [57] performed flight tests with a Cessna Citation in Dutch airspace using a controller running at 100 Hz while accounting for a range of real-world factors, providing a compelling demonstration of the INDI control technique in practical applications.

However, typical implementations impose constraints on controller frequency, sensor measurement delays, sampling rates, and actuator servo-control bandwidth. Practitioners are currently required to arbitrarily increase these parameters until simulations yield acceptable results. This heuristic approach is not always effective, leading to various modifications and mitigation strategies proposed in the literature, such as the weighted pseudo-inverse for con-

[1] This high frequency was used to enable the estimation of jerk and snap derivatives for trajectory tracking purposes.

trol effectiveness and allocation [67], considerations regarding angular accelerometers [68], and the challenge of finite actuator bandwidth [69].

There is a clear need for a more systematic approach to assessing the applicability of INDI for a given aircraft while considering practical constraints. This includes establishing concrete conditions under which INDI control operates as intended and identifying appropriate mitigation strategies when it does not—such as predictive filters to compensate for sensor delays. A fundamental question for any practitioner implementing INDI remains: "How fast is fast enough?", particularly when confronted with the assumptions of fast actuators and high sampling rates.

The aim of this paper is to establish applicability conditions through numerical tests and propose mitigation strategies to determine whether a given implementation—considering actuators, sensors, and sampling rate—is suitable for an INDI control law.

Section 3.2 presents the classical approach to implementing the INDI architecture. Then, Section 3.3 introduces an alternative formulation of the equations of motion that highlights the impact of neglected terms in the classical approach. The Feedback-Linearization stage of the INDI control law is characterized in Section 3.4 to better understand the effects of slow actuators, large delays, and poor sampling. To illustrate these effects, Section 3.5 provides an application example involving a rigid aircraft. Finally, Section 3.6 discusses the proposed tests and potential mitigation strategies.

3.2 Classic Incremental Nonlinear Dynamic Inversion

This section introduces the classic INDI approach, which begins with the linearization of the equations of motion and proceeds with the implementation of the INDI control law. As simplifications are made, qualitative assumptions must be clearly stated, as they will play a key role in the analyses presented in later sections.

Consider the following nonlinear system:

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

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^p$ is the input vector, and $\mathbf{f} : \mathbb{R}^{n+p} \to \mathbb{R}^n$ is a continuously differentiable vector function.

3.2.1 Linearization by Taylor series expansion

The system can be linearized around the time of the most recent measurement, t_0 [76]:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) + \mathcal{O}\Big((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2\Big) \tag{3.2}$$

with $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\mathbf{u}_0 = \mathbf{u}(t_0)$, $\mathbf{A}_0 = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\Big|_{t=t_0} \in \mathbb{R}^{n\times n}$, $\mathbf{G}_0 = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\Big|_{t=t_0} \in \mathbb{R}^{n\times p}$. If the acceleration and command signals are measured (or estimated as in [64, 77]), the vectors $\dot{\mathbf{x}}_0$ and $\mathbf{u}_0$ are known and can be used to achieve feedback linearization. In practice, $\mathbf{G}_0$ can typically be estimated with good precision[2] [57, 62], but obtaining an accurate model for $\mathbf{A}_0$ is generally challenging. However, it holds that:

$$\lim_{t\to t_0} \|\mathbf{A}_0(\mathbf{x} - \mathbf{x}_0)\| = 0$$
$$\lim_{t\to t_0} \Big\|\mathcal{O}\Big((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2\Big)\Big\| = 0$$

Thus, the following expression is obtained:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) + \epsilon_{l0} \approx \dot{\mathbf{x}}_0 + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) \tag{3.3}$$

where $\epsilon_{l0} = \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathcal{O}((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2)$ is the linearization error term, which becomes negligible with a sufficiently high sampling frequency.

3.2.2 INDI Control Law

The INDI control law is defined as:

$$\mathbf{u} = \hat{\mathbf{u}}_0 + \tilde{\mathbf{G}}_0^{\dagger}(\nu - \mathbf{H}_{sm}\hat{\mathbf{y}}_0) \tag{3.4}$$

where

- $\nu \in \mathbb{R}^p$ is the pseudo-command generated by a linear controller;
- $\hat{\mathbf{y}}_0 = \hat{\mathbf{y}}(t_0)$ is the measurement (or estimation) of $\mathbf{y} = \dot{\mathbf{x}}$ at t_0;
- $\hat{\mathbf{u}}_0 = \hat{\mathbf{u}}(t_0)$ is the measurement (or estimation) of $\mathbf{u}$ at t_0;
- $\mathbf{H}_{sm} \in \mathbb{R}^{p\times n}$ is a selector matrix selecting the measurement signals meant to track the pseudo-command ν;

[2] Alternatively, [78] proposed using the INDI concept for the very purpose of obtaining a reliable and precise Control Allocation scheme for $\mathbf{G}_0$.

- $\hat{\mathbf{G}}_0 = \hat{\mathbf{G}}(t_0)$ is the estimation of $\mathbf{G}$ at t_0;
- $\tilde{\mathbf{G}}_0 = \mathbf{H}_{sm}\hat{\mathbf{G}}_0 \in \mathbb{R}^{p\times p}$ is a square matrix approximation of the control effectiveness of $\mathbf{u}$ on $\mathbf{H}_{sm}\mathbf{y}$;
- $\mathbf{H}_{sm}\mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger} = (\mathbf{I} + \Delta\mathbf{G}) \approx \mathbf{I} \in \mathbb{R}^{p\times p}$ where $\tilde{\mathbf{G}}_0^{\dagger} \in \mathbb{R}^{p\times p}$ is the (modified[3] [79, 80]) on-board model (approximation) of $(\mathbf{H}_{sm}\mathbf{G}_0)^{\dagger}$, the Moore-Penrose pseudo-inverse of $\mathbf{H}_{sm}\mathbf{G}_0$, with error $\Delta\mathbf{G} \in \mathbb{R}^{p\times p}$ that is generally assumed to be negligible.

Adding the control law (3.4) to the system's dynamics equation (3.3) while neglecting measurement and modeling errors ($\hat{\mathbf{y}} - \mathbf{y} \approx \mathbf{0}$, $\hat{\mathbf{u}} - \mathbf{u} \approx \mathbf{0}$, $\Delta\mathbf{G} \approx \mathbf{0}$), yields

$$\zeta = \mathbf{H}_{sm}\mathbf{y} = \zeta_0 + \tilde{\mathbf{G}}_0(\mathbf{u} - \mathbf{u}_0) + \mathbf{H}_{sm}\epsilon_{l0} \tag{3.5}$$

$$\mathbf{u} = \mathbf{u}_0 + \tilde{\mathbf{G}}_0^{\dagger}(\nu - \zeta_0) \tag{3.6}$$

where ζ represents the measured output used for the feedback-linearization loop. The closed-loop system is then

$$\zeta \approx \nu + \mathbf{H}_{sm}\epsilon_{l0} \tag{3.7}$$

where feedback linearization is achieved for small enough ϵ_{l0}.

The INDI control architecture is presented in Fig. 3.1, where the feedback-linearization stage is highlighted. The block $\mathbf{H}_{act}(s)$ denotes the actuator system, the block ZOH_{ts} represents a Zero-Order Hold (ZOH) of sampling period T_s, sensors are represented by a sampler (with sampling period of T_s) and a synchronized delay $\mathbf{F}(z)$, and $\mathbf{C}(z)$ is the linear controller, which is shown as an "outer" loop in the diagram.

Several studies in the literature [39, 62, 64, 67] have demonstrated that the INDI technique is robust to modeling errors, provided that the linearization error ϵ_{l0} can be made arbitrarily small by increasing the sampling frequency. However, this claim relies on the assumption that the classical INDI conditions hold–specifically, that the actuator system behaves quasi-instantaneously, ensuring it reaches steady-state within a single sampling period, regardless of how small that period is. Given the objective of this paper, a more precise characterization of the linearization error ϵ_{l0} is necessary to determine the conditions under which it remains "sufficiently small" to uphold this robustness claim. The next section examines alternative equations of motion to better understand how actuators, sensors, and sampling rate influence the performance of the INDI control law.

[3] It is recommended to assign zeros on rows and columns of unused outputs and inputs respectively to prevent the least-square optimization done by the pseudo-inverse algorithm from using them. As explained in [79, 80], one can replace small values in $\mathbf{G}_0$ by zeros since those small values represent inefficient control effectiveness transfers that should not be used.

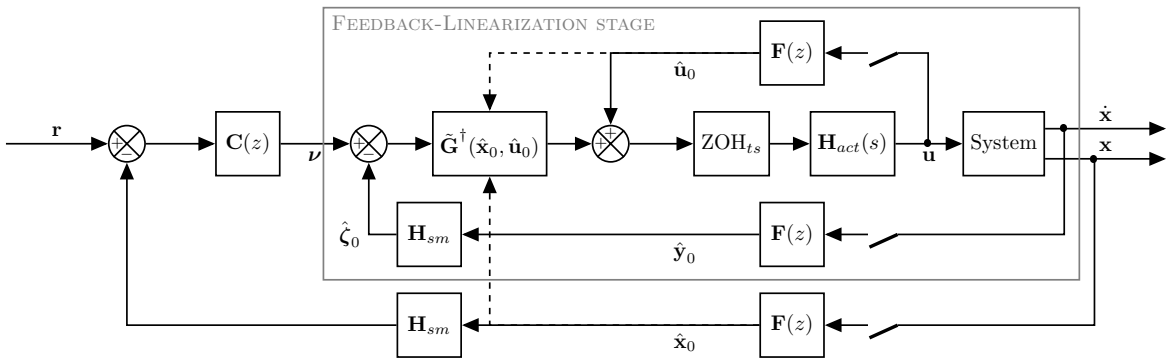

Figure 3.1 INDI Control architecture

3.3 Alternative solutions to the equations of motion

This section introduces an alternative formulation of the equations of motion, including both the open-loop system and the closed feedback-linearization loop, without relying on the simplifying assumptions under investigation.

3.3.1 Open-loop

3.3.1.1 Frozen-time Velocity-based linearization

The velocity-based linearization approach [69, 81] is a strong candidate, as state velocities are measured in an INDI context, and their dynamics are of particular interest.

Let $\mathbf{y} = \dot{\mathbf{x}}$ and take the time derivative of (3.1) :

$$\dot{\mathbf{y}}(t) = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) + \frac{\partial \mathbf{y}}{\partial \mathbf{u}}\dot{\mathbf{u}}(t) = \mathbf{A}(\mathbf{x},\mathbf{u})\mathbf{y}(t) + \mathbf{G}(\mathbf{x},\mathbf{u})\dot{\mathbf{u}}(t) \tag{3.8}$$

where $\mathbf{A}(\mathbf{x},\mathbf{u}) = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ and $\mathbf{G}(\mathbf{x},\mathbf{u}) = \frac{\partial \mathbf{y}}{\partial \mathbf{u}}$. A similar result can be achieved using Lie Derivatives as shown in [64]. A perturbed linearized system can then be obtained by evaluating $\mathbf{A}(\mathbf{x},\mathbf{u})$ and $\mathbf{G}(\mathbf{x},\mathbf{u})$ at instant t_0 ("frozen time"), yielding:

$$\dot{\mathbf{y}}(t) = \mathbf{A}_0\mathbf{y}(t) + \mathbf{G}_0\dot{\mathbf{u}}(t) + \epsilon_{vl} \tag{3.9}$$

where $\mathbf{A}_0 = \mathbf{A}(\mathbf{x}_0,\mathbf{u}_0)$ and $\mathbf{G}_0 = \mathbf{G}(\mathbf{x}_0,\mathbf{u}_0)$ are the same matrices found in (3.2), and where $\epsilon_{vl} = (\mathbf{A} - \mathbf{A}_0)\mathbf{y} + (\mathbf{G} - \mathbf{G}_0)\dot{\mathbf{u}}$ is the velocity-based linearization error. It has been shown in [81] that ϵ_{vl} is bounded over a given time interval, and that both the nonlinear system and the velocity-based linearization system are stable, if:

- Both $\mathbf{y}(t)$ and $\dot{\mathbf{u}}(t)$ are bounded over the interval;
- Both $\mathbf{A}(\mathbf{x}, \mathbf{u})$ and $\mathbf{G}(\mathbf{x}, \mathbf{u})$ are locally Lipschitz continuous over the interval;
- Both $|\mathbf{A}(\mathbf{x}_0, \mathbf{u}_0)|$ and $|\mathbf{G}(\mathbf{x}_0, \mathbf{u}_0)|$ are uniformly bounded;
- The eigenvalues of $\mathbf{A}(\mathbf{x}_0, \mathbf{u}_0)$ lie in the left-half plane.

Assuming that these conditions are met[4], the solution to (3.9) can be determined as follows:

$$\mathbf{y}(t) = e^{\mathbf{A}_0(t-t_0)}\mathbf{y}_0 + \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\mathbf{G}_0\dot{\mathbf{u}}(\tau)\mathrm{d}\tau + \epsilon_l \tag{3.10}$$

where $\epsilon_l = \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\epsilon_{vl}\mathrm{d}\tau$ is bounded (see [81]), and $e^{\mathbf{A}_0(t-t_0)}$ is the state transition matrix. The control $\mathbf{u}(t)$ is modeled as a staircase signal, which can be represented as a series of pulses [82]. Differentiating this signal results in a series of impulses occurring at intervals of T_s. Since the convolution integral of a series of impulses with a given function is straightforward to evaluate, (3.10) simplifies to:

$$\begin{aligned}\mathbf{y}(t) &= e^{\mathbf{A}_0(t-t_0)}\mathbf{y}(t_0) + \sum_{j=1}^{D} e^{\mathbf{A}_0((D-j)T_s)}\mathbf{G}_0(\mathbf{u}[j_0+j] - \mathbf{u}[j_0+j-1]) + \epsilon_l(t) && (3.11)\\ &= e^{\mathbf{A}_0 DT_s}\mathbf{y}(t-DT_s) + \mathbf{G}_0\mathbf{u}(t) - e^{\mathbf{A}_0(D-1)T_s}\mathbf{G}_0\mathbf{u}(t-DT_s)\\ &\quad + \sum_{k=1}^{D-1} e^{\mathbf{A}_0(k-1)T_s}\Big(e^{\mathbf{A}_0 T_s} - \mathbf{I}\Big)\mathbf{G}_0\mathbf{u}(t-kT_s) + \epsilon_l(t) && (3.12)\end{aligned}$$

where $D = \lceil \frac{t-t_0}{T_s} \rceil$ is the number of samples contained in $t - t_0$ (rounded up) and where $k = D - j$.

Note that in the case of "minimal delay", where $t - t_0 = T_s$ ($D = 1$), the above reduces to:

$$\mathbf{y}(t) = e^{\mathbf{A}_0(t-t_0)}\mathbf{y}_0 + \mathbf{G}_0(\mathbf{u}(t) - \mathbf{u}(t_0)) + \epsilon_l(t) \tag{3.13}$$

with $\lim_{t\to t_0} e^{\mathbf{A}_0(t-t_0)} = \mathbf{I}$ and $\lim_{t\to t_0} \epsilon_l(t) = \mathbf{0}$, which matches (3.3).

Considering that $e^{\mathbf{A}_0(t-t_0)}\mathbf{y}_0 = \mathbf{I}\mathbf{y}_0 + \mathbf{A}_0(t-t_0)\mathbf{y}_0 + \mathcal{O}((t-t_0)^2) \approx \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x}-\mathbf{x}_0) + \mathcal{O}((x-x_0)^2)$, a comparison between (3.3) and (3.13) illustrates that $\epsilon_l(t) = \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\epsilon_{vl}\mathrm{d}\tau$ is smaller than $\epsilon_{l0} = \mathbf{A}_0(\mathbf{x}-\mathbf{x}_0) + \mathcal{O}((\mathbf{x}-\mathbf{x}_0)^2, (\mathbf{u}-\mathbf{u}_0)^2)$. This alternative expression, which will be referred to as the "frozen-time velocity-based" (FTVB) model, is thus better suited for

[4]In practice, the eigenvalues of $\mathbf{A}(\mathbf{x}_0, \mathbf{u}_0)$ may not all lie in the left-half plane in the open-loop system but are expected to do so once all control loops are closed.

analysis as it provides greater accuracy than the classical equation by incorporating known values. The only exception is the smaller linearization error, whose bound is dependent on the Lipschitz constants of the nonlinear system.

3.3.1.2 Actuator dynamics - Augmented plant

In cases where actuator dynamics cannot be ignored, the plant is augmented, and the same approach is applied using the actuator command signal $\mathbf{u}_c(t)$ and $\xi = [\mathbf{y}^\top, \dot{\mathbf{u}}^\top]^\top$ as the input and output of the augmented system.

$$\begin{aligned}
\dot{\xi} = \begin{bmatrix} \dot{\mathbf{y}} \\ \ddot{\mathbf{u}} \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_0 & \mathbf{G}_0 \\ \mathbf{0} & \mathbf{A}_{act} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \dot{\mathbf{u}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{act} \end{bmatrix} \dot{\mathbf{u}}_c + \begin{bmatrix} \epsilon_{vl} \\ \mathbf{0} \end{bmatrix} = \mathbf{A}_{aug}\xi + \mathbf{B}_{aug}\dot{\mathbf{u}}_c + \epsilon_{aug,vl} \\
\Rightarrow \xi(t) &= e^{\mathbf{A}_{aug}(t-t_0)}\xi(t_0) + \mathbf{B}_{aug}\mathbf{u}_c(t) - e^{\mathbf{A}_{aug}(t-t_0-T_s)}\mathbf{B}_{aug}\mathbf{u}_c(t_0) \\
&\quad + \sum_{j=1}^{D-1} e^{\mathbf{A}_{aug}((D-j-1)T_s)}\Big(e^{\mathbf{A}_{aug}T_s} - \mathbf{I}\Big)\mathbf{B}_{aug}\mathbf{u}_c(t_0 + jT_s) + \epsilon_{aug,l}(t)
\end{aligned} \tag{3.14}$$

where

$$e^{\mathbf{A}_{aug}x} = \sum_{n=0}^{\infty} \frac{x^n}{n!}\mathbf{A}_{aug}^n = \sum_{n=0}^{\infty} \frac{x^n}{n!} \begin{bmatrix} \mathbf{A}_0^n & \sum_{k=0}^{n} \mathbf{A}_0^{n-k}\mathbf{G}_0\mathbf{A}_{act}^k \\ \mathbf{0} & \mathbf{A}_{act}^n \end{bmatrix} = \begin{bmatrix} e^{\mathbf{A}_0 x} & \int_0^x e^{\mathbf{A}_0(x-\tau)}\mathbf{G}_0 e^{\mathbf{A}_{act}\tau}\mathrm{d}\tau \\ \mathbf{0} & e^{\mathbf{A}_{act}x} \end{bmatrix}$$

and where

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{x^n}{n!} \sum_{k=0}^{n} \mathbf{A}_0^{n-k}\mathbf{G}_0\mathbf{A}_{act}^k &= \sum_{n=0}^{\infty}\sum_{k=0}^{n} \frac{x^{n-k}}{(n-k)!}\mathbf{A}_0^{n-k}\mathbf{G}_0\frac{x^k}{k!}\mathbf{A}_{act}^k = \left(\sum_{n=0}^{\infty}\frac{x^n}{n!}\mathbf{A}_0^n\right)\left(\mathbf{G}_0\sum_{n=0}^{\infty}\frac{x^n}{n!}\mathbf{A}_{act}^n\right) \\
&= \left(e^{\mathbf{A}_0 x}\right) \circledast \left(\mathbf{G}_0 e^{\mathbf{A}_{act}x}\right) = \int_0^x e^{\mathbf{A}_0(x-\tau)}\mathbf{G}_0 e^{\mathbf{A}_{act}\tau}\mathrm{d}\tau
\end{aligned}$$

in virtue of the Cauchy product's relation to convolution.

The previous open-loop equation is recovered when $e^{\mathbf{A}_{act}T_s} \approx \mathbf{0}$, which represents the case where actuator dynamics can be neglected. Moreover, this approach can also be used to derive a linear system around trim conditions or to generate a linear time-varying (LTV) model. However, the conclusions regarding ϵ_l do not directly apply in these cases.

3.3.1.3 Exact solution (LTV)

The exact solution to the nonlinear equations of motion (3.8) can be computed as a time-varying linear (LTV) system:

$$\mathbf{y}(t) = \mathbf{\Phi}(t, t_0)\mathbf{y}_0 + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{G}(\tau)\dot{\mathbf{u}}(\tau)\mathrm{d}\tau \tag{3.15}$$

where the state transition matrix $\mathbf{\Phi}(t, t_0) = e^{\int_{t_0}^{t} \mathbf{A}(\tau)\mathrm{d}\tau}$ if and only if $\mathbf{A}(t)$ commutes with its own integral.

The proposed alternative equations and the classical INDI open-loop equations can be compared with this exact solution to identify the terms that were neglected or approximated in simplifying the LTV system to an LTI one.

First, if $\mathbf{A}(t) \approx \mathbf{A}(t_0) = \mathbf{A}_0$, then $\mathbf{\Phi}(t, t_0) \approx e^{\mathbf{A}_0(t-t_0)}$. Furthermore, if $\int_{t_0}^{t} \mathbf{A}(\tau)\mathrm{d}\tau \approx \mathbf{A}_0(t - t_0) \approx \mathbf{0}$, then $\mathbf{\Phi}(t, t_0) \approx \mathbf{I}$. Finally, note that $\mathbf{G}(t) \approx \mathbf{G}(t_0) = \mathbf{G}_0$ is required for implementing the control law. These constitute the approximations made by both classical INDI and the alternative approaches to simplify the exact solution. Note that the classical INDI approximation is less accurate than the proposed FTVB alternative and only matches when $e^{\mathbf{A}_0(t-t_0)} \approx \mathbf{I}$. These conditions are in addition to those previously mentioned, such as fast and uniform actuator bandwidth.

3.3.2 Closed-loop

3.3.2.1 Frozen time Velocity-based linearization

Combining the INDI control law (3.4) with the FTVB model, while neglecting measurement and modeling errors ($\hat{\mathbf{y}} - \mathbf{y} \approx \mathbf{0}$, $\hat{\mathbf{u}} - \mathbf{u} \approx \mathbf{0}$, $\Delta\mathbf{G} \approx \mathbf{0}$), yields:

$$\mathbf{y}(t) \approx \mathbf{M}_0^D\mathbf{y}(t_0) + \mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger}(\nu(t) - \mathbf{H}_{sm}\mathbf{y}(t_0)) + \delta_m(t) + \epsilon_l(t) \tag{3.16}$$

$$\mathbf{y}(t) \approx (\mathbf{M}_0^D - \mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm})\mathbf{y}(t_0) + \mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger}\nu(t) + \delta_m(t) + \epsilon_l(t) \tag{3.17}$$

$$\Rightarrow \zeta(t) \approx \mathbf{H}_{sm}(\mathbf{M}_0^D - \mathbf{I})\mathbf{y}(t_0) + \nu(t) + \mathbf{H}_{sm}\delta_m(t) + \mathbf{H}_{sm}\epsilon_l(t) \tag{3.18}$$

where $\mathbf{M}_0 = e^{\mathbf{A}_0 T_s}$ is used for simplicity and ease of reading, and where

$$\delta_m(t) = \sum_{k=1}^{D-1} \mathbf{M}_0^{k-1}(\mathbf{M}_0 - \mathbf{I})\mathbf{G}_0\mathbf{u}(t - kT_s)$$

is deemed negligible if $\mathbf{M}_0 - \mathbf{I} \approx \mathbf{0}$.

To better discern the behaviour of the closed loop feedback-linearized system, equations (3.11) and (3.4) are discretized and the Z-transform is applied to obtain:

$$\zeta(z) = \mathbf{H}_{sm}\Big(\mathbf{I} - \Big(\mathbf{M}_0^D - \mathbf{Q}(z)\mathbf{H}_{sm}\Big)z^{-D}\Big)^{-1}(\mathbf{Q}(z)\nu(z) + \epsilon_l(z)) \tag{3.19}$$

with $\mathbf{Q}(z) = \frac{1-z^{-1}}{1-z^{-D}}\Big(\sum_{j=1}^{D}\mathbf{M}_0^{D-j}z^{-D+j}\Big)\mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger} = \frac{1-z^{-1}}{1-z^{-D}}\Big(\mathbf{I} - \mathbf{M}_0^D z^{-D}\Big)(\mathbf{I} - \mathbf{M}_0 z^{-1})^{-1}\mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger}$.
If $\mathbf{M}_0 \approx \mathbf{M}_0^D \approx \mathbf{I}$, then $\mathbf{Q}(z) \approx \mathbf{G}\tilde{\mathbf{G}}_0^{\dagger}$ and thus

$$\zeta(z) \approx \mathbf{H}_{sm}\Big(\mathbf{I} - \Big(\mathbf{I} - \mathbf{G}\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}\Big)z^{-D}\Big)^{-1}\Big(\mathbf{G}\tilde{\mathbf{G}}_0^{\dagger}\nu(z) + \epsilon_l(z)\Big) \tag{3.20}$$

$$\approx \Big(\mathbf{H}_{sm} + \Big(\mathbf{H}_{sm} - (\mathbf{H}_{sm}\mathbf{G}\tilde{\mathbf{G}}_0^{\dagger})\mathbf{H}_{sm}\Big)z^{-D}\Big(\mathbf{I} - \Big(\mathbf{I} - \mathbf{G}\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}\Big)z^{-D}\Big)^{-1}\Big)\Big(\mathbf{G}\tilde{\mathbf{G}}_0^{\dagger}\nu(z) + \epsilon_l(z)\Big) \tag{3.21}$$

$$\approx \nu(z) + \mathbf{H}_{sm}\epsilon_l(z) \tag{3.22}$$

where the matrix identity $(\mathbf{I} + \mathbf{P})^{-1} = \mathbf{I} - \mathbf{P}(\mathbf{I} + \mathbf{P})^{-1}$ was used.

Equation (3.22) matches the classical INDI closed-loop equation (3.7). However, unlike the classical approach, this alternative method provides explicit expressions to compute the feedback-linearized closed-loop system without approximation, while accounting for various uncertainties. Furthermore, under the same conditions, one can find that

$$\begin{aligned}\epsilon_{l0}[k] &= \Big(\mathbf{M}_0^D - \mathbf{I}\Big)\mathbf{y}[k-D] - \Big(\mathbf{M}_0^{D-1} - \mathbf{I}\Big)\mathbf{G}_0\mathbf{u}[k-D] \\ &\quad + \sum_{j=1}^{D-1}\mathbf{M}_0^{D-j-1}(\mathbf{M}_0 - \mathbf{I})\mathbf{G}_0\mathbf{u}[k-D+j] + \epsilon_l[k] \\ &\approx \epsilon_l[k]\end{aligned} \tag{3.23}$$

This highlights the core difference between the classical equation and the proposed FTVB alternative. It also demonstrates that the classical feedback-linearization error includes multiplicative errors, which are not rejected in the same way as additive errors. This distinction is important for the robustness proof of the INDI control law found in the literature [6]. Furthermore, this multiplicative error helps explain the drift observed in [61], as its control law entirely neglects $t - t_0$ and, consequently, $\mathbf{M}_0$, similar to the classical approach that inspired it.

3.3.2.2 Actuator dynamics - Augmented plant

The same approach is used for the augmented plant, with the distinction that the output feedback signals include $\mathbf{u}_0$:

$$
\begin{aligned}
\xi(t) &= \mathbf{M}_{aug}^{D}\xi(t_0) + \mathbf{B}_{aug}\mathbf{u}_c(t) - \mathbf{M}_{aug}^{D-1}\mathbf{B}_{aug}\mathbf{u}_c(t_0) \\
&\quad + \sum_{k=1}^{D-1} \mathbf{M}_{aug}^{k-1}(\mathbf{M}_{aug} - \mathbf{I})\mathbf{B}_{aug}\mathbf{u}_c(t - kT_s) + \epsilon_{aug}(t) \\
\zeta(t) &= \begin{bmatrix} \mathbf{H}_{sm} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \xi(t) \\
\mathbf{u}_c(t) &= \tilde{\mathbf{G}}_0^{\dagger}\nu(t) - \begin{bmatrix} \tilde{\mathbf{G}}_0^{\dagger} & -\mathbf{I} \end{bmatrix} \hat{\zeta}(t_0) \\
\Rightarrow \zeta(z) &= \begin{bmatrix} \mathbf{H}_{sm} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \Big(\mathbf{I} - \Big(\mathbf{M}_{aug}^{D} - \tilde{\mathbf{Q}}_{aug}(z)\begin{bmatrix} \tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm} & -\mathbf{I} \end{bmatrix}\Big) z^{-D}\Big)^{-1} \Big\{\tilde{\mathbf{Q}}_{aug}(z)\tilde{\mathbf{G}}_0^{\dagger}\nu(z) + \epsilon_{aug}(z)\Big\}
\end{aligned}
\tag{3.24}
$$

where $\mathbf{M}_{aug} = e^{\mathbf{A}_{aug}T_s}$, and where $\tilde{\mathbf{Q}}_{aug}(z) = (1 - z^{-1})\Big(\mathbf{I} - \mathbf{M}_{aug}^{D}z^{-D}\Big)(\mathbf{I} - \mathbf{M}_{aug}z^{-1})^{-1}\mathbf{B}_{aug}$ differs from $\mathbf{Q}(z)$ due to how the $\mathbf{u}_0$ feedback was treated.

Note that if $\mathbf{u}(t_0) = \mathbf{u}_c(t_0)$, then the actuator reaches steady state within DT_s seconds and that the above equations reduces to 3.19; in other words, the actuator dynamics are negligible and the augmented approach is not required.

3.4 Characterization of the INDI Feedback-Linearization Architecture

The equations of motion for both the open and closed-loop systems, derived in the previous section, were based on several assumptions that led to the neglect of certain terms. While the ideal closed-loop feedback-linearized system behaves as a pure integrator, the actual closed-loop behavior of the Feedback-Linearization stage can differ significantly. This discrepancy motivates the use of robust tuning methods for the linear controller stage of the INDI control law, challenging the notion that the INDI control scheme is inherently robust to modeling errors.

To better understand the limitations of the INDI control law's applicability, this section will examine the architecture of the Feedback-Linearization stage. The goal is to identify the conditions under which the assumptions required for the INDI control law to provide its advantages hold true. The section will begin with the ideal INDI closed-loop system and gradually reintroduce the neglected terms to highlight their impact on the closed-loop response.

3.4.1 Ideal closed-loop model

As discussed in Sections 3.2 and 3.3, the linearization errors ϵ_{l0} and ϵ_l can be made exactly zero if the synchronized delay $t - t_0 = 0$, implying no delay or discretization. By neglecting $\mathbf{A}_0$ and all measurement and modeling errors, the Feedback-Linearization stage simplifies into the block-diagram shown in Fig.3.2. Interestingly, the actuator increment stage can be simplified to

$$\mathbf{K}(s) = (\mathbf{I} - \mathbf{H}_{act}(s))^{-1} s\mathbf{H}_{act}(s) \Leftrightarrow \mathbf{H}_{act}(s) = (s\mathbf{I} + \mathbf{K}(s))^{-1}\mathbf{K}(s) \tag{3.25}$$

as shown in Fig. 3.2.

For first order actuators, $\mathbf{K}(s) = \mathbf{K}$ is a constant diagonal matrix. If the actuators have the same bandwidth, then $\mathbf{K}(s) = \Omega\mathbf{I}$ where Ω is the uniform actuator bandwidth. As such, $\mathbf{K}(s)$ will be referred to as the "actuator bandwidth", and higher order actuators would be treated as having variable frequency-dependent bandwidths.

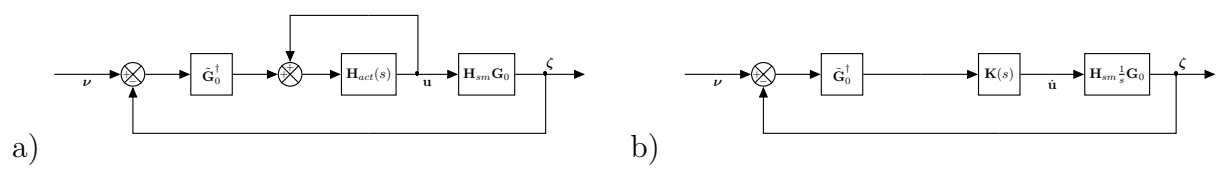

Figure 3.2 Idealized Feedback-Linearisation stage: a) Unsimplified and b) simplified actuator loop

Simplifying the block-diagram of Fig. 3.2 yields the ideal transfer function:

$$\frac{\zeta(s)}{\nu(s)} = \mathbf{H}_{sm}\mathbf{T}_0(s) = \mathbf{H}_{sm}\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger} \approx (s\mathbf{I} + \mathbf{K}(s))^{-1}\mathbf{K}(s) = \mathbf{H}_{act}(s) \tag{3.26}$$

where the equality is met if $\tilde{\mathbf{G}}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger} = \mathbf{K}(s)$, which is only the case for identical actuators.

This will motivate a mitigation technique for non-identical actuators, where the modification $\bar{\mathbf{G}}_0^{\dagger} = \Big(\tilde{\mathbf{G}}_0\mathbf{K}(0)\Big)^{\dagger}\mathbf{K}(0)$ is used to improve the (ideal) transfer decoupling. Unfortunately, simplifications offered in this section no longer apply directly as this change "breaks" the relation $\mathbf{H}_{sm}\mathbf{G}_0\tilde{\mathbf{G}}_0^{\dagger} = \mathbf{I}$ (i.e., $\Delta\mathbf{G} \not\approx \mathbf{0}$).

Since the ideal response matches the actuator response, increasing the actuator bandwidth directly decreases settling and rise times as is expected under the INDI control law.

3.4.2 Continuous delay-free closed-loop model

Building on the previous section, $\mathbf{A}_0$ is added back into the system as shown in Fig. 3.3.

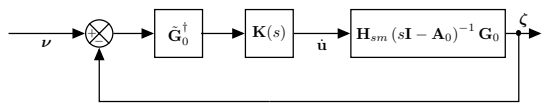

Figure 3.3 Continuous delay-free closed-loop

The closed-loop transfer function becomes:

$$\frac{\zeta(s)}{\nu(s)} = \mathbf{H}_{sm}\mathbf{T}_1(s) = \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \tag{3.27}$$

This system can be approximated by the ideal system of Fig. 3.2, if the actuator bandwidth alone is sufficient to dominate the system's response. To better illustrate how $\mathbf{A}_0$ impacts the system's response, consider the difference between (3.26) and (3.27):

$$\begin{aligned}\mathbf{H}_{sm}\mathbf{T}_1(s) - \mathbf{H}_{sm}\mathbf{T}_0(s) &= \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\mathbf{A}_0\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1} \\ &\quad \cdot \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \qquad (3.28)\\ \Rightarrow \mathbf{H}_{sm}\mathbf{T}_1(s) &= \mathbf{H}_{sm}(\mathbf{I} + \boldsymbol{\Delta}_a(s))\mathbf{T}_0(s), \qquad (3.29)\end{aligned}$$

with $\boldsymbol{\Delta}_a(s) = \Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\mathbf{A}_0$ and where the matrix identity $(\mathbf{A}-\mathbf{B})^{-1} = \mathbf{A}^{-1} + (\mathbf{A}-\mathbf{B})^{-1}\mathbf{B}\mathbf{A}^{-1}$ was used[5].

Equation (3.29) directly shows how the plant dynamics $\mathbf{A}_0$ drives a multiplicative (output) uncertainty $\boldsymbol{\Delta}_a$ on the ideal response, and that $\boldsymbol{\Delta}_a(s) \to \mathbf{0}$ as $\mathbf{K}(s) \to \infty\mathbf{I}$.

Note that this delay-free time response should meet all requirements (settling time, steady-state error, stability margins, etc.), as adding delays will deteriorate performance.

3.4.3 Delayed continuous closed-loop model

The sensor delay is now added to the continuous model and the effective controller of this closed-loop system must now be modified. Let $\mathbf{C}_a(s) = \mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger$ be the ideal controller, $\mathbf{P}(s) = \mathbf{H}_{sm}(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0$ be the plant, and $\mathbf{F}(s)$ be the delay term. A new controller

[5]This matrix identity can only be used if the first matrix is invertible ($\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}$ in this case), while the second matrix ($\mathbf{A}_0$ in this case) can be singular. Note that the restrictions already stated for $\mathbf{K}(s)$ ensures the applicability of this matrix identity.

$\mathbf{C}_b(s)$, that will cause the delayed system to behave like the ideal delay-free one, is now sought after. Using Youla parametrization [83, 84], such a controller would have the form

$$\mathbf{C}_b(s) = (\mathbf{I} - \mathbf{Q}_B(s)\mathbf{F}(s)\mathbf{P}(s))^{-1}\mathbf{Q}_B(s), \text{ with } \mathbf{Q}_B(s) \text{ stable and proper}$$

where one can naturally choose $\mathbf{Q}_B(s) = (\mathbf{I} + \mathbf{C}_a(s)\mathbf{P}(s))^{-1}\mathbf{C}_a(s)$ to yield

$$\mathbf{C}_b(s) = \Big(\mathbf{I} + \mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger(\mathbf{I} - \mathbf{F}(s))\mathbf{P}(s)\Big)^{-1}\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \tag{3.30}$$

which corresponds to a Smith predictor [85, 86].

Unfortunately, this control law is difficult to implement due to the requirement for an on-board model of $\mathbf{P}(s)$. As a result, the ideal model $\hat{\mathbf{P}}(s) = \frac{1}{s}\tilde{\mathbf{G}}_0$ is used as an approximation, leading to the block diagram shown in Fig. 3.4. Note that the delay term $\mathbf{F}(s)$ appearing in $\mathbf{C}_b(s)$ must be synchronized with the sensor delay block. The implemented controller is therefore:

$$\mathbf{C}_c(s) = \Big(\mathbf{I} + \mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger(\mathbf{I} - \hat{\mathbf{F}}(s))\hat{\mathbf{P}}(s)\Big)^{-1}\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \tag{3.31}$$

$$= \Big(\mathbf{I} + \frac{1}{s}(\mathbf{I} - \hat{\mathbf{F}}(s))\mathbf{K}(s)\Big)^{-1}\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \tag{3.32}$$

where $\hat{\mathbf{F}}(s)$ is the actuator delay supposed to be synchronized with the sensor delay ($\hat{\mathbf{F}}(s) = \mathbf{F}(s)$) and can commute with other matrices if all delays are identical ($\mathbf{F}(s) = e^{-s\tau}\mathbf{I}$).

Before analyzing this system further, note that (3.32) only holds if the relation $\tilde{\mathbf{G}}_0^\dagger\tilde{\mathbf{G}}_0 = \mathbf{I}$ holds, in addition to delays being identical across axes, in which case the diagram can be simplified into the one from Fig. 3.5(a). This diagram is equivalent to the one from Fig. 3.5(b), which matches the INDI Feedback-Linearization stage presented in Fig. 3.1 when discretized.

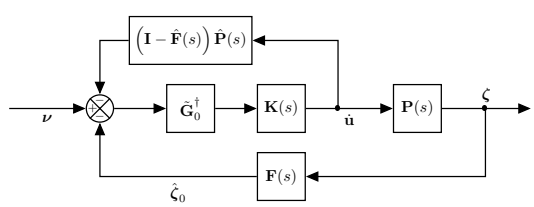

Figure 3.4 Continuous closed-loop

To better understand the effect of adding delays to the system and approximating $\mathbf{P}(s) \approx \hat{\mathbf{P}}(s)$ for implementation, consider the consequences of the added "actuator feedback" on

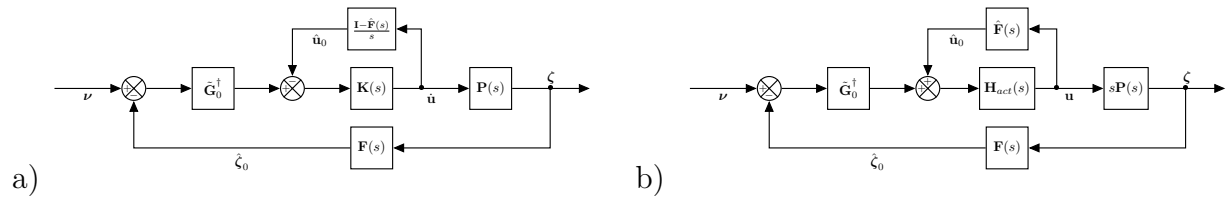

Figure 3.5 Simplified continuous closed-loop using a) $\mathbf{K}(s)$, and b) $\mathbf{H}_{act}(s)$

the system's response when compared to the delay-free version; the "effective bandwidth" becomes

$$\tilde{\mathbf{K}}(s) = \Big(\mathbf{I} + \frac{1}{s}(\mathbf{I} - \mathbf{F})\mathbf{K}(s)\Big)^{-1}\mathbf{K}(s) \tag{3.33}$$

where the term $\frac{1}{s}(\mathbf{I} - \mathbf{F}(s)) = \mathbf{I}\tau\Big(\frac{1-e^{-s\tau}}{s\tau}\Big) = \mathbf{I}\tau\mathrm{ZOH}_\tau(s)$ behaves like a ZOH of sampling period τ. This means the effective bandwidth matches $\mathbf{K}(s)$ at high frequency, and $\tilde{\mathbf{K}}(0) = (\mathbf{I} + \tau\mathbf{K}(0))^{-1}\mathbf{K}(0) = \Big(\frac{1}{\tau}\mathbf{I} + \mathbf{K}(0)\Big)^{-1}\frac{1}{\tau}\mathbf{K}(0) = (\tau\mathbf{I} + \mathbf{K}^{-1}(0))^{-1}$ after the τ delay has passed. As such, $\tilde{\mathbf{K}}(0)$ will always have lower bandwidth than $\mathbf{K}(0)$ or $\frac{1}{\tau}\mathbf{I}$, but still need to dominate the plant dynamics in a closed loop. This implies that both the actuator bandwidth and synchronized delay must be taken into account when assessing applicability.

This allows the delayed closed-loop transfer function to be written as:

$$\begin{aligned}
\frac{\zeta(s)}{\nu(s)} &= \mathbf{H}_{sm}\mathbf{T}_2(s) = (\mathbf{I} + \mathbf{P}(s)\mathbf{C}_c(s)\mathbf{F}(s))^{-1}\mathbf{P}(s)\mathbf{C}_c(s) \\
&= \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{F}\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger \\
&= \mathbf{P}(s)\mathbf{C}_a(s)\Big(\mathbf{I} + (\mathbf{I} - \hat{\mathbf{F}}(s))\hat{\mathbf{P}}(s)\mathbf{C}_a(s) + \mathbf{F}(s)\mathbf{P}(s)\mathbf{C}_a(s)\Big)^{-1} \\
&= \mathbf{P}(s)\mathbf{C}_a(s)\Big(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s) + \Big\{(\mathbf{I} - \hat{\mathbf{F}}(s))\hat{\mathbf{P}}(s) - (\mathbf{I} - \mathbf{F}(s))\mathbf{P}(s)\Big\}\mathbf{C}_a(s)\Big)^{-1} \\
&= \mathbf{P}(s)\mathbf{C}_a(s)(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s) + \mathbf{E}_i(s)\mathbf{C}_a(s))^{-1} \\
&= \mathbf{P}(s)\mathbf{C}_a(s)(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s))^{-1}\Big(\mathbf{I} + \mathbf{E}_i(s)\mathbf{C}_a(s)(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s) + \mathbf{E}_i(s)\mathbf{C}_a(s))^{-1}\Big) \\
&= \mathbf{H}_{sm}\mathbf{T_1}(s)(\mathbf{I} + \boldsymbol{\Delta}_d(s)), \qquad \text{with } \boldsymbol{\Delta}_d(s) = \mathbf{E}_i(s)\mathbf{C}_a(s)(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s) + \mathbf{E}_i(s)\mathbf{C}_a(s))^{-1}
\end{aligned} \tag{3.34}$$

where the term $\mathbf{E}_i(s) = \Big\{(\mathbf{I} - \hat{\mathbf{F}}(s))\hat{\mathbf{P}}(s) - (\mathbf{I} - \mathbf{F}(s))\mathbf{P}(s)\Big\}$ is caused by the implementation error (thus vanishes when $\mathbf{C}_c(s) = \mathbf{C}_b(s)$). It captures the model and delay mismatches that deteriorate the Smith predictor's performance and robustness.

The closed-loop transfer function can be simplified further when delays are synchronized

($\hat{\mathbf{F}}(s) = \mathbf{F}(s)$):

$$\begin{aligned}\mathbf{H}_{sm}\mathbf{T}_2(s) &= \mathbf{P}(s)\mathbf{C}_a(s)\Big(\mathbf{I} + \mathbf{P}(s)\mathbf{C}_a(s) + \frac{1}{s}(\mathbf{I} - \mathbf{F}(s))\big\{\tilde{\mathbf{G}}_0 - s\mathbf{P}(s)\big\}\mathbf{C}_a(s)\Big)^{-1} \\ &= \mathbf{H}_{sm}\mathbf{T}_1(s)(\mathbf{I} + \boldsymbol{\Delta}_d(s)) \\ &= \mathbf{H}_{sm}(\mathbf{I} + \boldsymbol{\Delta}_a(s))\mathbf{T}_0(s)(\mathbf{I} + \boldsymbol{\Delta}_d(s)) \end{aligned} \tag{3.35}$$

with

$$\begin{aligned}\boldsymbol{\Delta}_d(s) &= \tau\mathrm{ZOH}_\tau(s)\mathbf{H}_{sm}\mathbf{A}_0(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger \\ &\quad \cdot \Big(\mathbf{I} + \mathbf{H}_{sm}(\mathbf{I} - \tau\mathrm{ZOH}_\tau(s)\mathbf{A}_0)(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\Big)^{-1} \\ &= \tau\mathrm{ZOH}_\tau(s)\mathbf{H}_{sm}\mathbf{A}_0\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}(\mathbf{I} - \tau\mathrm{ZOH}_\tau(s)\mathbf{A}_0)\Big)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\end{aligned}$$

acting as a multiplicative (input) uncertainty, further deteriorating the closed-loop performance. Note that $\boldsymbol{\Delta}_d(s) \to \mathbf{0}$ when the delays are synchronized and made arbitrarily small ($\tau \to \mathbf{0}$).

Furthermore, discretization and other errors are expected to reduce performance and, as such, this continuous delayed system should meet all requirements as a condition for the INDI guarantees to hold for the true discrete implementation. It is recommended to use a similar approach to the one used to apply a Smith predictor to an integrating process [86].

3.4.4 Discrete closed-loop model

The final step in this investigation is to discretize the delayed system in order to recover the equations presented earlier.

It is important to note that the sampling period should be at most equal to the synchronized delay ($T_s \leq t - t_0$) to avoid increasing it. Additionally, the sampling period should be selected as an integer fraction of the delay period[6]. If the sampling period exceeds the sensor delay, the synchronized delay becomes the sampling period (resulting in a "minimal delay" ($D = 1$) that is larger than the sensor delay), and the previously presented continuous delay closed-loop must be re-evaluated.

The current objective is to avoid approximations during discretization. Since the controller's ZOH is located before the signal goes through the actuator system, the system is best discretized by using the ZOH approach (Backward Euler) on the augmented system[7]. If the

[6]This means minimizing $(t - t_0) - DT_s$ where $D = \frac{t-t_0}{T_s}$ is rounded up.

[7]This can be done simply in MATLAB using the following command `c2d([P*s*H_act;H_act],T_s)` where `P=ss(A_0,G_0,H_sm,0)` is the plant model, `H_act` is the actuator system, `s=tf('s')` is Laplace's variable

actuator bandwidth is sufficiently large, its steady-state would be reached within T_s seconds and would allow the actuator dynamics to be neglected[8] ($\mathbf{H}_{act}(z) = \mathbf{I}$) and use Eq.(3.19) directly.

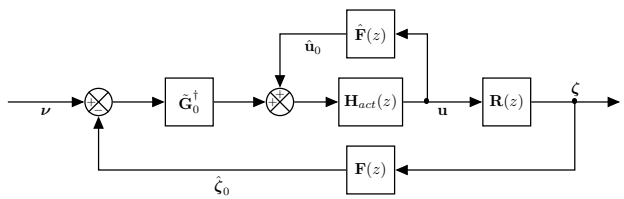

Figure 3.6 Discrete closed-loop

The discretized plant can thus be approximated as:

$$\mathbf{R}(z) = \mathbf{H}_{sm}(z\mathbf{I} - \mathbf{M}_0)^{-1}\mathbf{N}_0\frac{z-1}{T_s} = \mathcal{Z}\left\{(s\mathbf{P}(s))\frac{1}{s}\right\}\frac{1-z^{-1}}{T_s} = \mathcal{Z}\{(s\mathbf{P}(s))\mathrm{ZOH}_{ts}(s)\} \quad (3.36)$$

or

$$\mathbf{R}(z) = \mathbf{H}_{sm}\mathbf{A}_0(z\mathbf{I} - \mathbf{M}_0)^{-1}\mathbf{N}_0 + \tilde{\mathbf{G}}_0 = \mathcal{Z}\{(s\mathbf{P}(s))\mathrm{ZOH}_{ts}(s)\} \quad (3.37)$$

where $\mathbf{M}_0$ and $\mathbf{N}_0$ may be computed from the matrix exponential of $[\mathbf{A}_0, \mathbf{G}_0; \mathbf{0}, \mathbf{0}]T_S$ and where $s\mathbf{P}(s) = \mathbf{H}_{sm}\mathbf{A}_0(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0 + \tilde{\mathbf{G}}_0$. This equation is exact for the augmented plant (in which case the $\mathbf{H}_{act}(z)$ block is removed from the diagram as it is included within the augmented plant). Furthermore, this entire expression reduces to $\tilde{\mathbf{G}}_0$ when $\mathbf{M}_0 = e^{\mathbf{A}_0 T_s} \approx \mathbf{I}$, which matches expectations from the previous section. Note that the classical INDI approach is retrieved by simply neglecting the term $\mathbf{H}_{sm}\mathbf{A}_0(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0$ from the expression of $s\mathbf{P}(s)$ (or their discretized equivalents), as is expected from the expression of $\mathbf{\Delta}_d$.

3.4.5 Open-loop Linearization Errors

The open-loop linearization errors (ϵ_{l0} or ϵ_l) have been neglected so far in this analysis as they are expected to be small since the synchronized delay is supposed to be as small as possible.

As previously mentioned, $\epsilon_{vl}(t)$ is bounded by terms dependent on the Lipschitz constants L_a and L_g, and there is a claim that the INDI architecture is able to reject such disturbances [6].

---

and `T_s` is the sampling period.

[8]This situation corresponds to having the ZOH's dynamics dominating the input dynamics $\mathbf{H}_{act}(s)\mathrm{ZOH}_{ts}(s) \approx \mathrm{ZOH}_{ts}(s)$.

One can thus describe ϵ_{vl} as the addition of 2 unstructured uncertainties:

$$\epsilon_{vl} = (\mathbf{A} - \mathbf{A}_0)\mathbf{y} + (\mathbf{G} - \mathbf{G}_0)\dot{\mathbf{u}} = (t - t_0)(L_a \mathbf{\Delta}_{la}\mathbf{y} + L_g \mathbf{\Delta}_{lg}\dot{\mathbf{u}}), \quad \|\mathbf{\Delta}_{la}\|_\infty \leq 1, \|\mathbf{\Delta}_{lg}\|_\infty \leq 1 \tag{3.38}$$

The maximum allowable values of L_a and L_g for a given implementation can then be assessed via a robustness analysis of the uncertain system, and their actual value in practice can be estimated from knowledge of $\dot{\mathbf{A}}$ (e.g., Hessian matrix or maximum possible value of $\dot{\mathbf{A}}$ over the time interval $[t_0, t]$). If this bound on the Lipschitz constant is insufficient, it can be increased by reducing the synchronized measurement delay, as expected from the classical INDI approach. Alternatively, the uncertainty could be structured to capture knowledge about $\dot{\mathbf{A}}$ and reduce the amount of conservatism in the robustness analysis.

The robustness of the INDI Feedback-Linearization stage to other uncertainties can be assessed similarly, with such an analysis potentially yielding valuable insight for the subsequent tuning of the INDI linear controller stage.

3.5 Application example: Rigid Aircraft

This section applies the INDI feedback-linearization stage to a civil transport aircraft, considering various actuator systems, sensor delays, and sampling rates. The goal is to highlight implementation challenges that practitioners may encounter and to better illustrate the phenomena that the proposed equations are designed to analyze.

3.5.1 Model presentation

This section presents the rigid aircraft model used for the application example. It is based on a common large civil transport aircraft [87, 88]. In order to properly represent the state variables, two reference frames are considered: an inertial frame $\mathcal{O}$ on the ground (North-East-Down convention), and the body frame $\mathcal{B}$ centred at the centre of mass (x-axis towards the front, z-axis towards the ground).

The following state variables[9] are considered :

- $\mathbf{p} = \begin{bmatrix} p_N & p_E & p_D \end{bmatrix}^\top$ is the position of the origin of $\mathcal{B}$ with respect to the origin of $\mathcal{O}$.
- $\mathbf{V} = \begin{bmatrix} u & v & w \end{bmatrix}^\top$ is the translational velocity expressed in $\mathcal{B}$.
- $\mathbf{\Phi} = \begin{pmatrix} \phi, & \theta, & \psi \end{pmatrix}$ are the Euler angles defining the rotations between $\mathcal{O}$ and $\mathcal{B}$.

[9]The vector $\mathbf{p}$ is expressed in $\mathcal{O}$, while the velocities ω and $\mathbf{V}$ are expressed in $\mathcal{B}$.

- $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ is the angular velocity of $\mathcal{B}$ relative to $\mathcal{O}$.

With these state variables, the 6 degrees-of-freedom rigid body dynamics can be described as follow:

$$\begin{aligned}
\dot{\mathbf{p}} &= \mathbf{R}_{\mathcal{O}/\mathcal{B}}(\boldsymbol{\Phi})\mathbf{V} \\
\dot{\boldsymbol{\Phi}} &= \mathbf{H}(\boldsymbol{\Phi})\boldsymbol{\omega} \\
\dot{\mathbf{V}} &= \frac{1}{m}\sum \mathbf{F} + \mathbf{R}_{\mathcal{O}/\mathcal{B}}\mathbf{g} - \boldsymbol{\omega} \times \mathbf{V} \\
\dot{\boldsymbol{\omega}} &= \mathbf{J}_b^{-1}\Big(\sum \mathbf{M} - \boldsymbol{\omega} \times \mathbf{J}_b\boldsymbol{\omega}\Big)
\end{aligned}$$

where $\mathbf{F}$ and $\mathbf{M}$ are the external forces and torques applied to the rigid body (excluding gravity), $\mathbf{g} = [0, 0, g]^\top$ is the acceleration of gravity, m is the total mass, $\mathbf{J}_b$ is the aircraft inertia matrix, $\mathbf{R}_{\mathcal{O}/\mathcal{B}}$ is the rotation matrix from $\mathcal{O}$ to $\mathcal{B}$, $\mathbf{H}$ is the transformation matrix for angular velocities, and $\times$ denotes the cross-product.

$$\sum \mathbf{F} = \mathbf{F}_a + \mathbf{F}_T$$

$$\sum \mathbf{M} = \mathbf{M}_a + \mathbf{M}_T$$

where $\mathbf{F}_T$ and $\mathbf{M}_T$ depend on T_e and the engine configuration (positions and angles w.r.t. the centre of mass).

$$\begin{aligned}
\mathbf{F}_a &= [X_a, Y_a, Z_a]^\top = \frac{1}{2}\rho V_T^2 S[C_X, C_Y, C_Z]^\top \\
\mathbf{M}_a &= [L_a, M_a, N_a]^\top = \frac{1}{2}\rho V_T^2 bS[C_l, C_m, C_n]^\top
\end{aligned}$$

$$T_e \approx (110000 - 75000Ma - 7.5h + 2hMa + 62000Ma^2)\delta_{th}$$

The aerodynamic coefficients will be approximated by the following equations:

$$
\begin{aligned}
C_X &= \frac{-\cos\alpha}{\cos\beta}(C_D + \sin\beta C_Y) + \sin\alpha C_L \\
C_Z &= \frac{-\sin\alpha}{\cos\beta}(C_D + \sin\beta C_Y) - \cos\alpha C_L \\
C_D &= C_{D_0} + C_{D_2} C_L^2 \\
C_L &= C_{L_\alpha}(\alpha - \alpha_0) + \frac{\bar{c}}{2V_T}(C_{L_q} q + C_{L_{\dot{\alpha}}}\dot{\alpha}) + C_{L_{\delta_e}}\delta_e \\
C_Y &= C_{Y_\beta}\beta + \frac{b}{2V_T}(C_{Y_p} p + C_{Y_r} r) + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\
C_l &= C_{l_\beta}\beta + \frac{b}{2V_T}(C_{l_p} p + (C_{l_{r_0}} + C_{l_{r_\alpha}}\alpha) r) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + \frac{\bar{c}}{2V_T}(C_{m_q} q + C_{m_{\dot{\alpha}}}\dot{\alpha}) + C_{m_{\delta_e}}\delta_e \\
C_n &= (C_{n_{\beta_0}} + C_{n_{\beta_\alpha}}\alpha)\beta + \frac{b}{2V_T}((C_{n_{p_0}} + C_{n_{p_\alpha}}\alpha) p + C_{n_r} r) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r
\end{aligned}
$$

with their numerical values given in Table 3.1.

Table 3.1 Aerodynamic coefficient values

Coefficients	Value	Coefficients	Value	Coefficients	Value	Coefficients	Value	Coefficients	Value
C_{D0}	0.016	C_{Y_β}	-1.42	C_{l_β}	-0.03	C_{m_0}	0.04	$C_{n_{\beta_0}}$	0.2
C_{D2}	0.0335	C_{Y_p}	0.0	C_{l_p}	-0.5	C_{m_α}	-1.5	$C_{n_{\beta_\alpha}}$	-0.0195
C_{L_α}	5.5	C_{Y_r}	0.0	$C_{l_{r_0}}$	0.15	C_{m_q}	-12	$C_{n_{p_0}}$	-0.25
C_{L_q}	3.3	$C_{Y_{\delta_a}}$	0.0	$C_{l_{r_\alpha}}$	0.035	$C_{m_{\dot{\alpha}}}$	-12	$C_{n_{p_\alpha}}$	-0.035
$C_{L_{\dot{\alpha}}}$	2.7	$C_{Y_{\delta_r}}$	0.05	$C_{l_{\delta_a}}$	-0.07	$C_{m_{\delta_e}}$	-1.2	C_{n_r}	-0.18
$C_{L_{\delta_e}}$	0.32			$C_{l_{\delta_r}}$	0.02			$C_{n_{\delta_a}}$	-0.03
α_0	-0.02							$C_{n_{\delta_r}}$	-0.09

The aircraft is considered to be in trimmed, straight and level flight at the considered (nominal) flight condition (Mach of 0.6, altitude of 4000m).

Consider the actuator systems listed in Table 3.2 as candidates for evaluating the applicability of INDI, noting that $\mathbf{K} = \Omega\mathbf{I}$ and $\mathbf{H}_{act}(s) = (s\mathbf{I} + \mathbf{K})^{-1}\mathbf{K}$ when applicable.

3.5.2 Actuator

First, the actuator's suitability for INDI is assessed by examining the idealized closed-loop response of the Feedback-Linearization stage. Figure 3.7 presents the time responses for various actuators to allow their comparison. Note that the loss of bandwidth translates into longer rise times and settling times (significant for slow actuators), and notice the coupling induced when using non-identical (non-uniform, disparate) actuators (Act.# 5 & 6). It is

Table 3.2 Actuator systems considered as candidate

ID	Name	Description
1	Very fast	$\Omega = 1000$
2	Fast	$\Omega = 100$
3	Moderate	$\Omega = 10$
4	Slow	$\Omega = 1$
5	Disparate	$\mathbf{K} = \mathrm{diag}\Big([10, 10, 5]^\top\Big)$
6	Realistic[a]	$\mathbf{H}_{act}(s) = \mathrm{diag}\Big(\Big[\frac{-0.5982s+1235}{s^2+61.27s+1235}, \frac{-8.419s+2046}{s^2+102.8s+2046}, \frac{12.61s^2-1185s+27350}{s^3+77.71s^2+3330s+27350}\Big]^\top\Big)$

[a] Values taken from [87] (p.29), representing the actuators of a common large civil transport aircraft.

also apparent that actuator # 6 exhibit a non-minimal phase behaviour, which prevents the inversion of actuator dynamics as part of the nonlinear dynamic inversion. Furthermore, all responses converge to identity once settled, as expected for the idealized system.

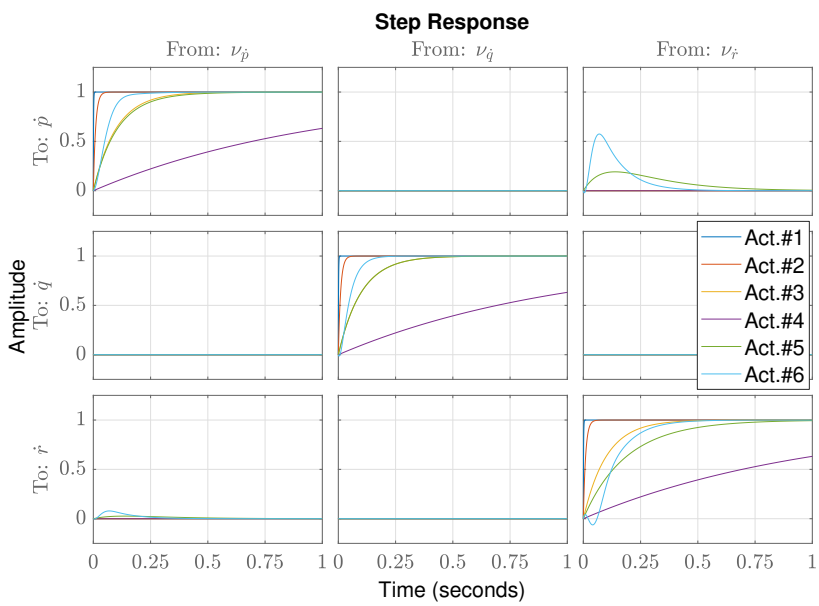

Figure 3.7 Comparison of ideal closed-loop time responses for various actuators

Modifying $\tilde{\mathbf{G}}_0^\dagger$ to account for disparate actuators can eliminate the coupling for actuator # 5, but only reduce it for # 6 due to its (non-invertible) higher order dynamics. The bandwidth of actuator # 4 appears insufficient to ensure a settling time under 1 second, which is unsuitable for most aircraft[10].

[10] See [89] for standard requirements, which includes an upper limit on the settling time of the roll rate between 1 and 3 seconds depending on the aircraft's class.

3.5.3 Plant

Now consider the plant dynamics and their impact on the closed-loop response. Figure 3.8 illustrates the frequency response of the delay-free continuous system for various uniform actuators. Notice that a reduction of actuator bandwidth reduces the effective tracking bandwidth of the closed-loop system as both the lower and upper cutoff frequencies shift towards the centre. Furthermore, the suppression of plant dynamics is proportional to the tracking bandwidth; as this bandwidth decreases, plant dynamics start to dominate. This is expected as we aim to have the term $\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}$ dominate $-\mathbf{A}_0$ in the denominator of the closed-loop transfer function, which directly depends on how large the actuator bandwidth $\mathbf{K}$ is. As such, the roots of $s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm} = 0$ should have their real part negative (closed-loop should be stable despite an unstable plant).

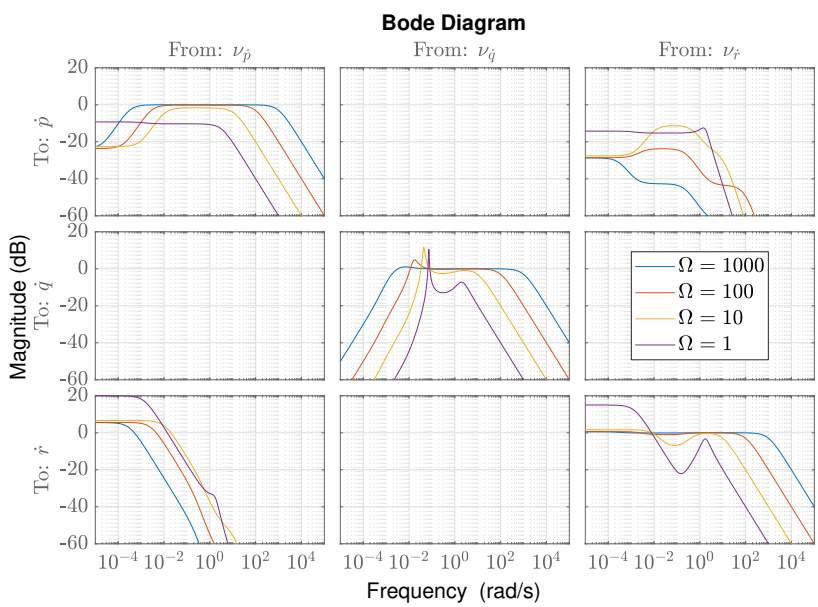

Figure 3.8 Comparison of delay-free closed-loop frequency responses for various actuators

Note that actuator #4 fails to stabilize the system as its bandwidth is insufficient to dominate the plant, which results in poor tracking across all frequencies.

3.5.4 Delay

Delay is now introduced in the system and the amount of synchronized delay that can be tolerated is assessed. Figure 3.9 presents the frequency responses for the closed-loop system with $0.05\,s$ of delay and various (uniform) actuators. The delay noticeably limits the lower cutoff frequency of the effective tracking bandwidth for all actuators, which leaves the

slower ones almost unchanged. The "effective actuator bandwidth" $\tilde{\mathbf{K}}(s)$ can be used as a replacement for $\mathbf{K}(s)$ for delayed systems in order to assess if the delay-induced reduction of bandwidth is acceptable. To better understand this, refer to Fig. 3.10 which presents the closed-loop frequency responses for various synchronized delays for actuator # 1. The performance degradation is similar to a reduction of actuator bandwidth, except that the higher frequencies are left untouched[11]. As such, increasing the actuator bandwidth or reducing the delay without considering the other will fail to improve performance as expected; both need to be adjusted to ensure that the resulting "effective actuator bandwidth" is appropriate for the level of performance desired.

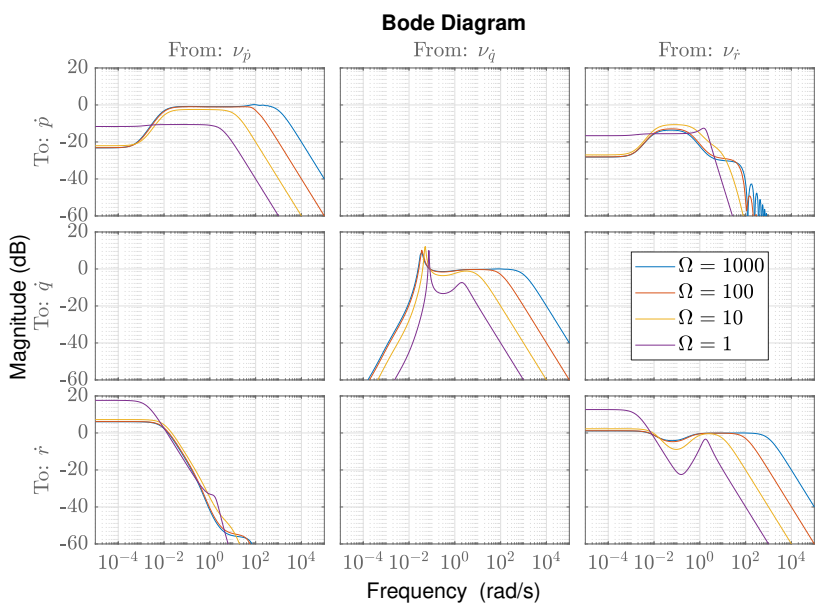

Figure 3.9 Comparison of delayed ($0.05s$) closed-loop frequency responses for various actuators

3.5.5 Sampling

Lastly, the system is discretized by adding the ZOH before the actuator system and the samplers before the synchronized measurement delays, yielding a discrete augmented plant in a closed loop with a discrete control law. Figures 3.11 and 3.12 show the frequency & time responses of the closed-loop for varying degrees of idealization, illustrating the performance degradation as we add back imperfections into the loop. For actuator # 6, we note that discretization has negligible effects on performance while plant dynamics are responsible for

[11]Figures 3.9 and 3.10 have curves that exhibit a non-smooth frequency response due to the ZOH term dominating the effective bandwidth expression for fast actuators and large enough delays.

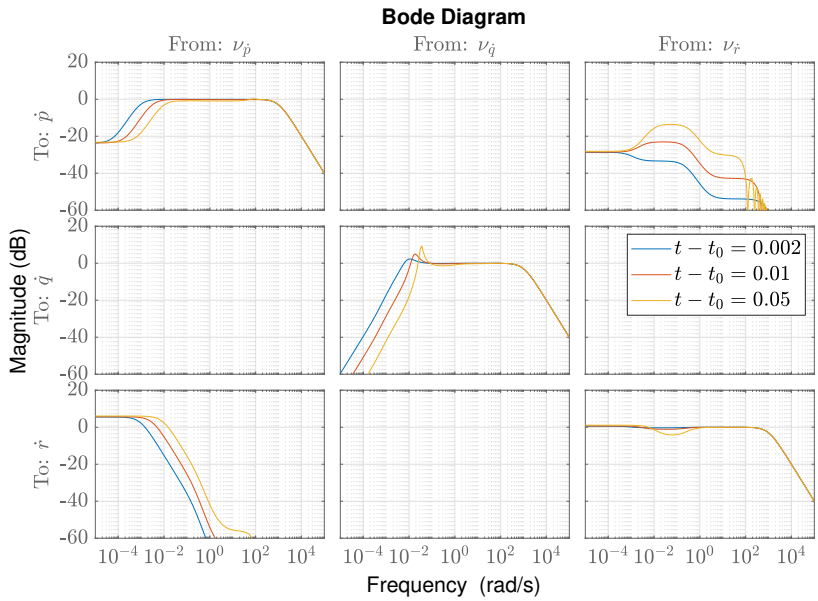

Figure 3.10 Comparison of delayed closed-loop frequency responses for various (synchronized) delays (using actuator # 1)

most of the 20% tracking error after $1s$. As such, merely increasing the sampling rate in this case will not yield any improvement in performance; the actuators and the measurement delays are the limiting factors. We note that the "Delayed (ideal)" signals match the "Delay-free" signals, as expected from the claim that $\mathbf{C}_b(s)$ (see Eq.(3.30)) compensates the delay to retrieve the delay-free tracking performance.

Additionally, Figs. 3.13 and 3.14 present the nonlinear time response of the closed-loop system with actuator #1 to a doublet command input (a step command input is unrealistic for the nonlinear model) for different sampling times. The main difference between the two figures is the intersample ripple effect exhibited in Fig. 3.13 that has vanished in Fig. 3.14 by increasing the sampling rate. This illustrates that while using the synchronized delay as the sampling period simplifies the dynamics equations, it also introduces an intersample ripple effect as the plant has enough time within a sampling period to start moving towards the reference only to start decaying back to the measured values (command-driven oscillation with period T_s, caused by a control spillover exciting the neglected (residual) plant dynamics [90]). This phenomenon can easily be missed if one neglects to simulate the actual plant output as it only manifest itself for discrete plant inputs. Slower actuators have less risk of exhibiting intersample rippling since their actuator outputs are smoother. The intersample ripple effect is suppressed as we increase the sampling rate, and grows larger as the sampling period (T_s) approaches the synchronized delay ($t - t_0$). This phenomenon was partly expected from

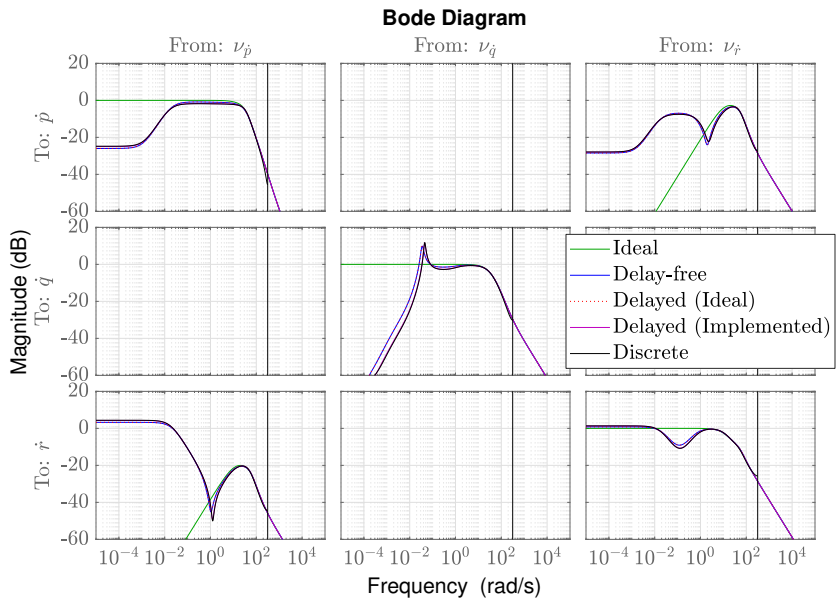

Figure 3.11 Comparison of idealized feedback-linearization closed-loop (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) - Frequency responses

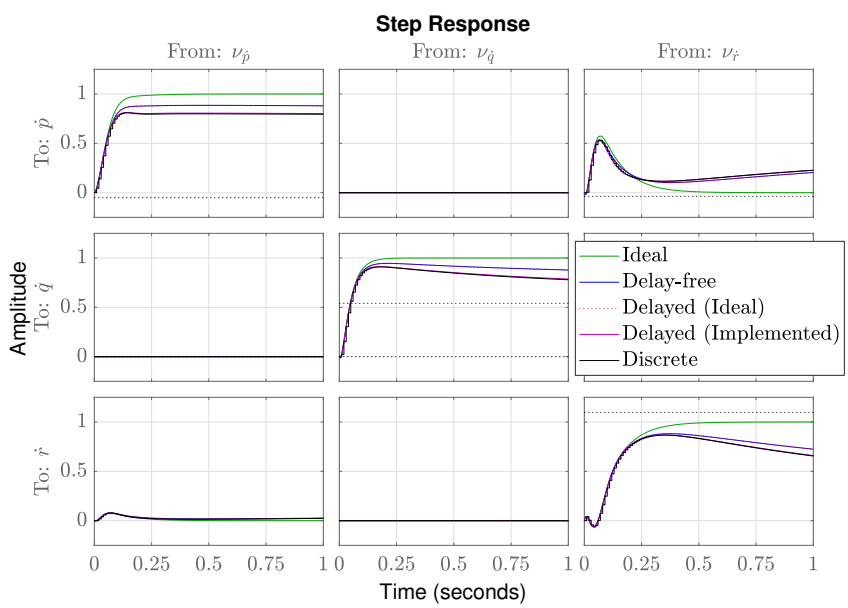

Figure 3.12 Comparison of idealized feedback-linearization closed-loop (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$) - Time responses

the presence of δ_m in Eq.(3.18); δ_m represents a portion of this ripple effect being captured in measurements, causing a multiplicative disturbance. For minimal delay $T_s = t - t_0$, δ_m vanishes while the intersample ripple effect is maximized and described by $e^{\mathbf{A}_0(\tau - t_0)}\mathbf{y}(t_0)$ for $\tau \in [t_0, t_0 + T_s]$.

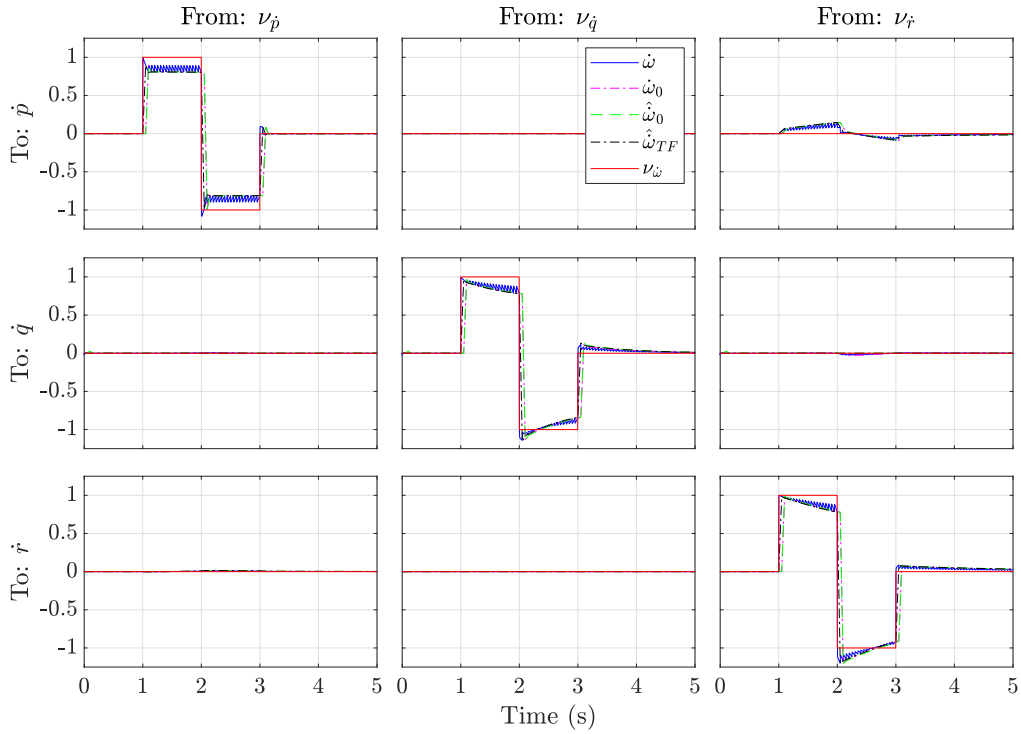

Figure 3.13 Simulations comparing the true nonlinear time response with its measured signal, linearized model approximation, and discrete transfer function approximation ($T_s = 0.05s$ chosen to highlight the intersample ripple effect)

3.5.6 Mitigation - Non-identical actuators

Disparate and realistic actuators (ID # 5 and 6) exhibit additional undesired coupled dynamics (especially yaw-to-roll acceleration coupling as shown in Fig. 3.12) as their differing bandwidths prevent the control effectiveness matrix from effectively decoupling the plant dynamics as intended. As mentioned before (see Section 3.4), this motivates the search for a mitigation technique to allow the use of non-identical actuators, for example the use of $\bar{\mathbf{G}}_0^\dagger$ (instead of $\tilde{\mathbf{G}}_0^\dagger$) which prevents the simplification carried out to obtain the $\mathbf{C}_c$ pseudo-controller. Starting from the unsimplified continuous loop of Fig. 3.4, the control effectiveness matrix is changed and $\mathbf{H}_{act}(s)$ is used instead of $\mathbf{K}(s)$ to yield the diagram of Fig. 3.15. Note, however, that this diagram is not implementable as it requires an undelayed actuator position

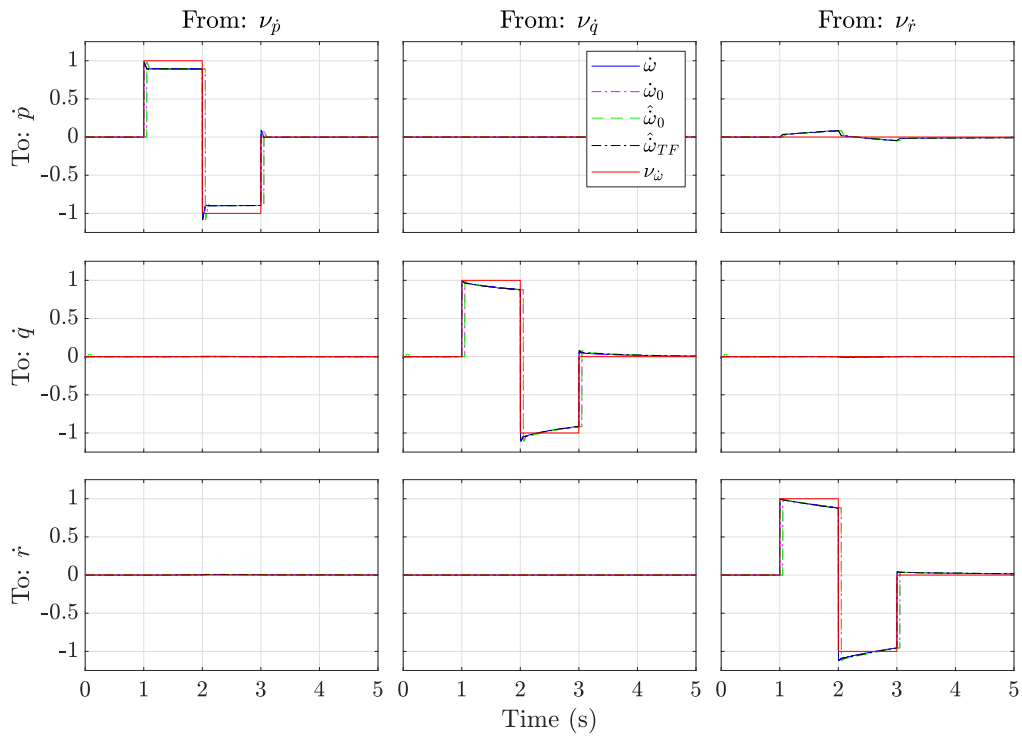

Figure 3.14 Simulations comparing the true nonlinear time response with its measured signal, linearized model approximation, and discrete transfer function approximation ($T_s = 0.002s$ chosen to suppress the intersample ripple effect)

feedback; Fig. 3.16 shows an equivalent diagram using an in-board model of the actuators as an approximation of the undelayed actuator position. Figures 3.17 and 3.18 show the time responses for the classical INDI and modified INDI Feedback-Linearization stage, where the yaw-to-roll coupling was significantly reduced[12] thanks to this mitigation approach. The classical INDI control law can be retrieved from this modified architecture by setting $\bar{\mathbf{G}}_0^\dagger = \tilde{\mathbf{G}}_0^\dagger$, $s\hat{\mathbf{P}}(s) = \mathbf{H}_{sm}\mathbf{G}_0$ and $\hat{\mathbf{H}}_{act}(s) = \mathbf{I}$. As with a Smith predictor, the models ($\hat{\mathbf{H}}_{act}(s)$ and $s\hat{\mathbf{P}}(s)$) can be improved to achieve better delay compensation, if need be.

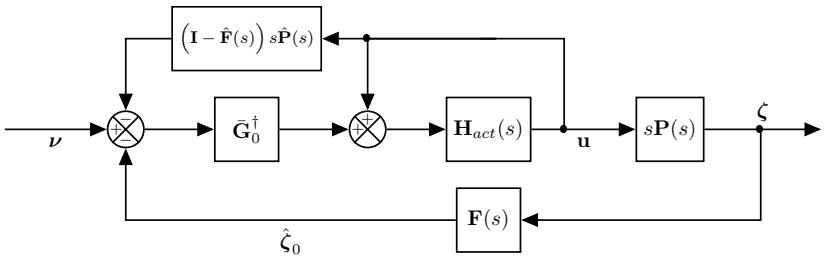

Figure 3.15 Modified INDI (not implementable)

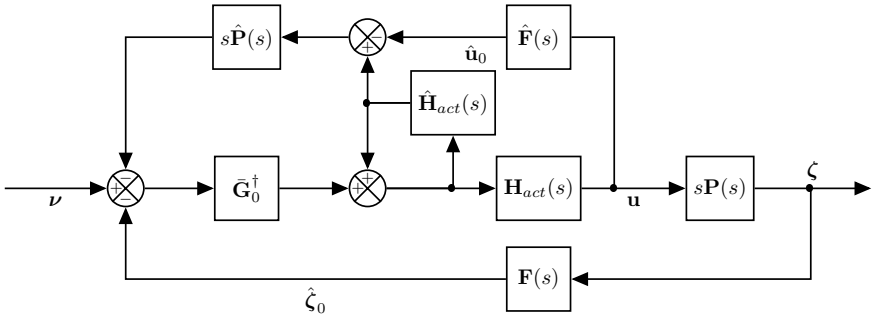

Figure 3.16 Modified INDI (implementable)

[12]The actuator-induced coupling was not completely eliminated due to actuator # 6 exhibiting higher order dynamics.

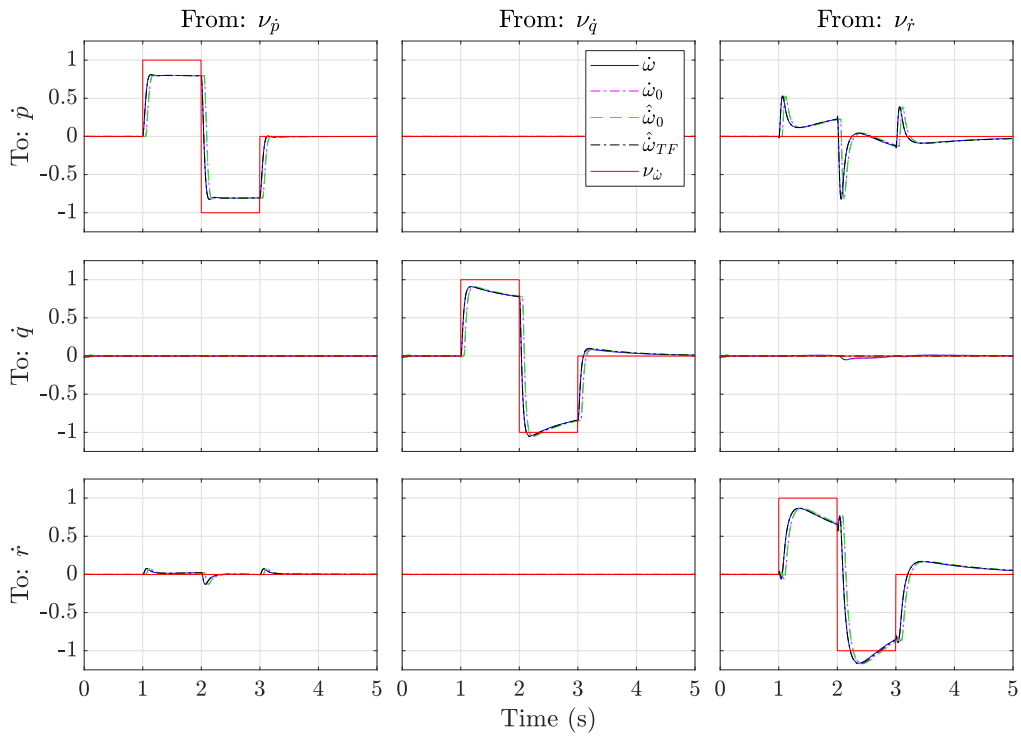

Figure 3.17 Unmodified response (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$)

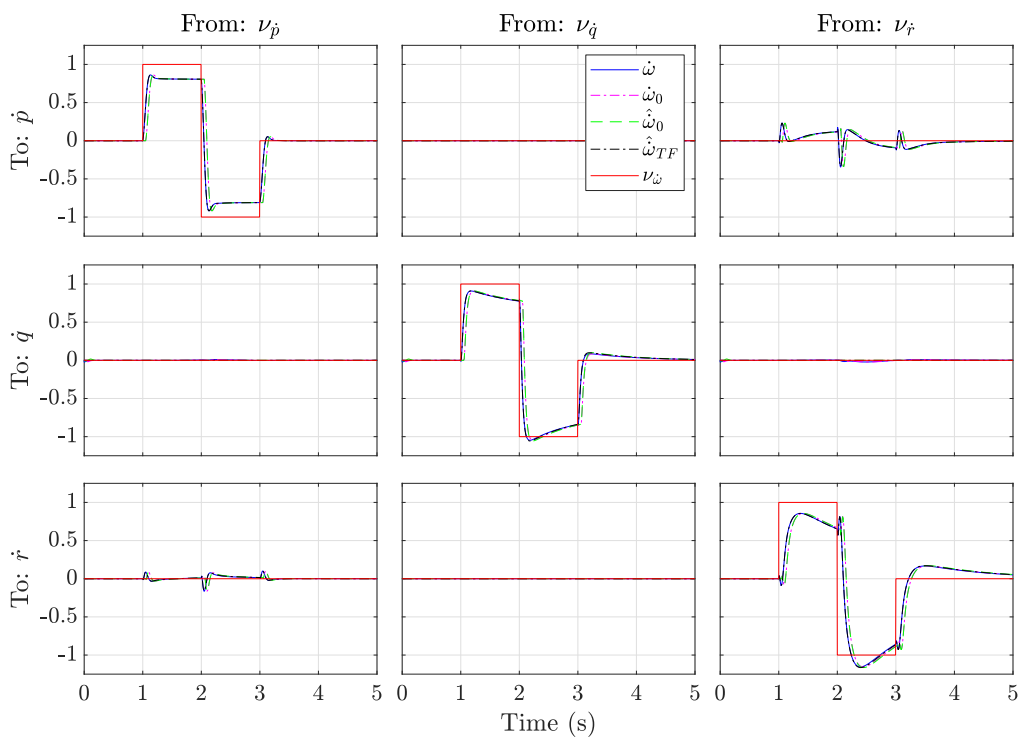

Figure 3.18 Modified INDI (Act.# 6, $t - t_0 = 0.05s$, $T_s = 0.01s$)

3.6 Discussion: Applicability conditions

In light of the analysis carried out in this paper, the conditions of applicability of the INDI technique can now be summarized and guidance for assessing the suitability of selected hardware for a given system can be provided. These have been synthesized into a series of simple tests presented in Table 3.3, including suggested mitigation approaches in the event of failure of the associated test. Note that tests involve norms that should include relevant weighting filters (as in [83, 84]) that capture relevant requirements, and that both $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms can be used depending on the specific requirements being enforced. As such, unfiltered norms (as presented) would be conservative and could obscure otherwise acceptable configurations. For this reason, the various tolerances depend on the application and requirements at hand and must thus be determined by the designer, or can be ignored for comparative analyses that do not need absolute thresholds.

Fulfillment of all criteria is required for Classical INDI applications, but mitigation can compensate most issues effectively for a modified INDI approach. Only the Stabilizing criterion is necessary, whose failure indicates that the actuator is slower than the plant's dynamics, given that special care is needed concerning the nonlinear effects of saturation and rate limitation surrounding actuators in practice.

Using a robust tuning method, such as the Structured $\mathcal{H}_\infty$ Synthesis, for the linear controller is recommended, especially when some criteria are not met. This synthesis has the added benefit of potentially reusing the weighting filters suggested for the preliminary assessment of the Feedback-Linearization stage.

Table 3.3 Tests for INDI applicability and proposed mitigation

Test	Expression	Mitigation	Comments
Decoupling (Identical actuators)	$\left\|\tilde{\mathbf{G}}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger - \mathbf{K}(s)\right\| < \epsilon_g$	Modified INDI	Simplification of the actuator loop ($\mathbf{C}_c$) relies on this decoupling and does not work with $\bar{\mathbf{G}}_0^\dagger$. Leads to actuator-induced coupled dynamics that cannot be rejected at the Feedback-Linearization stage, limiting achievable performance for the linear controller tuning process.
Stabilizing	$\left\|\left(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} - \mathbf{A}_0\right)^{-1}\right\| < 1$	Nonlinear	Slow actuators fail to bound the linearization error; advanced controller scheme required. Expression is part of $\mathbf{\Delta}_a$, acting as a (beneficial) low-pass filter.
Fast actuation	$\|\mathbf{\Delta}_a(s)\| < \epsilon_a$	Augmented plant, Modified INDI	Actuator dynamics must be taken into account if they are too slow to be neglected. Manifests as an output multiplicative error on the closed-loop transfer. Represents the neglected plant dynamics.
Slow varying (Locally Lipschitz)	$\|\mathbf{A}(t) - \mathbf{A}_0\| \leq L_a\tau$ $\|\mathbf{G}(t) - \mathbf{G}_0\| \leq L_g\tau$	Nonlinear, Anti-spillover filtering	Large Lipschitz constants (over the time interval $[t_0, t]$) prevents the rejection of linearization errors; advanced controller scheme required unless anti-spillover filtering can be used.
Small delay	$\left\|e^{\mathbf{A}_0\tau} - \mathbf{I}\right\| < \epsilon_t$	Modified INDI, Predictive filter	Plant dynamics must be taken into account if delay is too large to make it negligible OR delay must be made smaller by using predictive filters. Leads to model mismatch in the effective Smith Predictor loop that increases $\|\mathbf{\Delta}_d\|$ (and thus the difference between the classical INDI equations and the proposed alternative described by Eq.(3.23)).
Bandwidth ratio	$\tau\mathbf{K}(0) \geq \epsilon_{bw}\mathbf{I}$	Simpler control law (Negligible delay)	Delay compensation becomes unnecessary when actuator bandwidth dominates the effective bandwidth (can set $\hat{\mathbf{F}} = \mathbf{I}$ without significant loss of performance). Based on Nyquist's theorem ($\epsilon_{bw} \in (\frac{1}{2}, \frac{1}{5}, \frac{1}{10})$ depending on desired tolerance). Designer can either use the delay-free model directly or use a different (simpler) control law entirely.
Synchronized	$\left\|\mathbf{F} - \hat{\mathbf{F}}\right\| < \epsilon_f$	Online synchronization	Unknown or varying delays requires an onboard synchronization module [57] to ensure proper compensation. The upper bound on ϵ_f depends on the phase margin of the delay-free open-loop system [86].
Compensated	$\|\mathbf{\Delta}_d(s)\| < \epsilon_d$	Modified INDI, Predictive filter	Plant dynamics must be taken into account if delay compensation is inadequate OR delay must be made smaller by using predictive filters. Manifests as an input multiplicative error on the closed-loop transfer. Represents the mismatch between delays and plant models due to implementation.
Intersample rippling	$\left\|e^{\mathbf{A}_0 T_s} - \mathbf{I}\right\| < \epsilon_s$	Modified INDI, Filter	Low sampling rate may require mitigation to avoid undesired high-frequency oscillations (fatigue). Contributes to implementation errors due to model mismatch (similarly to $\mathbf{\Delta}_d$ for the continuous case).
Discrete delay	$\frac{\tau}{T_s} - D = 0$	New (larger) $\tau = DT_s$ (Must be reassessed)	Discretized delay is a multiple of the sampling time rounded up, effective delay increases if the multiple is not an integer or if the sampling period is larger than the measurement delay.

3.7 Conclusion

An alternate model was proposed using the frozen-time velocity-based linearization approach instead of truncating the Taylor series expansion, which reduces the feedback-linearization error of the INDI architecture during analysis. This model highlights undesirable phenomena, such as loss of tracking performance and intersample rippling, and allowed the characterization of the feedback-linearization stage that help discern the quantitative effects of the simplifying assumptions, along with potential mitigation approaches. An application example on a rigid aircraft was provided along with quantitative tests of applicability.

This investigation revealed that neglecting the slow plant dynamics results in a multiplicative uncertainty that is not robustly rejected by the INDI feedback-linearization architecture, that fast sampling fails at improving performance if the synchronized delay and actuator bandwidth are left unchanged, and that the delay compensation can be framed as a Smith Predictor. It also revealed a simple modification that reduces actuator-induced coupling when using non-identical actuators. Practical implementation will be made easier thanks to the quantitative applicability tests and associated proposed mitigation.

Future work will include the robust tuning of an INDI controller applied to a rigid aircraft with non-identical actuators and limited sampling capabilities using the Structured $\mathcal{H}_\infty$ Synthesis, and the extension of the technique using anti-spillover filtering for stable high-frequency plant dynamics (such as flexible aircraft).

CHAPTER 4 ARTICLE 2: ROBUST TUNING OF THE MODIFIED INCREMENTAL NONLINEAR DYNAMICS INVERSION FOR A RIGID AIRCRAFT

Frédérick Laliberté, David Saussié
Published in: Journal of Guidance, Control, and Dynamics
Submission date: March 30th 2025

Abstract

Recent applicability tests for the Incremental Nonlinear Dynamic Inversion (INDI) control scheme have been proposed to relax traditional assumptions. Additionally, a modification to this control architecture has been introduced as a mitigation technique for non-identical actuators, along with the integration of a robust linear controller to address performance degradation caused by non-ideal actuator bandwidths, sampling rates, and synchronized delays. This paper demonstrates the suitability of $\mathcal{H}_\infty$ Synthesis for robust controller tuning in the presence of significant sensor delays and realistic actuators on a rigid aircraft. Nonlinear performance is evaluated using the Worst-Case Analysis Toolbox, employing both a traditional Monte Carlo campaign and a worst-case optimization approach based on the Differential Evolution algorithm. Serving as a proof of concept, this study outlines a streamlined control design process for conventional aircraft, enabling automatic controller tuning and verification following model updates.

Nomenclature

$\mathbf{A}$	=	State matrix (State-Space model)
$\mathbf{B}$	=	Input matrix (State-Space model)
$\mathbf{C}$	=	Output matrix (State-Space model)
$\mathbf{D}$	=	Feedthrough matrix (State-Space model)
$\mathbf{x}$	=	State signal vector
$\mathbf{y}$	=	Output signal vector
$\mathbf{u}$	=	Input signal vector
ν	=	Pseudo-command vector
$\mathbf{r}$	=	Reference vector
$\mathbf{H}_{sm}$	=	Selector matrix of measured outputs meant to track pseudo-commands

$\hat{\mathbf{y}}$	=	Estimation or measurement of $\mathbf{y}$
T_s	=	Sampling period
α	=	Angle of attack
β	=	Angle of sideslip
ϕ	=	Bank angle
θ	=	Pitch angle
ψ	=	Azimuth angle
p	=	Roll rate
q	=	Pitch rate
r	=	Yaw rate
u	=	Longitudinal speed
v	=	Lateral speed
w	=	Vertical speed
p_n	=	Latitudinal position (North direction)
p_e	=	Longitudinal position (East direction)
p_d	=	Vertical position (Down direction)
Ma	=	Mach number
h	=	Altitude (over sea level)
δ_a	=	Aileron deflection angle
δ_e	=	Elevator deflection angle
δ_r	=	Rudder deflection angle
δ_{th}	=	Engine thrust lever angle
C_L	=	Aerodynamic coefficient - Lift force
C_D	=	Aerodynamic coefficient - Drag force
C_X	=	Aerodynamic coefficient - Body X-axis force
C_Y	=	Aerodynamic coefficient - Body Y-axis force
C_Z	=	Aerodynamic coefficient - Body Z-axis force
C_l	=	Aerodynamic coefficient - Body X-axis torque
C_m	=	Aerodynamic coefficient - Body Y-axis torque
C_n	=	Aerodynamic coefficient - Body Z-axis torque

4.1 Introduction

The aviation industry is striving to enhance cost efficiency in response to recent economic downturns. Developing a new aircraft model involves an extensive series of control-related activities, where increasingly flexible aircraft models require frequent updates, necessitat-

ing repeated tuning and verification of control laws. Combined with the traditional gain-scheduled control architecture, this results in a growing amount of time spent on repetitive and costly tasks. There is a clear need for more efficient methods to streamline this process.

Recent advances in sensor-based feedback-linearization control, commonly referred to as Incremental Nonlinear Dynamic Inversion (INDI) [6, 57, 65, 69, 73, 91–93], show promise as an alternative to gain scheduling, potentially leading to significant time savings. Instead of extensive gain scheduling, these approaches require modeling the control effectiveness matrix, which must be updated frequently to maintain sufficient accuracy. In [73], the authors proposed that the robustness assumptions underlying INDI could be relaxed at the cost of a predictable performance reduction. Additionally, they introduced a series of applicability tests to ensure that (approximate) feedback linearization can be achieved before implementation.

Additionally, modern controller tuning methods offer a range of algorithms that can significantly accelerate the process, such as Structured $\mathcal{H}_\infty$ Synthesis [46–50]. This approach builds on classical $\mathcal{H}_\infty$ synthesis, a robust controller design method that produces an optimal controller, typically of higher order than the plant it regulates, making practical implementation challenging. Structured $\mathcal{H}_\infty$ Synthesis addresses this limitation by using a predefined fixed architecture (provided *a priori*) and performing non-smooth optimization [46], yielding suboptimal yet implementable solutions. The algorithm for this optimization is available in MATLAB via the `systune` function, whose advanced features require the Robust Control Toolbox. Moreover, Structured $\mathcal{H}_\infty$ Synthesis supports multi-model and multi-objective optimization [49], allowing for customized requirement assignments across models, all optimized simultaneously. Its application to INDI was proposed as a future research direction in [73].

Furthermore, the Worst-Case Analysis Toolbox (WCAT) was developed in collaboration with the European Space Agency (ESA) to provide a framework that facilitates nonlinear analysis through optimization-based algorithms, enhancing Verification & Validation (V&V) activities. This toolbox can utilize the same norms used during synthesis as cost functions for nonlinear analysis. As a result, WCAT can perform Worst-Case analysis on a controller tuned using a Worst-Case approach (such as Structured $\mathcal{H}_\infty$ Synthesis), while also supporting traditional Monte Carlo (MC) campaigns.

Taken together, these three advances have the potential to complement each other, forming a streamlined process that automatically re-tunes and re-verifies the controller upon an aircraft model update.

This paper first presents the aircraft model and the hardware used in Section II, followed by the selection of a control law based on requirements in Section III. In Section IV, performance

is then assessed on the nonlinear model using both Monte Carlo and Worst-Case approaches.

4.2 Model

The model considered is based on a common large civil transport aircraft [87, 88] (same as in [73]), over its normal operation flight envelope. The model also includes suitable hardware to demonstrate the mitigation techniques proposed.

4.2.1 Aircraft rigid model

In order to properly represent the state variables, two reference frames are considered: an inertial frame $\mathcal{O}$ on the ground (North-East-Down convention), and the body frame $\mathcal{B}$ centered at the center of mass (x-axis towards the front, z-axis towards the ground). The following state variables are chosen: 1) $\mathbf{p} = \begin{bmatrix} p_n & p_e & p_d \end{bmatrix}^\top$ is the position of the origin of $\mathcal{B}$ with respect to the origin of $\mathcal{O}$ (expressed in $\mathcal{O}$), 2) $\mathbf{V} = \begin{bmatrix} u & v & w \end{bmatrix}^\top$ is the translational velocity expressed in $\mathcal{B}$, 3) $\mathbf{\Phi} = \begin{pmatrix} \phi, & \theta, & \psi \end{pmatrix}$ are the Euler angles defining the rotations between $\mathcal{O}$ and $\mathcal{B}$, and 4) $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ is the angular velocity of $\mathcal{B}$ relative to $\mathcal{O}$ (expressed in $\mathcal{B}$).

With these state variables, the six degrees-of-freedom rigid body dynamics can be described as follow:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{R}_{\mathcal{O}/\mathcal{B}}(\mathbf{\Phi})\mathbf{V}, & \dot{\mathbf{\Phi}} &= \mathbf{H}(\mathbf{\Phi})\boldsymbol{\omega}, \\ \dot{\mathbf{V}} &= \frac{1}{m}\sum \mathbf{F} + \mathbf{R}_{\mathcal{O}/\mathcal{B}}\mathbf{g} - \boldsymbol{\omega} \times \mathbf{V}, & \dot{\boldsymbol{\omega}} &= \mathbf{J}_b^{-1}\Big(\sum \mathbf{T} - \boldsymbol{\omega} \times \mathbf{J}_b\boldsymbol{\omega}\Big) \end{aligned} \tag{4.1}$$

where $\mathbf{F}$ and $\mathbf{T}$ are the external forces and torques applied to the rigid body (excluding gravity), $\mathbf{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^\top$ is the acceleration of gravity, m is the total mass, $\mathbf{J}_b$ is the aircraft inertia matrix, $\mathbf{R}_{\mathcal{O}/\mathcal{B}}$ is the rotation matrix from $\mathcal{O}$ to $\mathcal{B}$, $\mathbf{H}$ is the transformation matrix for angular velocities, and $\times$ denotes the cross-product. The external forces and torques ($\mathbf{F}$, $\mathbf{T}$) are categorized into aerodynamic forces and torques, as well as thrust generated by the engine (approximated as $T_e \approx (110000 - 75000Ma - 7.5h + 2hMa + 62000Ma^2)\delta_{th}$ at equilibrium). The aerodynamic forces and torques (F_i, T_i) are represented as:

$$\mathbf{F}_i = \bar{q}SC_i, \qquad \mathbf{T}_i = \bar{q}SkC_i \tag{4.2}$$

where $\bar{q}$ is the dynamic pressure, S a reference surface, k is a characteristic length (such as the wingspan b or mean aerodynamic chord $\bar{c}$, depending on the torque considered), and C_i are empirically determined aerodynamic coefficients.

The aerodynamic coefficients are approximated by the following formulas:

$$
\begin{aligned}
C_X &= \frac{-\cos\alpha}{\cos\beta}(C_D + \sin\beta C_Y) + \sin\alpha C_L \\
C_Z &= \frac{-\sin\alpha}{\cos\beta}(C_D + \sin\beta C_Y) - \cos\alpha C_L \\
C_D &= C_{D_0} + C_{D_2} C_L^2 \\
C_L &= C_{L_\alpha}(\alpha - \alpha_0) + \frac{\bar{c}}{2V_T}(C_{L_q} q + C_{L_{\dot{\alpha}}}\dot{\alpha}) + C_{L_{\delta_e}}\delta_e \\
C_Y &= C_{Y_\beta}\beta + \frac{b}{2V_T}(C_{Y_p} p + C_{Y_r} r) + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\
C_l &= C_{l_\beta}\beta + \frac{b}{2V_T}(C_{l_p} p + (C_{l_{r_0}} + C_{l_{r_\alpha}}\alpha) r) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + \frac{\bar{c}}{2V_T}(C_{m_q} q + C_{m_{\dot{\alpha}}}\dot{\alpha}) + C_{m_{\delta_e}}\delta_e \\
C_n &= (C_{n_{\beta_0}} + C_{n_{\beta_\alpha}}\alpha)\beta + \frac{b}{2V_T}((C_{n_{p_0}} + C_{n_{p_\alpha}}\alpha) p + C_{n_r} r) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r
\end{aligned}
$$

where constant coefficient values are given in Table 4.2.

The flight envelope considered is presented in Table 4.3. These flight conditions are intended to have a reasonable variation in aerodynamic behavior.

Dynamic pressure $\bar{q}$ and Mach number Ma depend on the density of the air ρ, which depend on altitude. As the flight conditions at the operational ceiling of the aircraft are in the stratosphere, the air density's nonlinear relation to altitude changes from a power relation to an exponential one. The standard atmospheric model [94] was used and implemented in Simulink for nonlinear simulations.

Table 4.2 Constant coefficient values

Coeff.	Value	Coeff.	Value	Coeff.	Value	Coeff.	Value	Coeff.	Value
C_{D0}	0.016	C_{Y_β}	-1.42	C_{l_β}	-0.03	C_{m_0}	0.04	$C_{n_{\beta_0}}$	0.2
C_{D2}	0.0335	C_{Y_p}	0.0	C_{l_p}	-0.5	C_{m_α}	-1.5	$C_{n_{\beta_\alpha}}$	-0.0195
C_{L_α}	5.5	C_{Y_r}	0.0	$C_{l_{r_0}}$	0.15	C_{m_q}	-12	$C_{n_{p_0}}$	-0.25
C_{L_q}	3.3	$C_{Y_{\delta_a}}$	0.0	$C_{l_{r_\alpha}}$	0.035	$C_{m_{\dot{\alpha}}}$	-12	$C_{n_{p_\alpha}}$	-0.035
$C_{L_{\dot{\alpha}}}$	2.7	$C_{Y_{\delta_r}}$	0.05	$C_{l_{\delta_a}}$	-0.07	$C_{m_{\delta_e}}$	-1.2	C_{n_r}	-0.18
$C_{L_{\delta_e}}$	0.32			$C_{l_{\delta_r}}$	0.02			$C_{n_{\delta_a}}$	-0.03
α_0	-0.02							$C_{n_{\delta_r}}$	-0.09

Table 4.3 Flight conditions

Altitude (m)	Mach						
	0.25	0.4	0.5	0.6	0.7	0.8	0.85
12000					# 19	# 20	# 21
10000					# 16	# 17	# 18
8000					# 13	# 14	# 15
6000				# 10	# 11	# 12	
4000			# 7	# 8	# 9		
2000		# 4	# 5	# 6			
1000	# 1	# 2	# 3				

4.2.2 Hardware models

In addition to the rigid aircraft model, hardware models for sensors, actuators, and on-board computers are also required. The following sections present their transfer functions, time delays, and sampling periods.

4.2.2.1 Actuators

The actuators considered in this paper are the "realistic actuators" described in [73], as they present the most challenges for INDI use and thus require the most mitigation. The actuator model (taken from [87] (p.29), representing the actuators of a common large civil transport aircraft) considered is:

$$\mathbf{H}_{act}(s) = \text{diag}\left(\left[\frac{-0.5982s + 1235}{s^2 + 61.27s + 1235}, \frac{-8.419s + 2046}{s^2 + 102.8s + 2046}, \frac{12.61s^2 - 1185s + 27350}{s^3 + 77.71s^2 + 3330s + 27350}\right]\right) \tag{4.3}$$

with associated actuator bandwidth function [73]:

$$\mathbf{K}(s) = (\mathbf{I} - \mathbf{H}_{act}(s))^{-1} s\mathbf{H}_{act}(s) \Leftrightarrow \mathbf{H}_{act}(s) = (s\mathbf{I} + \mathbf{K}(s))^{-1}\mathbf{K}(s) \tag{4.4}$$

These actuators have the following relevant characteristics for INDI use:

1. They are not of first order, leading to high-frequency transient responses that may contribute to the coupling of dynamics.

2. They are non-minimum phase, causing plant input delays and preventing their inclusion in a dynamic inversion scheme.

3. They are non-identical, violating the INDI assumption required for decoupling and simplifying the (incremental) actuator feedback loop.

4. They have a unit steady-state, meaning they exhibit no steady-state error.

5. They are strictly proper, meaning the associated bandwidth function is proper (causal).

6. Their associated bandwidth function reaches a constant steady-state, indicating stability and that $\mathbf{K}(0)$ is a finite gain.

The nonlinear model also includes saturation and rate limits on the control surface deflections. The ailerons are limited to a range of $[-20, 30]^\circ$, the elevators to $[-30, 17]^\circ$, and the rudder to $\pm 30^\circ$. Angular rates are limited to $\pm 40^\circ/\text{s}$ for the ailerons and rudder, and to $\pm 30^\circ/\text{s}$ for the elevators. Other nonlinear effects, such as mechanical backlash, are neglected

4.2.2.2 Sensors

The sensors considered are idealized with perfect precision but subject to a fixed measurement delay, which impacts accuracy. It is assumed that any measurement error contributes to steady-state error, which the robust controller to be designed is intended to reject. A synchronized delay of $\tau = 0.05$ s is considered in this paper. Practical applications may require a transfer function to represent sensors exhibiting frequency-dependent delays, or may utilize a Kalman filter to estimate or predict acceleration values through sensor fusion.

4.2.2.3 Sampling

All control laws, samplers, and zero-order holds are assumed to be synchronized to the onboard computer sampling period of $T_s = 0.01$ s, which implies a (synchronized) discrete delay of $D = \tau/T_s = 5$ samples (or timesteps). Practical applications may involve additional controllers, such as an actuator's servocontroller, which may not share the same sampling rate and thus require special considerations [65].

4.3 Control

The control law considered is responsible for tracking the angular rates, allowing the aircraft to follow pilot commands (i.e., the control augmentation system) or a desired trajectory (generated by "outer" loops, thus forming an autopilot). The Modified INDI architecture [73], which includes a Feedback-Linearization (FL) stage and a Linear-Controller (LC) stage, is

used, and the tunable gains (of the linear controller) are obtained using the Structured $\mathcal{H}_\infty$ Synthesis to enforce robustness.

This section first presents the requirements that must be fulfilled by the control law, then assesses INDI applicability to the considered system, followed by the control architecture of the modified INDI scheme and the choice of linear controller. Finally, the robust tuning process of the linear controller gains is presented.

4.3.1 Requirements

The requirements considered for the LC stage are designed to ensure good angular rate tracking, allowing the outer loops (responsible for tracking the other states of the aircraft to perform maneuvers and follow a given trajectory) to achieve the overall aircraft performance necessary for certification. In addition to a simple tracking requirement for each axis, the closed-loop dynamics are expected to be decoupled, robust to unmodeled dynamics and disturbances, and energy-efficient to avoid physical limitations of the actuator, such as saturation.

4.3.1.1 Tracking

The following time-domain requirements are considered: 1) a settling time (5%) of 4 s for roll, 5 s for pitch, and 8 s for yaw rates; 2) a maximum steady-state error of 5%; and 3) a maximum overshoot of 20% for the pitch axis, and 5% for the roll and yaw axes. These requirements were inspired by [57, 87] and are intended to ensure that the outer loops will be able to meet their own requirements with proper tuning.

4.3.1.2 Decoupling

Bounds are imposed to limit the impact of coupled dynamics on performance to at most 1% error for pitch couplings and at most 10% error for lateral-directional couplings. These bounds are intended to provide a reasonably decoupled closed-loop system for the outer loops to command. These decoupling requirements are similar to those imposed on traditional aircraft controllers, whose actuators are designed to assist with this task.

4.3.1.3 Stability margins

Each axis should have 6 dB of gain margin and 45° of phase margin at every flight condition to ensure sufficient stability margins throughout the flight envelope.

4.3.1.4 Control effort

An additional requirement is introduced to incentivize control effort efficiency during synthesis. This requirement limits the gain between the reference (controller input) and the pseudo-command (controller output) to a value of 10. This aims to keep the pseudo-command signals within saturation and rate limits during normal flight operations. Adverse nonlinear effects during abrupt maneuvers can be mitigated using pseudo-control hedging (PCH) [57].

4.3.2 INDI control law

This subsection presents the core principles of INDI along with useful models proposed in [73], including the modified INDI architecture.

4.3.2.1 Classic INDI

Considering a nonlinear system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n$, and $\mathbf{u} \in \mathbb{R}^p$, it can be linearized around the time of the last available measurement t_0 using a Taylor Series expansion [76]:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) + \mathcal{O}\Big((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2\Big) \tag{4.5}$$

where $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\mathbf{A}_0 = {\partial \mathbf{f}(\mathbf{x},\mathbf{u})}/{\partial \mathbf{x}}\big|_{t=t_0}$, $\mathbf{G}_0 = {\partial \mathbf{f}(\mathbf{x},\mathbf{u})}/{\partial \mathbf{u}}\big|_{t=t_0}$.

If the acceleration and command signals are measured (or estimated [64, 77]), $\dot{\mathbf{x}}_0$ and $\mathbf{u}_0$ are known and can be used to achieve feedback linearization. In practice, $\mathbf{G}_0$ can usually be estimated with good precision [57, 62, 78], but a good model for $\mathbf{A}_0$ is generally difficult to achieve. However, $\lim_{t\to t_0} \|\mathbf{A}_0(\mathbf{x} - \mathbf{x}_0)\| = 0$ and $\lim_{t\to t_0} \|\mathcal{O}((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2)\| = 0$, which leads to

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) + \epsilon_{l0} \approx \dot{\mathbf{x}}_0 + \mathbf{G}_0(\mathbf{u} - \mathbf{u}_0) \tag{4.6}$$

with $\epsilon_{l0} = \mathbf{A}_0(\mathbf{x}-\mathbf{x}_0)+\mathcal{O}((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2)$ as the linearization error term, which becomes negligible with a sufficiently high sampling frequency.

The INDI control law is defined as:

$$\mathbf{u}(t) = \hat{\mathbf{u}}(t_0) + \tilde{\mathbf{G}}_0^{\dagger}(\nu - \mathbf{H}_{sm}\hat{\mathbf{y}}(t_0)) \tag{4.7}$$

where $\mathbf{y} = \dot{\mathbf{x}}$, $\nu \in \mathbb{R}^p$ is the pseudo-command generated by a linear controller, and $\tilde{\mathbf{G}}_0^{\dagger} \in \mathbb{R}^{p\times p}$ is the (modified [79, 80]) on-board model (approximation) of $(\mathbf{H}_{sm}\mathbf{G}_0)^{\dagger}$, the Moore-Penrose pseudo-inverse of $\mathbf{H}_{sm}\mathbf{G}_0$ (the control effectiveness of $\mathbf{u}$ on $\mathbf{H}_{sm}\mathbf{y}$ at time t_0). Note that

$\mathbf{H}_{sm}\mathbf{G}_0\tilde{\mathbf{G}}_0^\dagger = (\mathbf{I} + \Delta\mathbf{G}) \approx \mathbf{I}$ where the error $\Delta\mathbf{G} \in \mathbb{R}^{p\times p}$ is generally assumed to be negligible. Adding this control law (Eq.(4.7)) to the system's dynamics equation (Eq.(4.6)) while neglecting measurement and modeling errors ($\hat{\mathbf{y}} \approx \mathbf{y}$, $\hat{\mathbf{u}} \approx \mathbf{u}$, $\Delta\mathbf{G} \approx \mathbf{0}$), and assuming quasi-instantaneous actuators ($\mathbf{H}_{act}(s) \approx \mathbf{I}$), yields

$$\zeta = \mathbf{H}_{sm}\mathbf{y} = \zeta_0 + \tilde{\mathbf{G}}_0(\mathbf{u} - \mathbf{u}_0) + \mathbf{H}_{sm}\epsilon_{l0}, \qquad \mathbf{u} = \mathbf{u}_0 + \tilde{\mathbf{G}}_0^\dagger(\nu - \zeta_0) \tag{4.8}$$

where ζ represents the measured output used for the Feedback-Linearization loop. The closed-loop system is then $\zeta \approx \nu + \mathbf{H}_{sm}\epsilon_{l0}$, where $\mathbf{H}_{sm}\epsilon_{l0} \approx \mathbf{0}$ for sufficiently small delays ($t \approx t_0$).

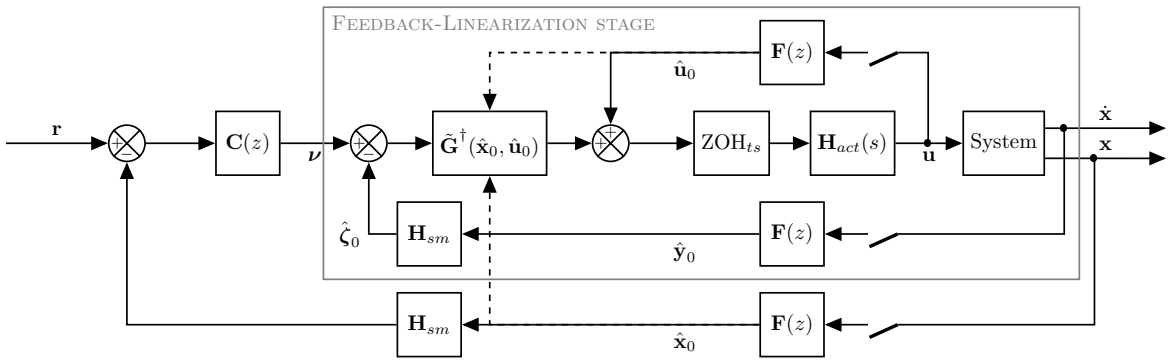

Figure 4.1 INDI Control Law

The INDI control architecture is presented in Fig. 4.1, where the Feedback-Linearization stage was highlighted, where $\mathbf{H}_{act}(s)$ is the actuator system, where ZOH$_{ts}$ represents a Zero-Order Hold (ZOH) of sampling period T_s, where sensors are represented by a sampler (with sampling period of T_s) and a synchronized delay $\mathbf{F}(z)$, and where $\mathbf{C}(z)$ is the linear controller.

4.3.2.2 Modified INDI and intermediate models

It was proposed in [73] to linearize the nonlinear system using a frozen-time velocity-based approach (instead of a Taylor series expansion):

$$\dot{\mathbf{y}}(t) = \mathbf{A}_0\mathbf{y}(t) + \mathbf{G}_0\dot{\mathbf{u}}(t) + \epsilon_{vl}(t) \Rightarrow \mathbf{y}(t) = e^{\mathbf{A}_0(t-t_0)}\mathbf{y}_0 + \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\mathbf{G}_0\dot{\mathbf{u}}(\tau)d\tau + \epsilon_l(t) \tag{4.9}$$

where $\epsilon_{vl}(t) = (\mathbf{A}(t) - \mathbf{A}_0)\mathbf{y}(t) + (\mathbf{G}(t) - \mathbf{G}_0)\dot{\mathbf{u}}(t)$ and $\epsilon_l(t) = \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\epsilon_{vl}(\tau)\mathrm{d}\tau$. Note that $\epsilon_{l0} = \epsilon_l + \left(e^{\mathbf{A}_0(t-t_0)} - \mathbf{I}\right)\mathbf{y}_0 + \left(\int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\mathbf{G}_0\dot{\mathbf{u}}(\tau)d\tau - \mathbf{G}_0(\mathbf{u}(t) - \mathbf{u}_0)\right) \Rightarrow \|\epsilon_{l0}\| \geq \|\epsilon_l\|$, which indicates that this approach allows for better modeling accuracy.

The system can then be represented by the following expressions [73]:

1. For ideal actuators and sensors ($\mathbf{H}_{act}(s) = \mathbf{I}$, $t = t_0$), the closed-loop simplifies to $\zeta = \nu$ as is the case for classic INDI.

2. For realistic actuators and an ideal plant ($\mathbf{A}_0 = \mathbf{0}$ and $\epsilon_l = \mathbf{0}$, $t = t_0$ is implied), the closed-loop simplifies to

$$\zeta(s) = \mathbf{H}_{sm}\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\nu(s) \tag{4.10}$$

   which will be referred to as the ideal model.

3. For realistic actuators and ideal sensors ($\mathbf{H}_{act}(s) \neq \mathbf{I}$, $t = t_0$), the closed-loop is

$$\zeta(s) = \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\Big(\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^\dagger\nu(s) + \epsilon_l(s)\Big) \tag{4.11}$$

   which will be referred to as the "delay-free" continuous model.

4. For realistic actuators and sensors ($\mathbf{H}_{act}(s) \neq \mathbf{I}$, $t > t_0$), the closed-loop becomes

$$\begin{aligned}\zeta(s) &= \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{F}(s)\mathbf{H}_{sm}\Big)^{-1}\Big(\mathbf{G}_0\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger\nu(s) + \epsilon_l(s)\Big) &(4.12)\\ &= \Big(\mathbf{I} + \mathbf{P}_0(s)\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger\mathbf{F}(s)\mathbf{H}_{sm}\Big)^{-1}\Big(\mathbf{P}_0(s)\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger\nu(s) + \mathbf{H}_{sm}\epsilon_l(s)\Big) &(4.13)\end{aligned}$$

   where $\tilde{\mathbf{K}}(s) = \Big(\mathbf{I} + \frac{1}{s}(\mathbf{I} - \hat{\mathbf{F}}(s))\mathbf{K}(s)\Big)^{-1}\mathbf{K}(s) = \Big(\mathbf{I} - \hat{\mathbf{F}}(s)\mathbf{H}_{act}(s)\Big)^{-1}s\mathbf{H}_{act}(s)$ is the effective bandwidth, $\mathbf{P}_0(s) = \mathbf{H}_{sm}(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0$ is the plant, and $\mathbf{F}(s) = e^{-s(t-t_0)}\mathbf{I}$ is the delay (with $\hat{\mathbf{F}}(s) = \mathbf{F}(s)$ when delays are synchronized). This will be referred to as the delayed continuous model.

Note that discretizing these equations should be done using an augmented plant model, that includes the actuator dynamics, in order to capture their convolution [73].

Furthermore, actuator-induced coupling can be mitigated using this modification:

$$\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^\dagger \mapsto \Big(\mathbf{I} + s\hat{\mathbf{P}}_0(s)\Big(\hat{\mathbf{H}}_{act}(s) - \hat{\mathbf{F}}(s)\mathbf{H}_{act}(s)\Big)\bar{\mathbf{G}}_0^\dagger\Big)^{-1}\hat{\mathbf{K}}(s)\bar{\mathbf{G}}_0^\dagger \tag{4.14}$$

where $\hat{\mathbf{K}}(s) = \Big(\mathbf{I} - \hat{\mathbf{H}}_{act}(s)\Big)^{-1}s\mathbf{H}_{act}(s)$, $\bar{\mathbf{G}}_0^\dagger = \Big(\tilde{\mathbf{G}}_0\mathbf{K}(0)\Big)^\dagger\mathbf{K}(0)$ is a modified control effectiveness matrix, $\hat{\mathbf{H}}_{act}(s)$ is the on-board model of the actuators, and $\hat{\mathbf{P}}_0(s) = {}^1\!/_s\tilde{\mathbf{G}}_0$ is the on-board model of the plant. Also note that both classic and modified expressions are equivalent for identical actuators ($\bar{\mathbf{G}}_0^\dagger = \tilde{\mathbf{G}}_0^\dagger$) due to the choice of the on-board plant model (ideal model for ideal actuators).

4.3.3 INDI applicability and mitigation - Feedback-Linearization stage

The INDI control law requires several assumptions to be met in order to be applicable to a given system. As mentioned in Section 4.2.2.1, the actuators considered do not meet the classical INDI assumptions of idealized actuators. However, [73] proposed a collection of applicability tests that are less conservative than the classical assumptions, and suggested associated possible mitigation approaches when a test fails. Furthermore, these tests involve system norms that can be shaped via weighting filters to enforce relevant requirements.

The applicability tests, weighting filters, mitigation, and assessment are presented in the following lines.

4.3.3.1 Applicability tests

Table 4.4 presents the INDI applicability tests and includes considered mitigation and relevant comments. The linear controller (second stage of the INDI control law) will be tuned robustly using the Structured $\mathcal{H}_\infty$ Synthesis in order to mitigate expected tracking errors, given that Eq.(4.12) can be written as

$$\frac{\zeta(s)}{\nu(s)} = \mathbf{H}_{sm}(\mathbf{I} + \mathbf{\Delta}_a(s))\Big\{\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}(s)\tilde{\mathbf{G}}_0^{\dagger}\Big\}(\mathbf{I} + \mathbf{\Delta}_d(s)) \tag{4.15}$$

where the ideal model from Eq.(4.10) is perturbed by two terms ($\mathbf{\Delta}_a$ and $\mathbf{\Delta}_d$) capturing the (neglected) plant dynamics and delay-induced errors.

Table 4.4 Tests for INDI applicability and proposed mitigation

Test	Expression	Mitigation	Comments
Decoupling	$\left\|\mathbf{W}_1\left(\tilde{\mathbf{G}}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger - \mathbf{K}\right)\right\|_\infty < 1$	Modified INDI	Not expected to hold with realistic actuators. Modified $\tilde{\mathbf{G}}^\dagger$ should pass this test.
Stabilizing	$\left\|\mathbf{W}_2\left(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} - \mathbf{A}_0\right)^{-1}\right\|_\infty < 1$	-	Must be passed for all flight conditions.
Fast Actuation	$\|\mathbf{W}_3\boldsymbol{\Delta}_a\|_\infty < 1,$ $\boldsymbol{\Delta}_a = \left(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} - \mathbf{A}_0\right)^{-1}\mathbf{A}_0$	Robust tuning	Robust tuning will compensate this tracking error.
Slow varying (Locally Lipschitz)	$\|\mathbf{W}_{4l}(\mathbf{A}(t) - \mathbf{A}_0)\mathbf{W}_{4r}\|_\infty \leq 1$ $\Rightarrow \tau\left\|\mathbf{W}_{4l}\left(\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t}\right)\mathbf{W}_{4r}\right\|_\infty \leq 1,$ $\|\mathbf{W}_{4l}(\mathbf{G}(t) - \mathbf{G}_0)\mathbf{W}_{4r}\|_\infty \leq 1$ $\Rightarrow \tau\left\|\mathbf{W}_{4l}\left(\frac{\mathrm{d}\mathbf{G}}{\mathrm{d}t}\right)\mathbf{W}_{4r}\right\|_\infty \leq 1$	-	Must be passed, acts as an upper bound on acceptable synchronized delays. The maximum value of the matrix derivatives ($^{\mathrm{d}\mathbf{A}}/_{\mathrm{d}t}$, $^{\mathrm{d}\mathbf{G}}/_{\mathrm{d}t}$) over the entire flight envelope, including non-trimmed configurations within saturation and rate limits, are used as a conservative upper bound for the (local) Lipschitz constants.
Small delay	$\left\|\mathbf{W}_5\left(e^{\mathbf{A}_0\tau} - \mathbf{I}\right)\right\|_\infty < 1$	-	Should be passed for the angular acceleration output, not required to pass for slower states thanks to outer-loop controllers.
Bandwidth ratio	$\tau\mathbf{K}(0) \geq \epsilon_{bw}\mathbf{I}$	-	Expected to pass due to realistic hardware, making delays non-negligible and requiring compensation. Considering $\epsilon_{bw} = 0.2$ as threshold, which corresponds to requiring a pole to be (at least) 5 times smaller than the others to be considered as dominant.
Synchronized	$\left\|\mathbf{W}_7\left(\mathbf{F} - \hat{\mathbf{F}}\right)\right\|_\infty < 1$	-	Expected to pass given idealized sensors.
Compensated	$\|\mathbf{W}_8\boldsymbol{\Delta}_d\|_\infty < 1,$ $\boldsymbol{\Delta}_d = \mathbf{E}_i\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\left(\mathbf{I} + (\mathbf{P} + \mathbf{E}_i)\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\right)^{-1},$ $\mathbf{E}_i = (\mathbf{I} - \hat{\mathbf{F}})\hat{\mathbf{P}} - (\mathbf{I} - \mathbf{F})\mathbf{P}$	Robust tuning	Robust tuning will compensate this tracking error.
Intersample rippling	$\left\|\mathbf{W}_9\left(e^{\mathbf{A}_0 T_s} - \mathbf{I}\right)\right\|_\infty < 1$	-	Expected to pass given that considered realistic hardware are less aggressive than idealized ones.
Discrete delay	$\frac{\tau}{T_s} - D = 0$	-	Expected to pass by construction of the discrete delay considered.

4.3.3.2 Applicability Weighting filters

The Feedback-Linearization stage is expected to yield a closed-loop system that behaves as a linear plant, as close as possible to a group of decoupled pure integrators. The angular acceleration frequency response being near 0 dB translates into the angular rate frequency response being near the ideal loop shape (pure integrator); this implies that any deviation from 0 dB at the FL stage will require compensation at the LC stage, limiting achievable performance. This highlights a need to assess whether the FL stage achieves sufficient performance as to allow a properly tuned LC stage to fulfill the requirements stated in Section 4.3.1.

In order to assess the applicability of the INDI control law, weighting filters are used to shape the test norms as to enforce minimal performance, which ensures that the requirements can be fulfilled. These filters represent the inverse of the maximum "error profile" tolerated as seen on a singular values plot. As such, they should generally have high gains over the FL

stage's required tracking bandwidth and low gains otherwise, but must also be bi-stable (bi-proper) transfer functions (or constant gains). A passed test means that all flight conditions have their singular values below the maximum error profile defined by the corresponding weighting filter, which should imply an acceptable FL stage response.

The selected weighting filters are:

$$\mathbf{W}_1 = \frac{s+50}{50s+0.25}\mathbf{I}, \quad \mathbf{W}_2 = \frac{s+0.005}{s+0.5}\mathbf{H}_{sm}, \quad \mathbf{W}_3 = \frac{20s+2.5}{s+50}\mathbf{H}_{sm},$$

$$\mathbf{W}_{4l} = \mathbf{D}_{scaling}, \quad \mathbf{W}_{4r} = \mathbf{D}_{scaling}^{-1}, \quad \mathbf{W}_5 = \mathbf{W}_9 = \mathbf{H}_{sm}$$

$$\mathbf{W}_7 = \frac{0.05s+1000}{s+50}\mathbf{I}, \quad \mathbf{W}_8 = \frac{(s+0.0005)(s+500)}{(s+5)(s+50)}\mathbf{I}$$

where $\mathbf{D}_{scaling}$ is a diagonal matrix that scales the position states to km units to improve the condition number of the tested matrix (numerical stability), and the selector matrix $\mathbf{H}_{sm}$ is used to restrict the norms to terms that contribute to the angular acceleration response. The shapes of these filters are relevant as they reflect the portion of the tracking bandwidth that concerns them: 1) $\mathbf{W}_1$ and $\mathbf{W}_7$ are low-pass filters, 2) $\mathbf{W}_2$ and $\mathbf{W}_3$ are high-pass filters, 3) $\mathbf{W}_8$ is a band-stop filter, and 4) the others are constant gain filters.

4.3.3.3 Mitigation

A modified INDI architecture was proposed in [73] to mitigate the use of non-identical actuators (see Eq.(4.14)). Figure 4.1 shows the classic INDI FL stage architecture while the modification is presented in Fig. 4.2 (where $\mathbf{F}(s) = e^{-s\tau}\mathbf{I} = e^{-sDT_s}\mathbf{I}$). As the on-board plant model is (by default) the classical ideal model $s\hat{\mathbf{P}}(s) = \tilde{\mathbf{G}}_0$, both diagrams are equivalent when using first-order identical actuators, for which the control effectiveness decoupling is ideal and $\bar{\mathbf{G}}_0 = \tilde{\mathbf{G}}_0$. For realistic actuators, the modified INDI should improve decoupling without sacrificing performance. Furthermore, this modification allows the improvement of the on-board plant model, if needed.

4.3.3.4 Applicability Assessment

The applicability tests are evaluated and the results are presented in Table 4.5. Table 4.6 presents the configurations that yield the largest matrix derivatives (used as approximations of the Lipschitz constants $L_a \geq {}^1\!/_\tau\|\mathbf{A}-\mathbf{A}_0\|$ & $L_g \geq {}^1\!/_\tau\|\mathbf{G}-\mathbf{G}_0\|$ and computed using MATLAB's Symbolic Toolbox), with irrelevant values being indicated by $*$. The altitude of 11 km corresponds to the tropopause, as modeled. These configurations represent an abrupt

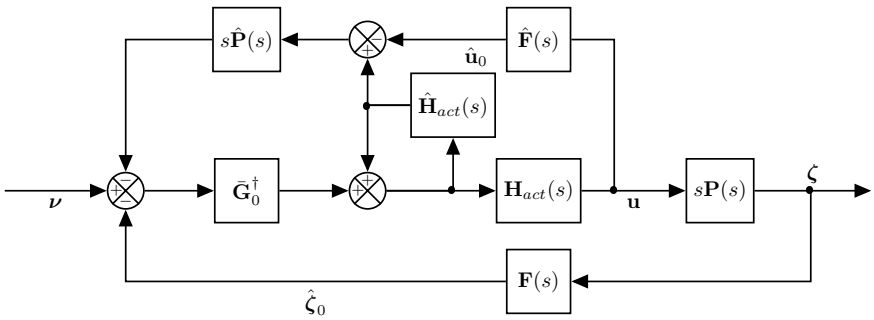

Figure 4.2 Modified INDI Feedback-Linearization Stage

maneuver at high altitude and high speed, which is at the limit of what can be considered normal flight operation.

The INDI control law is thus suitable for the considered application under the following modifications:

1. Control Effectiveness Matrix is modified to account for differing (steady state) actuator bandwidths.
2. Actuator position feedback loop is modified to avoid simplifications relying on the decoupling property (which does not hold for the current application).
3. On-board Actuator model are required and should be accurate.
4. On-board Plant model can be approximated by the classical INDI ideal plant $\hat{\mathbf{P}}(s) = {}^{1}/_{s}\tilde{\mathbf{G}}$.
5. Linear-Controller stage must not assume an ideal (pure integrator) plant, it must be tuned robustly.

The robust tuning of the linear controller can be done by framing $\boldsymbol{\Delta}_a$ and $\boldsymbol{\Delta}_d$ as the uncertainties to be robust against, leading to conservative stability margins, or by using the Structured $\mathcal{H}_\infty$ Synthesis on a collection of accurate linear models of the plant (as will be shown in Section 4.3.5).

4.3.4 Tunable controller architecture - Linear-Controller stage

The architecture of the tunable Linear-Controller stage of the INDI control law is intended, ideally, to shape a decoupled pure integrator system into desired decoupled linear responses.

Table 4.5 Tests results for INDI applicability

Test	Value	Status	Comments
Decoupling	0.4713	Passed (Mitigated)	Modified INDI
Stabilizing	0.3561	Passed	-
Fast Actuation	1.2167	Marginal	Robust tuning will compensate this tracking error.
Slow varying (Locally Lipschitz)	0.9893 0.7915	Passed	Adding $\mathbf{H}_{sm}$ to $\mathbf{W}_{4l}$, to restrict the maximal matrix derivatives to terms contributing to angular accelerations, yields values of 0.1399 and 0.2691, implying that states with slower dynamics are the ones driving this worst case configuration.
Small delay	0.1725	Passed	-
Bandwidth ratio	0.3029	Passed	Effective bandwidth is NOT dominated by the actuator bandwidth, delay requires compensation provided by INDI.
Synchronized	0.9859	Passed	Value for 2% time delay synchronization error.
Compensated	1.3003	Marginal	Robust tuning will compensate this tracking error. Sensitive to synchronization error. Computed for a 2% synchronization error (with $\|\mathbf{E}_i\|_\infty = 0.7931$). For no synchronization error, value becomes 1.0927 (with $\|\mathbf{E}_i\|_\infty = 0.7916$), which is still marginal.
Intersample rippling	0.0361	Passed	Sampling rate is sufficiently high compared to the actuator bandwidth to make this effect negligible.
Discrete delay	0	Passed	-

Table 4.6 Normal flight operation limits and configurations yielding maximal Lipschitz constants

State	Upper Bound	Lower Bound	Worst Case L_a	Worst Case L_g
Altitude (m)	12000	1000	11000	10395
Speed (m/s)	$\begin{bmatrix}286 & 20 & 74\end{bmatrix}^\top$	$\begin{bmatrix}71 & -20 & -74\end{bmatrix}^\top$	$\begin{bmatrix}286 & 20 & 74\end{bmatrix}^\top$	$\begin{bmatrix}286 & 20 & 74\end{bmatrix}^\top$
Act. Pos. (°)	$\begin{bmatrix}30 & 17 & 30\end{bmatrix}^\top$	$\begin{bmatrix}-20 & -30 & -30\end{bmatrix}^\top$	$\begin{bmatrix}29.9 & 17 & -30\end{bmatrix}^\top$	$\begin{bmatrix}* & 17 & *\end{bmatrix}^\top$
Ang. Rate (°/s)	$\begin{bmatrix}45 & 45 & 45\end{bmatrix}^\top$	$\begin{bmatrix}-45 & -45 & -45\end{bmatrix}^\top$	$\begin{bmatrix}-45 & 45 & 45\end{bmatrix}^\top$	$\begin{bmatrix}* & 44.9 & *\end{bmatrix}^\top$
Angles (°)	$\begin{bmatrix}45 & 45 & *\end{bmatrix}^\top$	$\begin{bmatrix}-45 & -45 & *\end{bmatrix}^\top$	$\begin{bmatrix}-10.2 & -45 & *\end{bmatrix}^\top$	$\begin{bmatrix}* & * & *\end{bmatrix}^\top$
Act. Rate (°/s)	$\begin{bmatrix}40 & 30 & 40\end{bmatrix}^\top$	$\begin{bmatrix}-40 & -30 & -40\end{bmatrix}^\top$	$\begin{bmatrix}-39.8 & -30 & 40\end{bmatrix}^\top$	$\begin{bmatrix}* & -30 & *\end{bmatrix}^\top$
Ang. Acc. (°/s^2)	$\begin{bmatrix}90 & 90 & 90\end{bmatrix}^\top$	$\begin{bmatrix}-90 & -90 & -90\end{bmatrix}^\top$	$\begin{bmatrix}90 & -90 & -90\end{bmatrix}^\top$	$\begin{bmatrix}* & -90 & *\end{bmatrix}^\top$
Acc. (m/s^2)	$\begin{bmatrix}1 & 1 & 1\end{bmatrix}^\top$	$\begin{bmatrix}-1 & -1 & -1\end{bmatrix}^\top$	$\begin{bmatrix}1 & 1 & -1\end{bmatrix}^\top$	$\begin{bmatrix}-1 & -1 & -1\end{bmatrix}^\top$
Jerk (m/s^3)	$\begin{bmatrix}1 & * & 1\end{bmatrix}^\top$	$\begin{bmatrix}-1 & * & -1\end{bmatrix}^\top$	$\begin{bmatrix}0.87 & * & -0.97\end{bmatrix}^\top$	$\begin{bmatrix}0.77 & * & -0.93\end{bmatrix}^\top$

As such, a traditional Proportional-Integral-Derivative (PID) controller for each individual axes is favored due to the widespread literature available about them.

The linear controller is chosen as

$$\mathbf{C}(z) = \mathbf{C}_p + \mathbf{C}_i \frac{T_s}{z-1} + \mathbf{C}_d \frac{z-1}{T_f(z-1)+T_s} \tag{4.16}$$

with $\mathbf{C}_x$ being constant diagonal matrix gains to be tuned, and where $T_f = {}^1\!/_{5T_s}$ is the pseudo-derivative constant. Note that the angular acceleration signal will be used directly instead of applying the pseudo-derivative ${}^{(z-1)}\!/_{(T_f(z-1)+T_s)}$ to the angular rate signals, as shown on Fig. 4.3. The pseudo-derivative can be avoided altogether by using a reference model to shape angular rate command signals (acts as a low-pass filter that prevents commands from exceeding the control bandwidth), which would also allow the addition of PCH to help mitigate actuator saturation issue [57].

While recommended, the reference model and PCH are excluded for this paper, as to simplify the intended application example without loss of generality.

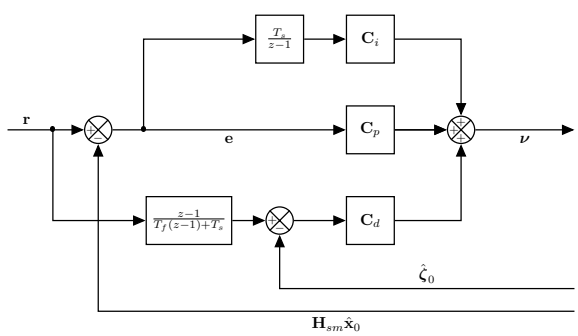

Figure 4.3 Considered INDI Linear-Controller Stage ($\mathbf{C}(z)$)

4.3.5 Robust gain tuning - Structured $\mathcal{H}_\infty$ Synthesis

This section presents how the Structured $\mathcal{H}_\infty$ Synthesis can be applied to the control tuning problem considered. This includes how the requirements are captured through various models and optimization objectives, how restrictions are imposed on the algorithm during optimization, and how the algorithm is used incrementally in order to improve its convergence.

Note that both classic and modified architectures will be tuned using the same synthesis approach for comparison purposes.

4.3.5.1 Objectives and models

The multi-model and multi-objective capabilities of the Structured $\mathcal{H}_\infty$ Synthesis are exploited through an appropriate framing of the optimization problem.

First, the requirements are enforced through a combination of optimization objectives and associated (virtual) weighting filters. Tracking and decoupling can be combined into a single multivariate (MIMO) tracking objective or provided as a group of 9 univariate (SISO) objectives that are optimized simultaneously, both of which are supported by MATLAB's `TuningGoal` class. Unfortunately, most objectives supported by the `TuningGoal` class will first verify the system's stability, which is problematic for aircraft as they exhibit marginally unstable modes (such as spiral and phugoid) that are tolerated as they are handled by human pilots or autopilot outer loops. As such, all objectives are formulated using `WeightedGain` (or the similar `Gain` for constant weighting filters), which includes no underlying verification and thus allows the most customization options.

Second, new virtual models are generated by adding virtual connections to create the relevant signals for the assessment of objectives, which are then filtered by the virtual weightings. The combined objective (or group of objectives) of tracking and decoupling requires the simple addition of a tracking error output filtered by $\mathbf{W}_t$. The control effort objective simply adds the pseudo-command as an output and limits its gain w.r.t. the input with $\mathbf{W}_u$. The stability margin requirement is captured by a disk margin [95] objective that requires the input and output of an hypothetical uncertainty block shown in Fig. 4.4 (where the Feedback-Linearization stage acts as the plant for the Linear-Controller stage), which corresponds to the balanced sensitivity function (using zero disk eccentricity). The weighting filter $\mathbf{W}_{dm}$ for this objective will simply be the desired minimal disk radius α_{dm}.

Third, the objectives are split into "soft" and "hard" goals, which serve to indicate to the algorithm whether a given objective is to be treated as a constraint to fulfill first or a minimization target to be optimized afterwards. The stability margin objective is chosen as the only constraint to be respected (but not optimized) before the other objectives are tackled to seek the best performance achievable (under the constraint of sufficient stability margins). This also means that the algorithm will abandon an optimization run if it cannot meet the stability margins requirement, indicating a problematic configuration that will require further investigation. When multiple soft goals are provided, the algorithm will employ virtual scaling to make the cost of one objective larger than the other, thereby focusing improvement for that specific goal. Scaling is changed between iterations as to progressively optimize all objectives designed as soft goals.

The weighting filters used are:

$$\mathbf{W}_t = \begin{bmatrix} \frac{s+2/4}{1.05s+0.05*2/4} & \frac{1}{0.01} & \frac{1}{0.1} \\ \frac{1}{0.01} & \frac{s+2/5}{1.2s+0.05*2/5} & \frac{1}{0.01} \\ \frac{1}{0.1} & \frac{1}{0.01} & \frac{s+2/8}{1.05s+0.05*2/8} \end{bmatrix}, \quad \mathbf{W}_u = \frac{1}{10}, \quad \mathbf{W}_{dm} = \alpha_{dm} = 0.8284$$

where $\alpha_{dm} = 2\max((GM-1)/(GM+1), \tan(PM/2))$ is the minimal disk radius, with "GM" and "PM" denoting the desired Gain and Phase Margins (2 or 1/2 and $\pm\pi/4$ respectively, corresponding to ± 6 dB and $\pm 45°$). Note that while the tracking weighting filter is a matrix (consistent with a MIMO objective), each transfer is evaluated separately (SISO objectives are used to reduce conservatism); this results in 10 soft goals and one hard goal being used in the synthesis.

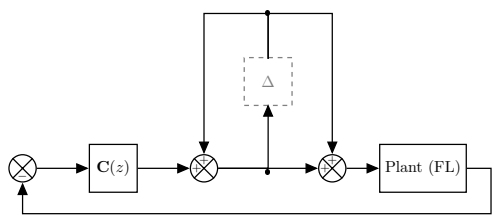

Figure 4.4 Balanced disk margin assessment diagram

4.3.5.2 Restrictions

Further restrictions are imposed on the synthesis to improve algorithm convergence to acceptable solutions: the controller gains must all be positive and limited to 10 for proportional gains and 5 for integral and derivative gains, and all requirements are to be enforced on the frequency range of $(10^{-5}, 10^{5})$ rad/s. This frequency range was selected to exclude numerical errors at very low frequencies, which are meant to be regulated by outer loops in any case.

4.3.5.3 Sequential approach

The tuning process will make several calls to the `systune` function of MATLAB, which represent several runs of the algorithm. Each run use a subset of the models considered, and includes all models used in previous runs. This means that additional flight conditions are added to the synthesis to progressively improve the gain from the ideal, to nominal, to all cases. This allows for better convergence of the algorithm for the initial guess provided, but

additional randomized runs are also included due to the non-smooth nature of the optimization problem.

Additionally, the intermediate models presented in Section 4.3.2.2 are considered, as to progressively add the effect of various non-ideal hardware parameters. As each subsequent model yields worse performances than the last, requirements are expected to be fulfilled by all models. As such, the collection provided to the algorithm contains several models for each flight condition, augmented with virtual weighting filters and associated virtual connections (such as the one shown by Fig. 4.4). Padé approximants of order 8 are used to convert delayed continuous systems into rational transfer functions. The final call to the `systune` function contains the discrete models best describing the linear system (i.e. a discretization of Eq.(4.13) using the Backwards-Euler method with an augmented plant [73]), but excludes all continuous models due to `systune`'s limitations. The discrete models have their gains initialized to the best gains found in the previous tuning iteration on all considered continuous models.

4.3.5.4 Results

The results of the synthesis are presented in Table 4.7 and Table 4.8, while Figure 4.5 presents the tuned closed-loop step responses. As expected, the costs are generally lower for the modified architecture, especially for decoupling. The classical architecture have lower integral gain values and (mostly) higher derivative gain values, while the modified architecture employs lower proportional & derivative gain values for the roll axis and higher gain values for the yaw axis. These differences are signs that the algorithm struggles to enforce the stability margins constraint for the classic architecture and that performance was sacrificed, which is confirmed by the fact that the decoupling requirement was only met by the modified architecture. Linear responses show how the modified architecture improved decoupling, the rise time of the pitch and yaw rates, and the roll rate overshoot.

Since these controllers were tuned for linear approximation of the aircraft at specific flight conditions, a nonlinear assessment of the requirements over the entire flight envelope is required to have a better picture of the performance achieved.

4.4 Verification

The tuned controllers must be tested on the nonlinear system to assess their performance and the fulfillment of requirements. This verification is done using the WCAT to carry out various simulation campaigns. The toolbox uses cost functions to measure performance

Table 4.7 Synthesis results - Costs

Goal	Type	Classic	Modified
Tracking (& Decoupling)	Soft	$\begin{bmatrix} 1.0164 & 0.0002 & 1.3499 \\ 0.0001 & 0.9084 & 0.0000 \\ 1.3499 & 0.0000 & 1.0995 \end{bmatrix}$	$\begin{bmatrix} 1.0031 & 0.0016 & 0.9012 \\ 0.0000 & 0.9431 & 0.0000 \\ 1.0659 & 0.0000 & 1.0659 \end{bmatrix}$
Control effort	Soft	0.2980	0.3907
Stability margin	Hard	1.0000	0.9996

Table 4.8 Synthesis results - Gain values

Gain	Classic			Modified		
	p	q	r	p	q	r
$\mathbf{C}_p$	1.9040	3.1441	0.8755	1.2894	3.6599	1.1074
$\mathbf{C}_i$	0.0000	0.3193	0.0178	0.1207	0.8149	0.0614
$\mathbf{C}_d$	0.1546	0.2168	0.0817	0.0181	0.1681	0.1502

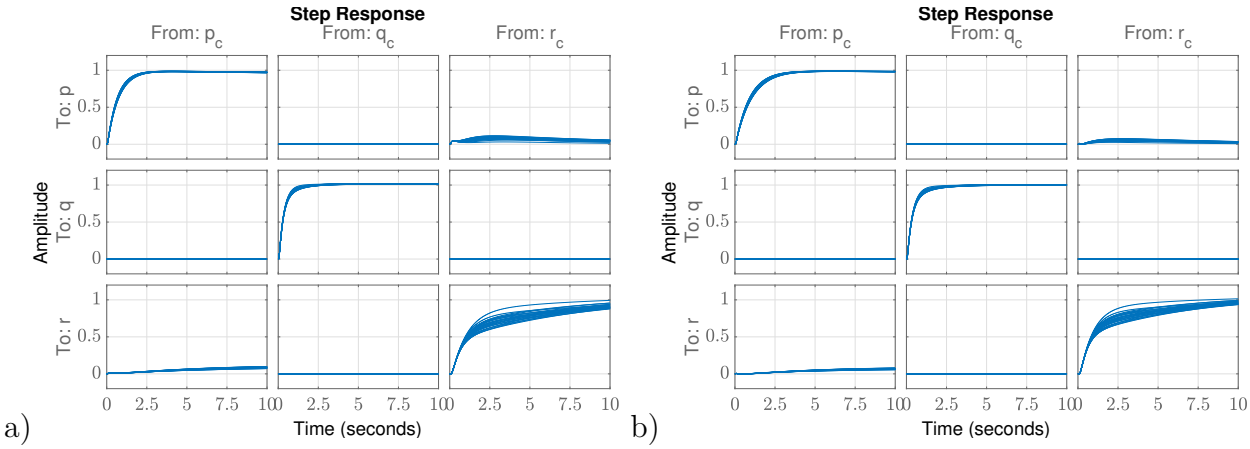

Figure 4.5 Linear Controller Stage response: a) Classic and b) Modified architectures

and includes various algorithms to seek the global minimum (worst case). The toolbox also supports the traditional Monte Carlo method, which can be used to compare and contrast with a namesake Worst Case analysis.

4.4.1 Worst-Case Analysis Toolbox

This section presents the WCAT-II MATLAB toolbox [70, 71], developed from the University of Exeter with ESA supporting grant ESTEC (Contract No: 19783/06/NL/JD:4000104541). This second version of the toolbox is based on previous work by the University of Leicester [72].

Traditional V&V uses MC simulation campaign to assess the nonlinear system's fulfillment

of requirements in a probabilistic manner; the ratio of failures and successes is assumed to be proportional to the risk of failure of the aircraft. While this approach is practical (easy & simple to implement & carry out), it cannot provide guaranteed proof and may require massive amounts of computations to achieve the desired level of confidence.

In face of these drawbacks, alternative approaches using optimization-based worst-case search algorithms were developed. The WCAT contains a variety of algorithms ranging from local & global optimization (including hybrid methods), gradient & non-gradient based search, surrogate modeling & Bernstein expansion, and more.

The Differential Evolution (DE) algorithm [70, 96, 97] was selected as a compromise between MATLAB toolbox license requirements and the optimization performance. The DE algorithm can be described in 5 steps:

1. *Random Initialization*: Generate trial vectors randomly (as with MC), evaluate them, and assign them as the first generation of target vectors. The vectors' elements are values of the selected uncertain parameters defining a test case.

2. *Mutation*: Generate mutated vectors from the latest generation of target vectors $\bar{x}_n = x_k + F_m(x_i - x_j)$ with x_i, x_j, x_k being 3 vectors picked at random from the target vectors, $\bar{x}_n$ being the newly created mutated vector, and F_m a selected "step-size" parameter ($x_i - x_j$ being the direction of mutation).

3. *Crossover*: Generate new trial vectors ($\hat{x}_n$) via crossover between mutated and target vectors, where each entry of every trial vector is randomly assigned either the mutated (x_n) or target ($\bar{x}_n$) vector's entry value depending on a selected probability parameter $\rho_c \in [0, 1]$.

4. *Evaluation & Selection*: Evaluate and compare the trial vectors' results with the target one; keep the "worst" member cases as the new generation of target vectors.

5. *Termination*: Repeat steps 2 to 4 until termination.

The termination conditions offered by the toolbox are either fixed, stopping after a set number of generations, or adaptive, stopping once the improvement from a generation to the next falls under a given threshold. Fixed termination conditions are used in this paper as to ensure all campaigns use the same total number of simulations.

The only uncertain parameters used for these vectors are altitude and speed with uniform probability distributions over the entire flight envelope. To avoid explicitly constraining the parameters to the flight envelope, the speed parameter was chosen to be a dimensionless

value between 0 and 1 to be converted into the corresponding Mach number at the given altitude during simulation.

Both Monte Carlo and Worst Case approaches will be used to compare and contrast their respective contributions to the verification process.

4.4.2 Nonlinear requirement assessment

In order to verify that the control law fulfills the requirements of Section 4.3.1, said requirements need to be captured by cost functions. Since the selected $\mathcal{H}_\infty$ norms used during synthesis already fulfill this role, they serve as starting points for the elaboration of cost functions that need to accept time series as input and give a scalar as output.

4.4.2.1 Tracking

The tracking requirement is assessed using a sine-chirp frequency-sweep command [98] as the simulation input and by applying a Fast-Fourier Transform (FFT) to the simulation output. This provides a frequency response of the nonlinear system that can be used to compute a $\mathcal{H}_\infty$ norm as the cost function, allowing the comparison of nonlinear results against the cost values obtained by the controller synthesis in Table 4.7. The reference amplitude is $1°/s$ as to avoid hitting any saturation or rate limits. The tracking error between the reference and the plant output is filtered using the tracking weighting filter $\mathbf{W}_t$. This filtered tracking error is the simulation output, which is post-processed to retrieve the peak error for each of the 9 transfers. The cost function's output is then the maximum error between said 9 transfers.

4.4.2.2 Stability Margins

The margin requirement is assessed using a sine-chirp frequency-sweep command [98] as the simulation input, on the modified model of Fig. 4.4. This modified model injects the input chirp disturbance at the output of the disk margin uncertainty block (removed, shown as hashed Δ block) and returns the balanced sensitivity transfer function's response as the simulation output (input of the uncertainty block). The balanced sensitivity transfer function's frequency response (filtered by $\mathbf{W}_{dm}$) is obtained through the FFT from which the peak gain can be extracted as the cost function's output.

4.4.3 Results

In total, 8 simulation campaigns were carried out for each combination of architectures (classic and modified), requirements (tracking and stability margins), and analysis approaches (Monte Carlo and Worst Case (DE)). Figures 4.6 and 4.7 present the cost values of each simulation and offer a general overview of the results, while Tables 4.9 and 4.10 present specific relevant numerical results. Additionally, Figures 4.8 and 4.9 present the parameter space to better appreciate how the cost values are spread out for Monte Carlo simulations and how they pool around local minima for the Worst Case approach, with Figures 4.10 and 4.11 showing the same information on the flight envelope instead as a more intuitive portrait of the situation.

The most striking difference between the classic and modified architectures is the range and spread of cost values across the various simulations (flight conditions); the modified architecture achieves an acceptable level of performance consistently, indicating a good robustness to the plant variations as expected from a (properly implemented) INDI controller.

The DE algorithm performs a global optimization and has successfully identified the global minimum despite most of the last generation's population ending up near a different local minimum at a different corner of the envelope.

The worst cases for the modified architecture occur at the corners of the flight envelope, which indicates good nominal performance across the envelope. Worst case for tracking is near the lowest speed and altitude, which is often a flight condition difficult to control.

Poorer stability margins at the operational ceiling of the aircraft were also expected, with lower speeds being most sensitive to perturbations. This discourages abrupt maneuvers at high altitudes, especially considering that it also coincides with the estimated maximal Lipschitz constant of the envelope (see Table 4.6).

The fact that the classic architecture's controller achieves better tracking at the edge and

Table 4.9 Monte Carlo Analysis results

		Tracking		Stability Margin	
		Classic	Modified	Classic	Modified
Worst Case:	Altitude (m)	5531	1056	11905	11955
Worst Case:	Speed (Ma)	0.7742	0.2793	0.7007	0.7057
Worst Case:	Cost	-1.3545	-1.0930	-1.2482	-1.2602
Best Case:	Cost	-1.0032	-1.0540	-1.1815	-1.2222
	Mean Cost	-1.0981	-1.0636	-1.2094	-1.2376
Cost Standard Deviation		0.1026	0.0070	0.0160	0.0083

Table 4.10 Worst Case Analysis results

		Tracking		Stability Margin	
		Classic	Modified	Classic	Modified
Worst Case:	Altitude (m)	5682	1024	11979	12000
Worst Case:	Speed (Ma)	0.7840	0.2566	0.7027	0.7005
Worst Case:	Cost	-1.3587	-1.1017	-1.2489	-1.2608
Mean Last Gen.:	Altitude (m)	5303	8815	11108	10998
Mean Last Gen.:	Speed (Ma)	0.7502	0.7258	0.7369	0.7306
Mean Last Gen.:	Cost	-1.3252	-1.0788	-1.2403	-1.2554

corners of the flight envelope, where the modified architecture's controller is at its worst, hints at the beneficial effect of the realistic actuators' design. The lack of consistency for the classic architecture (high standard deviation for costs), on the other hand, confirms that poor decoupling at the FL stage can hinder the achievable performance by the linear controller.

While the tracking costs are similar to the ones obtained on linear models, the stability margins have been degraded due to the neglected feedback-linearization error ϵ_l. A cost value of -1.26 corresponds to a (conservative) Gain and Phase margins of ± 5.98 dB and $\pm 36.67°$ respectively.

These figures also show how the Worst Case approach increases the density of cases around local minima, which helps identifying the (global) worst case with better precision. For instance, the Monte Carlo approach may have missed the Modified INDI's tracking worst case if a smaller number of simulations would have been chosen.

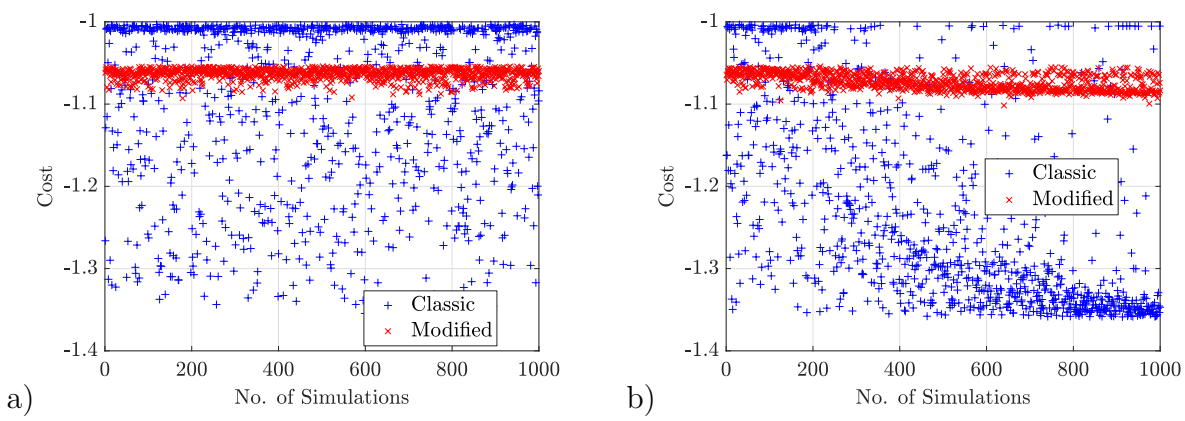

Figure 4.6 Nonlinear tracking assessment - Cost values:

a) Monte Carlo approach and b) Worst Case approach

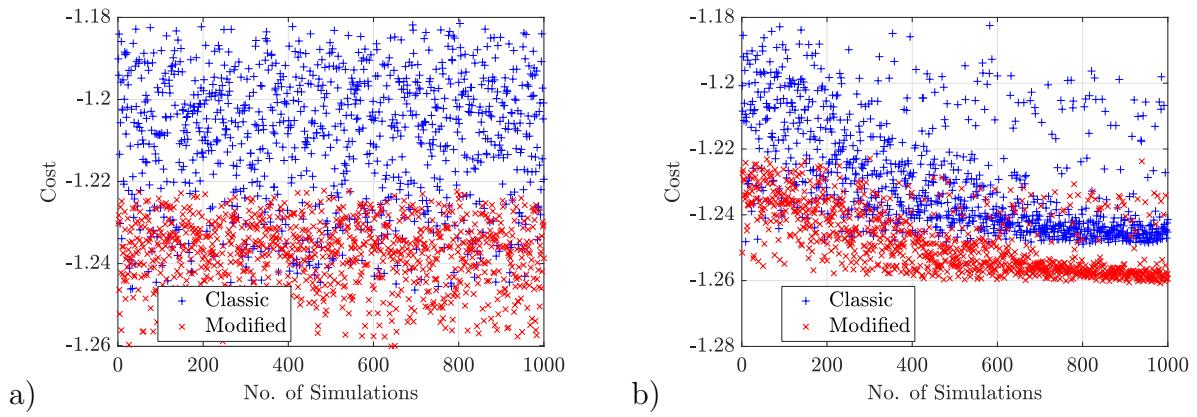

Figure 4.7 Nonlinear stability margin assessment - Cost values:

a) Monte Carlo approach and b) Worst Case approach

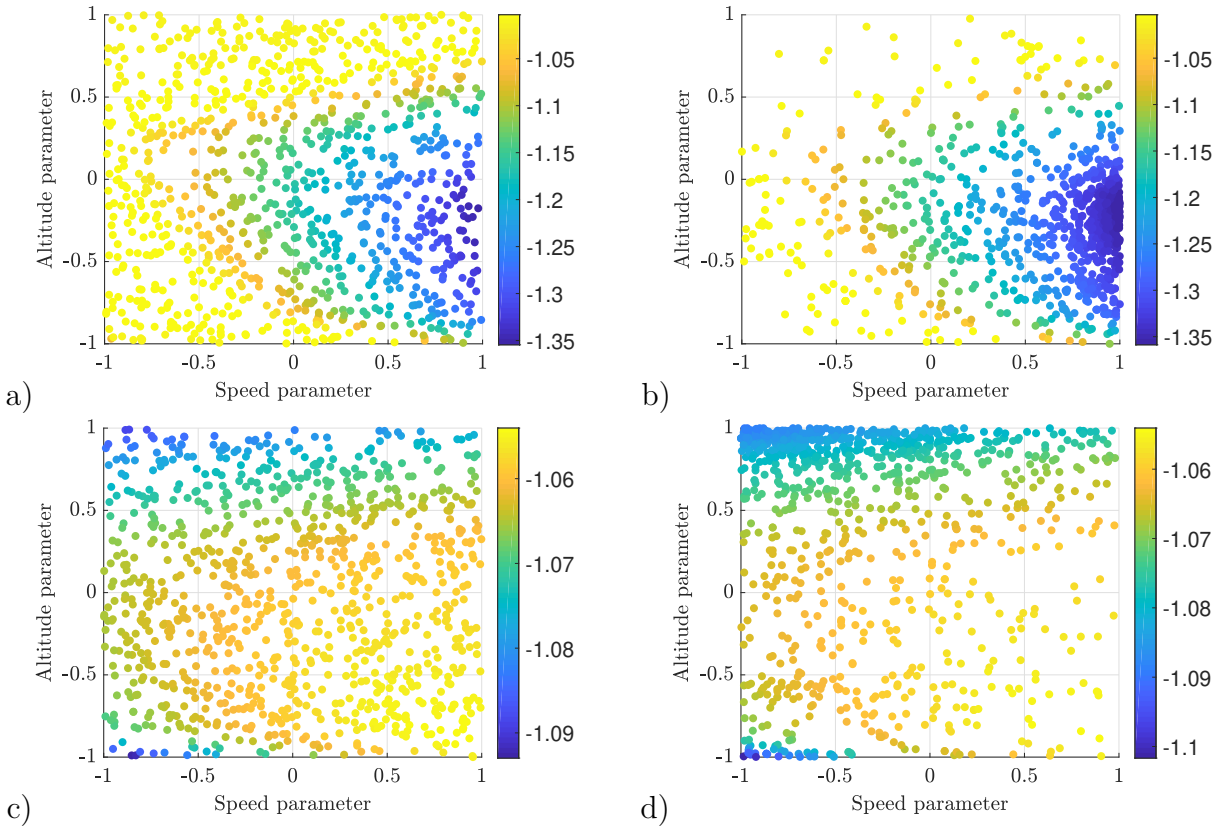

Figure 4.8 Nonlinear tracking assessment - Parameter space:

a) MC approach with Classic controller, b) WC approach with Classic controller, c) MC approach with Modified controller, and d) WC approach with Modified controller

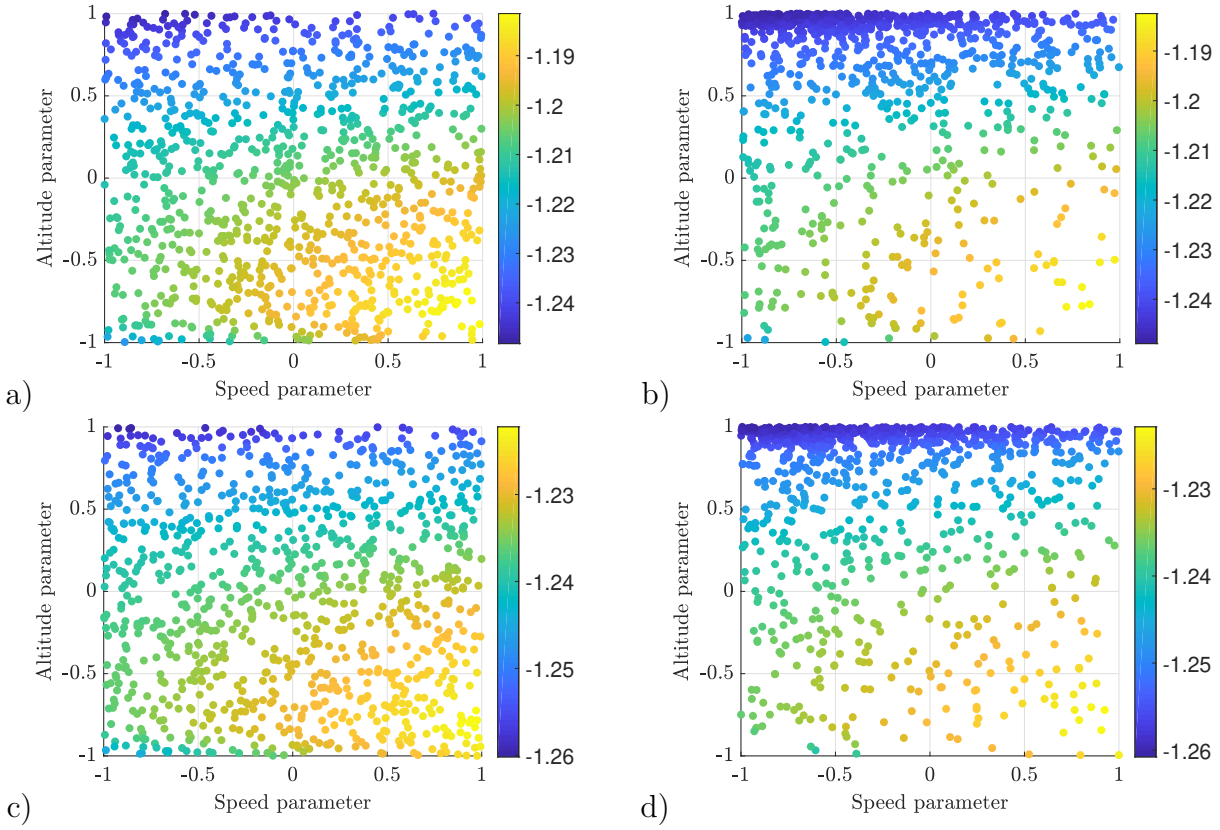

Figure 4.9 Nonlinear stability margin assessment - Parameter space:

a) MC approach with Classic controller, b) WC approach with Classic controller, c) MC approach with Modified controller, and d) WC approach with Modified controller

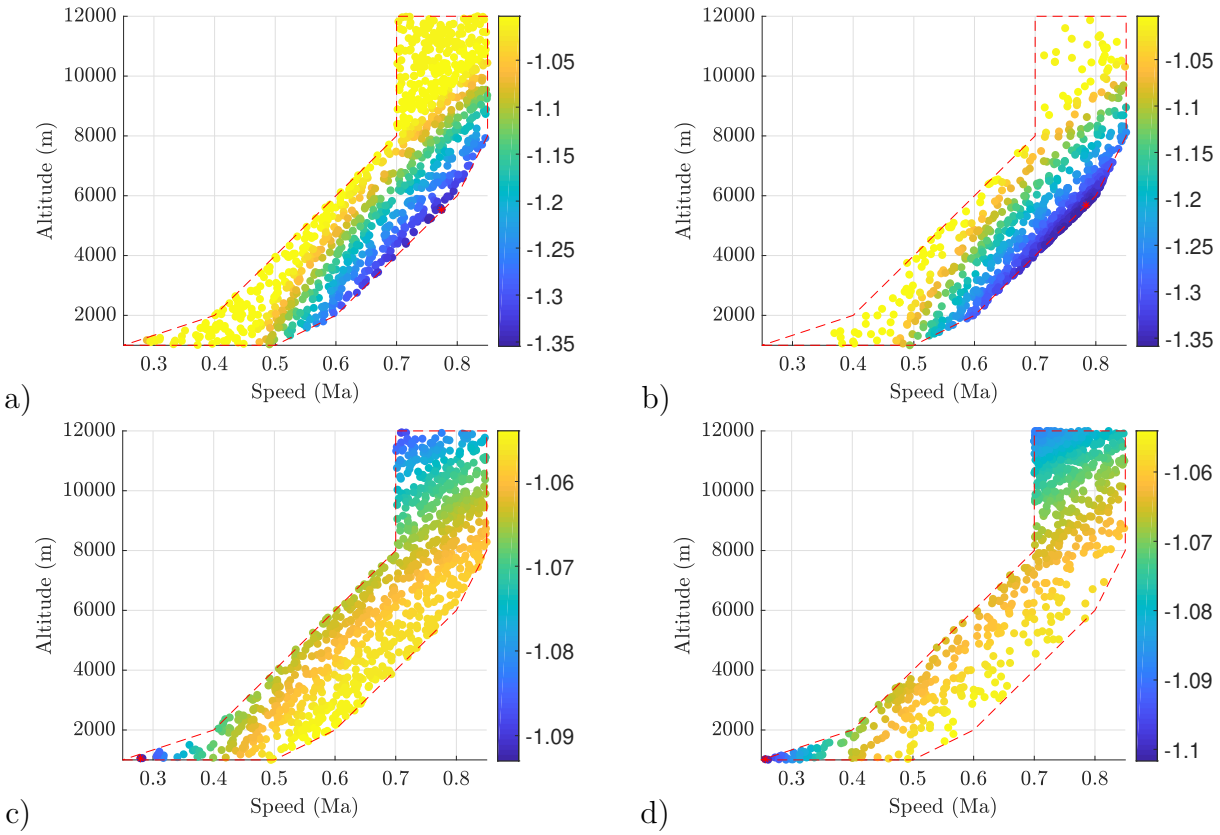

Figure 4.10 Nonlinear tracking assessment - Flight Envelope:

a) MC approach with Classic controller, b) WC approach with Classic controller, c) MC approach with Modified controller, and d) WC approach with Modified controller

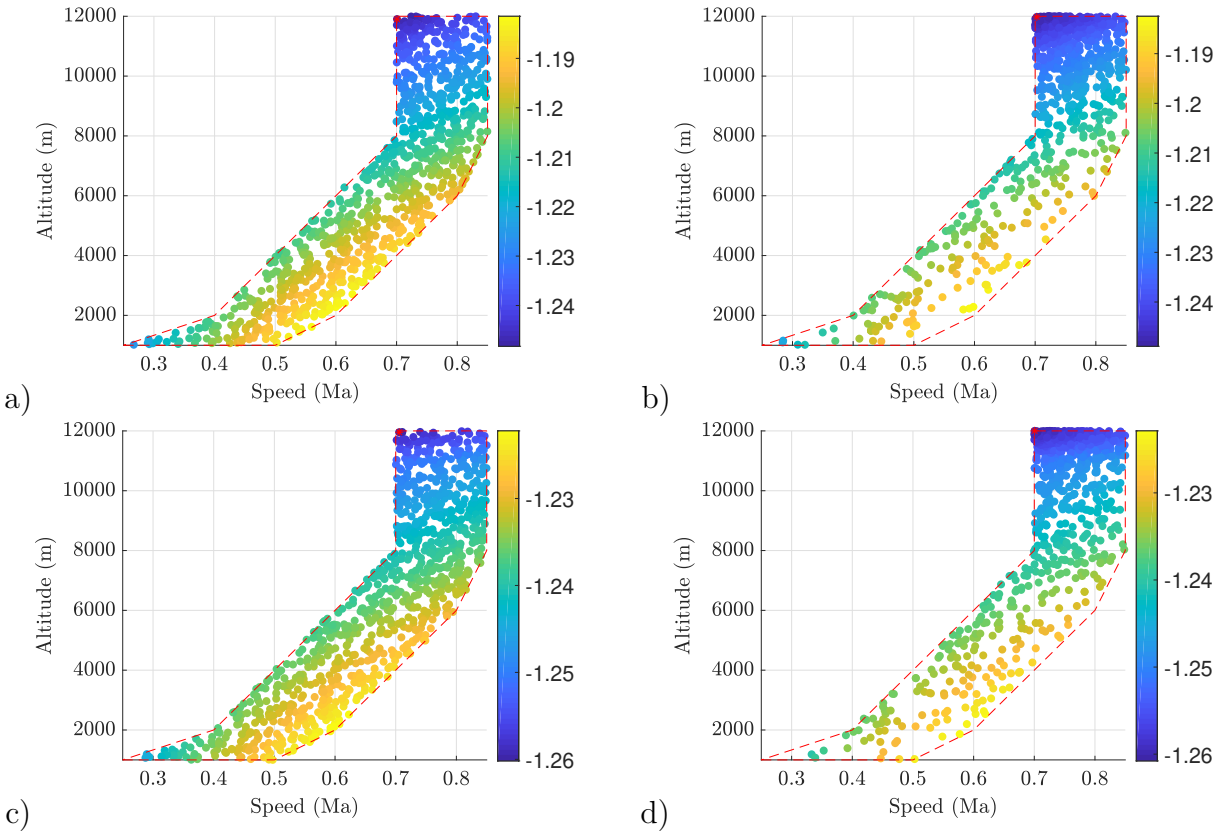

Figure 4.11 Nonlinear stability margin assessment - Flight Envelope:

a) MC approach with Classic controller, b) WC approach with Classic controller, c) MC approach with Modified controller, and d) WC approach with Modified controller

4.5 Conclusion

The modified INDI controller proposed in [73] was applied to a common civil transport aircraft, robustly tuned using the Structured $\mathcal{H}_\infty$ Synthesis, and its performance was verified with the use of the WCAT-II toolbox in MATLAB. The applicability of the INDI control law was assessed using the proposed tests from [73], demonstrating how classic INDI assumptions are conservative and how the use of realistic actuators can be mitigated. The linear controller was robustly tuned with success, even when using the unmitigated classic INDI architecture (albeit with lesser performance and predictable decoupling issues), demonstrating its usefulness when robustness to model variations is not guaranteed at the Feedback-Linearization stage. Performance was assessed through nonlinear simulations using peak gains in the frequency domain, which is equivalent to the $\mathcal{H}_\infty$ norms used during linear synthesis and allows for better comparison of the results. The analysis was done using both Monte Carlo and Worst Case (DE algorithm) approaches to get a richer view of how the performance metrics vary over the flight envelope.

In sum, a novel controller design process was presented and shows promise as an efficient alternative to the time-consuming traditional controller design, which is centered around gain-scheduled PID controllers that are manually tuned and verified through extensive Monte Carlo campaigns.

Future works should explore adding uncertainties to various parameters, such as weight, position of the center of mass, inertia, and aerodynamic coefficients. Including unmodeled dynamics as uncertainties should also be considered as a way to ensure better robustness against the feedback-linearization error ϵ_l. Application of INDI to flexible aircraft [39] should also be revisited.

CHAPTER 5 ARTICLE 3: ROBUST CONTROL FOR FLEXIBLE AIRCRAFT USING INCREMENTAL NONLINEAR DYNAMICS INVERSION WITH NOTCH FILTERS

Frédérick Laliberté, David Saussié
Published in: Journal of Guidance, Control, and Dynamics
Submission date: March 31th 2025

Abstract

This paper proposes the use of the Incremental Nonlinear Dynamic Inversion control scheme as an alternative to the common flexible aircraft control strategy used in the industry, namely a gain-scheduled linear controller with notch filters. The work is based on the refinement of applicability conditions of the Incremental Nonlinear Dynamic Inversion control, as well as the demonstration of its use on a rigid aircraft with realistic hardware compensated by a robust tuning via the Structured $\mathcal{H}_\infty$ Synthesis. The concept of singular perturbations is used to link the spillover effects, which are central to aeroservoelastic notch filter design for flexible aircraft, with the known robustness properties of the proposed control scheme. The aeroservoelastic filters and the angular rate linear controller are robustly tuned using the Structured $\mathcal{H}_\infty$ Synthesis, and requirement fulfillment is assessed via limited Monte Carlo simulation campaigns over the considered flight envelope.

Nomenclature

$\mathbf{A}$	=	State matrix (State-Space model)
$\mathbf{B}$	=	Input matrix (State-Space model)
$\mathbf{C}$	=	Output matrix (State-Space model)
$\mathbf{D}$	=	Feedthrough matrix (State-Space model)
$\mathbf{E}$	=	Descriptor matrix (State-Space model)
$\mathbf{x}$	=	State signal vector
$\mathbf{y}$	=	Output signal vector
$\mathbf{u}$	=	Input signal vector
$\mathbf{G}$	=	Control effectiveness matrix
ν	=	Pseudo-command vector
$\mathbf{H}_{sm}$	=	Selector matrix of measured outputs meant to track pseudo-commands

T_s	=	Sampling period
ϵ	=	Singular parameter
σ	=	Singular value
ω	=	Angular frequency
$\mathbf{\Delta}$	=	Uncertainty matrix
α	=	Angle of attack
β	=	Angle of sideslip
ϕ	=	Bank angle
θ	=	Pitch angle
ψ	=	Azimuth angle
p	=	Roll rate
q	=	Pitch rate
r	=	Yaw rate
δ_a	=	Aileron deflection angle
δ_e	=	Elevator deflection angle
δ_r	=	Rudder deflection angle
$\bar{q}$	=	Dynamic pressure
$\mathbf{\Phi}$	=	Bending modeshape matrix
$\mathbf{\Psi}$	=	Torsion modeshape matrix
$\mathbf{q}$	=	Flexible mode vector for bending
ξ	=	Flexible mode vector for torsion
$\mathbf{r}$	=	Mass element's position vector (relative to Fuselage frame)
$\mathbf{r}_f$	=	Fuselage's mass element's position vector (relative to Fuselage frame)
$\mathbf{r}_{fw}$	=	Position vector of the origin of the Wing frame (relative to Fuselage frame)
$\mathbf{F}_{eng}$	=	Engine thrust vector (relative to Fuselage frame)
$\mathbf{r}_{eng}$	=	Engine position vector (relative to Fuselage frame)
$\mathrm{d}m_w$	=	Wing mass element
$\mathbf{f}_w$	=	Force by unit span vector acting on a wing mass element (relative to the wing's local frame)
$\mathbf{r}_w$	=	Wing mass element's position vector (relative to the wing's local frame)
C_L	=	Lift aerodynamic coefficient
C_D	=	Drag aerodynamic coefficient
C_m	=	Pitch moment aerodynamic coefficient

5.1 Introduction

New advances in aircraft design, such as the use of composite materials, have led to performance improvements that help reduce fuel consumption, and thus operating costs in the context of an adverse economy. These changes have also brought light to the issue of structural dynamics, which invalidates the usual rigid-body assumptions used during flight control design given that dynamic flexible deformations have become significant. Traditional methods such as gain-scheduling may become increasingly tedious and costly to develop in this context, which prompts researchers to look into better approaches.

The study of dynamic interactions between aerodynamic forces, flexible structures, and control surfaces has been ongoing for decades under the term of aeroservoelastic (ASE) effects. Of note are the characterization and modeling of unsteady aerodynamics by [8], NASA's Benchmark Active Controls Technology (BACT) [1] and integrated approach to flexible aircraft modeling using the strip theory [4], along with computed-aided design approaches (such as the vortex and doublet lattice methods [99]) used in conjunction with finite element models of the structure [36]. Mitigation of ASE effects were investigated alongside their characterization, and while other techniques were explored [100–103], notch filters [3, 43] remain the most popular in the industry due to its simplicity and ease of implementation.

Recent advances in incremental techniques have yielded the Incremental Nonlinear Dynamic Inversion (INDI) control, which shows potential in replacing the traditional gain-scheduling technique by an unique linear controller that can handle the entire flight envelope. This is done at the cost of a frequently updated control effectiveness matrix, implementable as a look-up table, and reliance on hardware performance from actuators, sensors and sampling rate. This control approach has proved successful in a variety of aerospace vehicles [57,66,104,105], including flexible aircraft [39, 106] and launcher using ASE filters [107]. Unfortunately, a barrier for the adoption of INDI by the industry is the qualitative nature of its assumptions, which is being gradually brought down as the technology matures. For instance, the stability and robustness properties of INDI were refined in [6], which addressed errors in the literature, while [73] provided quantitative applicability tests along with possible mitigation techniques to address practical shortcomings.

Modern controller tuning methods offer solutions to the problem of designing an unique robust controller for an uncertain system. For instance, [74] demonstrated the use of the Structured $\mathcal{H}_\infty$ Synthesis on a rigid aircraft using INDI with a slew of mitigating modifications to account for the several qualitative assumptions that could not be met. This paper intends to extend this demonstration to flexible aircraft using ASE filters, instead of using additional sensors

and a very high sampling frequency [39].

The following pages will present technical preliminaries concerning INDI, ASE filtering, and the Structured $\mathcal{H}_\infty$ Synthesis. It is followed by the flexible aircraft nonlinear model used, before tackling control design. Finally, the proposed controller's performance will be verified through Monte Carlo nonlinear simulation campaigns.

5.2 Technical Preliminaries

This section presents core concepts that will be applied to the control of a flexible aircraft.

5.2.1 Incremental Nonlinear Dynamic Inversion

Consider a nonlinear system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where $\mathbf{x}, \mathbf{f} \in \mathbb{R}^n$, and $\mathbf{u} \in \mathbb{R}^p$. Using a first-order Taylor series expansion around the time of the last measurement (t_0) [76], the system can be expressed as:

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(t_0) + \mathbf{A}(t_0)(\mathbf{x}(t) - \mathbf{x}(t_0)) + \mathbf{G}(t_0)(\mathbf{u}(t) - \mathbf{u}(t_0)) + \mathcal{O}\Big((\mathbf{x}(t) - \mathbf{x}(t_0))^2, (\mathbf{u}(t) - \mathbf{u}(t_0))^2\Big) \tag{5.1}$$

where $\mathbf{A}(t_0) = \mathbf{A}_0 = {}^{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}/_{\partial \mathbf{x}}\big|_{t_0}$ and $\mathbf{G}(t_0) = \mathbf{G}(\mathbf{x}(t_0), \mathbf{u}(t_0)) = \mathbf{G}(\mathbf{x}_0, \mathbf{u}_0) = \mathbf{G}_0 = {}^{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}/_{\partial \mathbf{u}}\big|_{t_0}$ are Jacobian matrices. This expansion provides a linear approximation

$$\dot{\mathbf{x}}(t) \approx \dot{\mathbf{x}}_0 + \mathbf{G}_0(\mathbf{u}(t) - \mathbf{u}_0) \tag{5.2}$$

for systems with slow dynamics compared to their control bandwidth ($\mathbf{x} - \mathbf{x}_0 << \mathbf{u} - \mathbf{u}_0$) and small synchronized measurement delays ($t \approx t_0 \Rightarrow \mathcal{O}((\mathbf{x} - \mathbf{x}_0)^2, (\mathbf{u} - \mathbf{u}_0)^2) \approx 0$).

Based on this linear approximation, the following control law will achieve feedback linearization ($\mathbf{H}_{sm}\dot{\mathbf{x}} \approx \nu$):

$$\mathbf{u}(t) = \mathbf{u}_0 + \tilde{\mathbf{G}}_0^\dagger(\nu(t) - \mathbf{H}_{sm}\dot{\mathbf{x}}_0) \tag{5.3}$$

where the pseudo-command ν is to be provided by a linear controller, where $\tilde{\mathbf{G}}_0 = \mathbf{H}_{sm}\mathbf{G}_0 \in \mathbb{R}^{p\times p}$ is a square matrix representing the on-board model of the control effectiveness matrix $\mathbf{G}$ between the input $\mathbf{u}$ and the selected measured derivatives $\mathbf{H}_{sm}\dot{\mathbf{x}}$, and where $\dagger$ denotes the Moore-Penrose pseudo-inverse.

The INDI contol law shows promise in replacing a gain-scheduled controller by a unique linear controller for all flight conditions, as its feedback-linearization stage offers robustness against aerodynamic variations. In order to achieve this robustness, the following assumptions were made:

1. Actuators are quasi-instantaneous and identical.
2. State derivative sensors have a high sampling rate and a small delay.
3. All measurement delays are synchronized.
4. Plant dynamics are sufficiently slow w.r.t. controller bandwidth and sampling frequency.
5. Modeling of the control effectiveness matrix $\mathbf{G}_0$ is accurate and updated in real-time.

It was shown in [73] that these qualitative assumptions are conservative and translate into requirements on the hardware that can be assessed via applicability tests. Failed applicability tests can be mitigated in order to fulfill requirements and achieve an approximate feedback linearization, albeit at the cost of a loss of performance that will need to be compensated by the linear controller.

It was also explained in [6] that neglected high-frequency dynamics, such as flexible modes or actuator dynamics, can be considered as singular perturbations. Unlike the regular perturbations that INDI has been shown to be robust against, a singular perturbation is a perturbation that increases the order of the system. A singularly perturbed system admits the following decomposition:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u}) \tag{5.4}$$

$$\epsilon\dot{\mathbf{y}} = \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{u}) \tag{5.5}$$

where ϵ is the singular parameter of the perturbation and is usually defined as the ratio of bandwidths between the slow (nominal) and fast (residual) parts of the model. When $\epsilon \to 0$, the above equation reduces to the nominal (unperturbed) system. According to Tikhonov's theorem [41], there exist a ϵ_{max} where a controller designed to stabilize the nominal system will also stabilize the singularly perturbed system if $\epsilon < \epsilon_{max}$. This threshold is referred to as the Singular Perturbation Margin (SPM) in [108]. Methods for assessing and increasing the SPM of a given system is an ongoing area of research.

5.2.2 Aeroservoelastic filtering

Flexible systems are infinite-dimensional (or distributed parameter) systems, so a reduced-order model is required in order to design a finite-dimensional controller. This usually involves decomposing the system into a finite-dimensional nominal system (e.g. rigid dynamics) and

an infinite-dimensional residual system (e.g. structural dynamics or flexible modes). A suitable choice is to decompose singular perturbations [101].

Consider the following infinite-dimensional flexible system admitting a singular perturbation formulation:

$$\mathbf{E}(\epsilon)\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5.6}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{5.7}$$

where $\mathbf{E}(\epsilon)$ has a bounded inverse when $\epsilon > 0$, but $\mathbf{E}(0)$ is singular. Then, using suitable projection matrices, the state vector $\mathbf{x}$ can be decomposed into a nominal finite-dimensional state vector $\mathbf{x}_n$ and a residual infinite-dimensional state vector $\mathbf{x}_r$ acting as a singular perturbation, resulting in the following system:

$$\dot{\mathbf{x}}_n = \mathbf{A}_n\mathbf{x}_n + \mathbf{A}_{nr}\mathbf{x}_r + \mathbf{B}_n\mathbf{u} \tag{5.8}$$

$$\epsilon\dot{\mathbf{x}}_r = \mathbf{A}_{rn}\mathbf{x}_n + \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{u} \tag{5.9}$$

$$\mathbf{y} = \mathbf{C}_n\mathbf{x}_n + \mathbf{C}_r\mathbf{x}_r \tag{5.10}$$

where $\mathbf{A}_{rn}\mathbf{x}_n$ and $\mathbf{A}_{nr}\mathbf{x}_r$ are referred to as modeling errors that can be eliminated by using a modal system representation, $\mathbf{B}_r\mathbf{u}$ as the control spillover, and $\mathbf{C}_r\mathbf{x}_r$ as the observation spillover.

Assuming that spillovers are sufficiently small and that the quasi-steady ($\epsilon = 0$) residual state is a stable root of the residual dynamics, then a controller designed on the rigid aircraft model would be suitable for the flexible aircraft in virtue of Tikhonov's theorem (see Section 5.2.1).

In the case where spillovers are significant, the residual state vector $\mathbf{x}_r$, assumed open-loop stable as all open-loop instabilities ought to be part of the nominal state vector $\mathbf{x}_n$, could be further decomposed into states that are forced unstable by the control loop and states that remain stable [103]. Notch filters can be used to "comb out" problematic frequencies associated with the closed-loop instabilities in order to reduce and minimize their contribution to observation spillover [43]. This prevents the control law from altering the poles of the suppressed flexible modes and thus prevents the controller from destabilizing the residual dynamics.

5.2.3 Structured $\mathcal{H}_\infty$ Synthesis

The Structured $\mathcal{H}_\infty$ Synthesis is a suboptimal solution to the $\mathcal{H}_\infty$ problem using a fixed controller architecture that offers multi-model and multi-objective capabilities [47, 49]. It is

implemented under the `systune` function in MATLAB.

The framework within which the synthesis is carried out is presented in Fig. 5.1, where $\mathbf{\Delta}$ contains the uncertainty blocks, $\mathbf{K}$ contains the tunable gains, and $\mathbf{P}$ contains the plant, the controller architecture, virtual weighting filters enforcing requirements, and all other interconnections. In essence, this general model is obtained by "pulling out" all tunable gains and uncertainties, leaving the $\mathbf{P}$ system matrix behind, and grouping them into block-diagonal (structured) $\mathbf{K}$ and $\mathbf{\Delta}$ matrices. The algorithm then minimize $\mathcal{H}_\infty$ norms defined

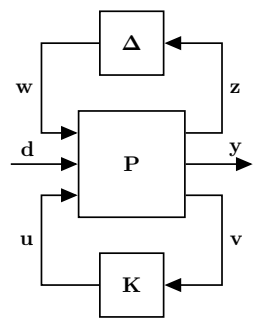

Figure 5.1 Standard framework

as

$$\|\mathbf{W}_l(s)\mathbf{T}(s)\mathbf{W}_r(s)\|_\infty = \sup_\omega \bar{\sigma}(\mathbf{W}_l(j\omega)\mathbf{T}(j\omega)\mathbf{W}_r(j\omega)) \tag{5.11}$$

where $\mathbf{W}_l(s)$ and $\mathbf{W}_r(s)$ are the virtual weighting filters, $\bar{\sigma}$ denotes the maximum singular value, and $\mathbf{T}(s)$ represent the relevant transfer function. For example, the multivariate (MIMO) transfer function from $\mathbf{d}$ to $\mathbf{y}$ is given by $\mathbf{T} = \mathbf{F}_\mathcal{U}(\mathbf{F}_\mathcal{L}(\mathbf{P}, \mathbf{K}), \mathbf{\Delta})$ where $\mathbf{F}_\mathcal{U}$ and $\mathbf{F}_\mathcal{L}$ are called the upper and lower linear fractional transformations.

This framework allows requirements to be enforced by seeking the smallest peak gain of relevant transfer functions that have been shaped by weighting filters to penalize (amplify) frequencies of interest. The norm (or cost) will exceed 1 if the transfer function exceeds the limit represented by the weighting filter, thus indicating a failure to meet the associated requirement. As such, constraints are considered "hard" tuning goals as their associated norms must be below 1, while performance-related objectives are usually "soft" goals as their associated norms need to be as small as possible. It is important to note that these norms only communicate the worst case value and that the optimization process seeks the improvement of the worst cases.

This non-smooth algorithm has been implemented in MATLAB, and can be used simultaneously for multiple norms (including $\mathcal{H}_2$ norms) which allows the enforcement of multiple

objectives on multiple models.

5.3 Model

This section describes the nonlinear model used to represent a flexible aircraft, the hardware necessary for control (such as actuators and sensors), and the flight envelope considered.

5.3.1 Flexible aircraft

The flexible aircraft model used in this paper is based on the work of [4, 8, 39, 106].

First, the relevant reference frames considered are 1) the Earth frame with origin $\mathcal{O}$, 2) the Body (fuselage) frame with origin $\mathcal{O}_f$, 3) the Left and Right Wing frames with origin $\mathcal{O}_w$, and 4) the Left, Right and Vertical Tail (empennage) frames with origin $\mathcal{O}_e$. Figure 5.2 presents the location of the origins of the various frames, along with the axes of the Right Wing and Right Tail frames as examples.

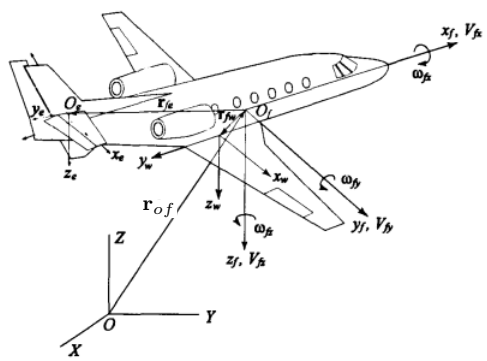

Figure 5.2 Reference frames [4]

These frames can be used to define rotations matrices between each other, with rotation angles provided in the appendix of [4]. These rotation matrices will be expressed as $\mathbf{C}_f = \mathbf{C}_f(\phi, \theta, \psi)$ for the rotation between the Earth and the fuselage, $\mathbf{C}_w$ for the rotation between the fuselage and a given wing ($\mathbf{C}_{w,r}$, $\mathbf{C}_{w,l}$), and $\mathbf{C}_e$ for the one between the fuselage and an empennage component ($\mathbf{C}_{e,r}$, $\mathbf{C}_{e,l}$, $\mathbf{C}_{e,v}$). These rotation matrices follow the roll-pitch-yaw convention.

The aircraft is subjected to the following kinematic equations linking the inertial and body rates:

$$\dot{\mathbf{r}}_{of} = \mathbf{C}_f^\top \mathbf{V}_{of}, \quad \dot{\theta} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix}^{-1} \omega_f \tag{5.12}$$

where $\mathbf{r}_{of}$ is the aircraft's position ($\mathcal{O}_f$) relative to the origin $\mathcal{O}$, $\mathbf{V}_{of}$ is the aircraft's speed (relative to the body frame $\mathcal{O}_f$), ω_f is the angular rate between frames $\mathcal{O}_f$ and $\mathcal{O}$ (expressed in body frame $\mathcal{O}_f$), and $\theta = (\phi, \theta, \psi)$ are the attitude angles.

The speed of a point on the aircraft in its local frame is

$$\mathbf{V}_f(\mathbf{r}_f, t) = \mathbf{V}_{of}(t) + \omega_f(t) \times \mathbf{r}_f \tag{5.13}$$

$$\begin{aligned}\mathbf{V}_w(\mathbf{r}_w, t) &= \mathbf{C}_w \mathbf{V}_{of}(t) + \mathbf{C}_w\Big(\omega_f(t) \times \Big(\mathbf{r}_{fw} + \mathbf{C}_w^\top(\mathbf{r}_w + \mathbf{u}_w(\mathbf{r}_w, t))\Big)\Big) \\ &\quad + \dot{\psi}_w(\mathbf{r}_w, t) \times \mathbf{r}_w + \dot{\mathbf{u}}_w(\mathbf{r}_w, t)\end{aligned} \tag{5.14}$$

$$\mathbf{V}_e(\mathbf{r}_e, t) = \mathbf{C}_e \mathbf{V}_{of}(t) + \mathbf{C}_e\Big(\omega_f(t) \times \Big(\mathbf{r}_{fe} + \mathbf{C}_e^\top \mathbf{r}_e\Big)\Big) \tag{5.15}$$

where $\mathbf{u}_w$ and ψ_w are the bending and torsional elastic displacement w.r.t. the wing frame. The Galerkin method is used to discretize these distributed parameters into flexible modes:

$$\mathbf{u}_w(\mathbf{r}_w, t) = \mathbf{\Phi}(\mathbf{r}_w)\mathbf{q}(t) \tag{5.16}$$

$$\psi_w(\mathbf{r}_w, t) = \mathbf{\Psi}(\mathbf{r}_w)\xi(t) \tag{5.17}$$

For this paper, 5 bending and 5 torsion modes are selected for each wing, with the rest of the aircraft deemed as rigid (as in [39]). The modeshapes considered are the eigenfunctions of an uniform cantilever beam for the bending modes and of a uniform clamped-free shaft for the torsion modes.

The aircraft's momentum is given by

$$\mathbf{p}(t) = \mathbf{M}(t)\mathbf{V}(t) \Leftrightarrow \begin{bmatrix}\mathbf{V}_{of} & \omega_f & \dot{\mathbf{q}} & \dot{\xi}\end{bmatrix}^\top = \mathbf{M}(\mathbf{q})^{-1}\begin{bmatrix}\mathbf{p}_{vof} & \mathbf{p}_{\omega f} & \mathbf{p}_{uw} & \mathbf{p}_{\psi w}\end{bmatrix}^\top \tag{5.18}$$

where $\mathbf{M}$ is a time-varying inertia matrix (whose expression is found in [4]) and $\mathbf{V}$ is the velocity vector. Equations provided in [4] for the inertia matrix corresponds to the integral of all relevant infinitesimal masses, while its appendix provide the distribution of mass in lumped format. As such, this integral must be discretized into lumped sums and the parallel axis theorem must be used where appropriate.

The equations of motions of the flexible aircraft can then be defined using momenta equations

$$\dot{\mathbf{p}}_{vof} = -\omega_f \times \mathbf{p}_{vof} + \mathbf{F} \tag{5.19}$$

$$\dot{\mathbf{p}}_{\omega f} = -\mathbf{V}_{of} \times \mathbf{p}_{vof} - \omega_f \times \mathbf{p}_{\omega f} + \mathbf{T} \tag{5.20}$$

$$\dot{\mathbf{p}}_{uw} = \frac{\partial \mathcal{T}}{\partial \mathbf{u}_w} - \mathcal{K}_{uw}\mathbf{q} - \mathcal{C}_{uw}\dot{\mathbf{q}} + \mathbf{Q}_{uw} \tag{5.21}$$

$$\dot{\mathbf{p}}_{\psi w} = -\mathcal{K}_{\psi w}\xi - \mathcal{C}_{\psi w}\dot{\xi} + \mathbf{Q}_{\psi w} \tag{5.22}$$

where $\partial\mathcal{T}/\partial\mathbf{u}_w = -\mathbf{\Phi}^\top(\mathbf{C}_w\omega_f) \times (\int \mathbf{V}_w \mathrm{d}m_w)$ represents the inertial forces acting on the wing (in the wing's frame), where the modal bending and torsion stiffness matrices are given by

$$\mathcal{K}_{uw} = \int EI(x)\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\mathbf{\Phi}(x)\right)^\top \frac{\mathrm{d}^2}{\mathrm{d}x^2}\mathbf{\Phi}(x)\mathrm{d}x \tag{5.23}$$

$$\mathcal{K}_{\psi w} = \int GJ(x)\left(\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{\Psi}(x)\right)^\top \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{\Psi}(x)\mathrm{d}x \tag{5.24}$$

where $\mathcal{C}_{uw}$ and $\mathcal{C}_{\psi w}$ are the modal bending and torsion damping matrices assumed to be proportional to $\mathcal{K}_{uw}$ and $\mathcal{K}_{\psi w}$, and where external forces, torques and generalized forces ($\mathbf{F}$,$\mathbf{T}$,$\mathbf{Q}_{uw}$,$\mathbf{Q}_{\psi w}$) are given by

$$\mathbf{F} = \int_{D_f} \mathbf{f}_f + \mathbf{F}_{eng}\delta(\mathbf{r} - \mathbf{r}_{eng})\mathrm{d}D_f + \mathbf{C}_w^\top \int_{D_w} \mathbf{f}_w \mathrm{d}D_w + \mathbf{C}_e^\top \int_{D_e} \mathbf{f}_e \mathrm{d}D_e \tag{5.25}$$

$$\begin{aligned}\mathbf{T} = &\int_{D_f} \mathbf{r}_f \times (\mathbf{f}_f + \mathbf{F}_{eng}\delta(\mathbf{r} - \mathbf{r}_{eng}))\mathrm{d}D_f + \int_{D_w} \left(\mathbf{r}_{fw}\mathbf{C}_w^\top + \mathbf{C}_w^\top(\mathbf{r}_w + \mathbf{\Phi}\mathbf{q})\right) \times \mathbf{f}_w \mathrm{d}D_w \\ &+ \int_{D_e} \left(\mathbf{r}_{fe}\mathbf{C}_e^\top + \mathbf{C}_e^\top \mathbf{r}_e\right) \times \mathbf{f}_e \mathrm{d}D_e\end{aligned} \tag{5.26}$$

$$\mathbf{Q}_{uw} = \int_{D_w} \mathbf{\Phi}^\top \mathbf{f}_w \mathrm{d}D_w \tag{5.27}$$

$$\mathbf{Q}_{\psi w} = \int_{D_w} \mathbf{\Phi}^\top (\mathbf{r}_w \times \mathbf{f}_w)\mathrm{d}D_w \tag{5.28}$$

with $\mathbf{f}_\bullet$ being the sum of forces per unit span expressed in the local frame of a section that include aerodynamic and gravity forces acting on the section. For the purpose of computing torques, all aerodynamic forces are applied to the line of aerodynamic centers, which is at the quarter-chord, and all gravity forces are applied at the lumped mass location as specified

in the appendix of [4]. The sum of forces per unit span for the wing is given as an example:

$$\mathbf{f}_w = \mathbf{f}_{w,a} + \mathbf{f}_{w,g} \tag{5.29}$$

$$\mathbf{f}_{w,a} = \mathbf{C}(\alpha_w, 0, 0)^\top \bar{q} c \begin{bmatrix} 0 & C_D & -C_L \end{bmatrix}^\top \tag{5.30}$$

$$\mathbf{f}_{w,g} = \mathbf{C}_w \mathbf{C}(\phi, \theta, \psi) \begin{bmatrix} 0 & 0 & \mu_w g \end{bmatrix}^\top \tag{5.31}$$

$$\alpha_w = \arctan(V_{w,z}, V_{w,y}) \tag{5.32}$$

with c being the section's chord (area per unit span) and μ_w the section's mass linear density (mass per unit span). Each section (strip) is assumed rigid chord-wise with its own $\mathbf{f}_w$ needing to be computed and applied at the relevant $\mathbf{r}_w$. The aerodynamic coefficient steady-flow values for each section is given in the appendix of [4] and are corrected using the Prandtl-Glauert factor to consider compressibility. Note that the drag coefficient is given by $C_D = C_{D0} + k_D C_L^2$, where values for the zero-lift drag C_{D0} and the quadratic drag coefficient k_D can also be found in the appendix of [4].

Note that this approach follows the strip theory, which only considers steady aerodynamics.

5.3.2 Unsteady aerodynamics - Modified strip theory

As in [39], the wing and tail aerodynamics are modeled using the modified strip theory in order to include unsteady effects. Figure 5.3 presents a rigid chord-wise section of the right wing where 1) $\mathcal{W}_r$ denotes the local reference frame of the section (whose x axis coincides with the undeformed wing span), 2) e denotes the elastic axis (which coincides with the x axis when the wing is undeformed), 3) a denotes the half-chord position, 4) h denotes the hinge position, 5) u_w denotes the vertical displacement of the section (bending), 6) ψ_w denotes the angular displacement of the section (torsion), 7) α_{rb} denotes the rigid-body contribution to the local angle of attack, and 8) δ_a denotes the aileron deflection angle. The values a and h are measured relative to the point e in the direction of the y axis and are normalized by the half-chord. As shown, a and u_w have negative values.

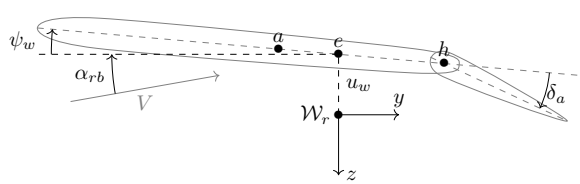

Figure 5.3 Right wing section

The section's lift and pitch moment coefficients are:

$$C_L = C_{L\alpha,sf} C(k) \alpha_{3/4} + C_{L,nc} \tag{5.33}$$

$$C_m = \left(\frac{1}{4} + \frac{a}{2}\right) C_{L\alpha,sf} C(k) \alpha_{3/4} + C_{m,nc} \tag{5.34}$$

where the first terms are the circulatory parts and $C_{L,nc}$ and $C_{m,nc}$ are the non-circulatory parts of the lift and pitch moment coefficients, $C_{L\alpha,sf}$ is the steady-flow lift curve (available in the appendix of [4]), $C(k)$ is Theordorsen's frequency-domain function [8] that will be approximated by Wagner's equivalent time-domain function and implemented as a state-space model of aerodynamic lag using Duhamel's integral [39], and $\alpha_{3/4}$ is the effective angle of attack of the strip at the 3/4 chord given by:

$$\alpha_{3/4} = \alpha_{rb} + \alpha_{u\psi} + \alpha_{\delta a} \tag{5.35}$$

$$\alpha_{rb} = \arctan\left(\bar{V}_{w,r,z}, \bar{V}_{w,r,y}\right) \tag{5.36}$$

$$\bar{\mathbf{V}}_{w,r} = \mathbf{C}_{w,r} \mathbf{V}_{of} + \mathbf{C}_{w,r} \omega_f \times \left(\mathbf{r}_{fw} + \mathbf{C}_{w,r}^\top \mathbf{r}_{w,r}\right) \tag{5.37}$$

$$\alpha_{u\psi} = \psi_w + \frac{\dot{u}_w}{V} + \frac{c}{2V}(0.5 - a)\dot{\psi}_w \tag{5.38}$$

$$\alpha_{\delta a} = \frac{T_{10}}{\pi}\delta_a + \frac{T_{11}}{2\pi}\frac{c}{2V}\dot{\delta}_a \tag{5.39}$$

where $\bar{\mathbf{V}}_{w,r}$ is the speed of the wing element without elastic terms, $V = \sqrt{\bar{V}_{w,r,y}^2 + \bar{V}_{w,r,z}^2}$ is the free-stream velocity for the section, and $\alpha_{u\psi}$ and $\alpha_{\delta a}$ denotes the contribution from elastic modes and aileron deflection to the effective local angle of attack. The constants T_{10} and T_{11} are provided in [8].

Non-circulatory lift corresponds to "added inertia" effects that are negligible for rigid aircraft, so only contributions from elastic modes are considered in this paper [106]. The non-circulatory terms are then given by:

$$C_{L,nc} = \frac{C_{L\alpha,sf}}{2\pi}\frac{c}{4V}\left(\dot{\psi}_w + \frac{\ddot{u}_w}{V} - a\frac{c}{2V}\ddot{\psi}_w\right) \tag{5.40}$$

$$C_{m,nc} = -\frac{C_{L\alpha,sf}}{2\pi}\frac{c}{8V}\left((0.5 - a)\dot{\psi}_w - a\frac{\ddot{u}_w}{V} + \frac{c}{2V}(0.125 + a^2)\ddot{\psi}_w\right) \tag{5.41}$$

As such, non-circulatory terms can be expressed as aerodynamic inertia and damping matrices for ease of implementation.

The section's drag coefficient is computed similarly as with the strip theory, but while excluding non-circulatory lift as it does not contribute to quadratic drag.

Distributed forces ($\mathbf{f}_{w,a}$) and torques can then be computed from the sectional C_L, C_D and C_m. Unlike forces calculated using strip theory, forces and torques obtained from the above lift and pitch moment coefficients are applied at the local axis ($\mathcal{W}_r$, instead of the aerodynamic center).

Equations for each of the tail's lifting surfaces can be obtain similarly, while neglecting non-circulatory terms and $\alpha_{u\psi}$ since the tail is assumed rigid.

5.3.3 Hardware models

The hardware involved in the control loop, excluding the above flexible aircraft model, are the actuators, the sensors, and the controller sampler.

Actuators The actuators are modeled as a first-order system with a 0.02 s time constant as in [39], which can be represented by

$$\dot{\mathbf{u}}(t) = \mathbf{K}(\mathbf{u}_c(t) - \mathbf{u}(t)) = 50(\mathbf{u}_c(t) - \mathbf{u}(t)) \Rightarrow \frac{\mathbf{u}(s)}{\mathbf{u}_c(s)} = \mathbf{H}_{act}(s) = \frac{50}{s+50}\mathbf{I} \tag{5.42}$$

Additionally, the ailerons are subjected to a rate limit of 100°/s and a saturation limit of ±30°, while the elevators and rudder are limited to 60°/s and ±20°. Other nonlinear effects, such as mechanical backlash, are neglected.

Sensors The sensors are idealized with a measuring time delay inferior to the controller's sampling rate. As such, the unit step delay (z^{-1}) will be used to represent all sensors.

Sampling rate The sampling rate considered is 100 Hz ($T_s = 0.01$ s), which is significantly smaller than the one used in [39] since our controller will not estimate any flexible state.

5.3.4 Flight Envelope

The flight envelope considered is presented in Fig. 5.4, where design flight conditions of Table 5.2 were highlighted. The studied plane was kept anonymous in [4], which means this flight envelope is arbitrary; it was inspired by the Cessna Citation Excel as its silhouette and trim values were similar to those provided in the reference. These flight conditions are intended to have a reasonable variation in aerodynamic behavior. As the dynamic pressures and Mach number depend on the density of the air, which depend on altitude, the standard atmospheric model [94] is used during nonlinear simulations.

Table 5.2 Flight conditions considered for design

Altitude (m)	Mach										
	0.25	0.3	0.35	0.4	0.45	0.475	0.5	0.55	0.6	0.65	0.7
12000							# 19		# 20		# 21
10000							# 16		# 17		# 18
8000					# 13			# 14		# 15	
6000				# 10		# 11		# 12			
4000			# 7	# 8	# 9						
2000		# 4	# 5	# 6							
1000	# 1	# 2	# 3								

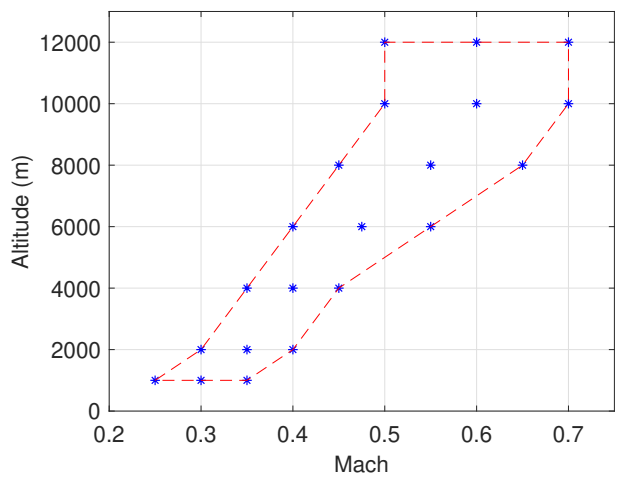

Figure 5.4 Flight envelope

5.4 Control

The INDI control architecture used in this paper is presented in Fig. 5.5 and was based on the one used in [74]. It includes a Feedback-Linearization (FL) stage (see Section 5.2.1), a Linear-Controller (LC) stage, and a Reference Model (RM). The LC stage features a Proportional-Integral-Derivative (PID) controller (with tunable gains $\mathbf{K}_p$, $\mathbf{K}_i$, and $\mathbf{K}_d$) presented in Section 5.4.2, while the RM is a first-order transfer function with unit steady-state and tunable gain K_{rm}. The addition of a RM is meant to shape the input command as to prevent aggressive maneuvers that the controller will fail to track due to actuation rate limiters. While excluded in this paper, this architecture can support the addition of Pseudo-Control Hedging (PCH) to address actuator saturation concerns [57]. Note that the Zero-Order Hold (at the input of $\mathbf{H}_{act}(s)$, the actuators) and the samplers (at the input of $\mathbf{F}(z) = z^{-1}\mathbf{I}$, the synchronized delay) are not shown in this diagram, but are nonetheless required for imple-

mentation.

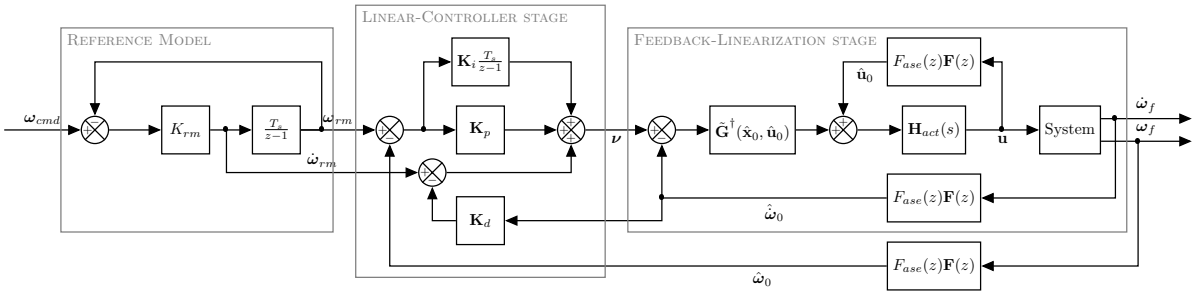

Figure 5.5 INDI Control Law

During the controller design, the flexible aircraft will be modeled as a rigid aircraft perturbed by unmodeled flexible modes through spillover, which requires ASE filtering (see Section 5.2.2). To ensure measurement delay synchronization, the ASE filter $F_{ase}(z)$ is applied on all measurements for all axes. Furthermore, unsteady aerodynamics, such as aerodynamic lags, are neglected for the rigid model.

5.4.1 Feedback-Linearization stage - Angular acceleration loop

While the inner stage of INDI control is intended to provide an approximate pure integrator system, the hardware considered (see Section 5.3.3) may not fulfill the idealized assumptions stated in Section 5.2.1. As such, an alternative expression using frozen-time velocity-based linearization, as described in [73], is considered in order to obtain a more accurate closed-loop system representation for the FL stage:

$$\ddot{\mathbf{x}}(t) = \mathbf{A}_0\dot{\mathbf{x}}(t) + \mathbf{G}_0\dot{\mathbf{u}}(t) + \epsilon_{vl}(t) \Rightarrow \dot{\mathbf{x}}(t) = e^{\mathbf{A}_0(t-t_0)}\dot{\mathbf{x}}_0 + \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\mathbf{G}_0\dot{\mathbf{u}}(\tau)d\tau + \epsilon_l(t) \quad (5.43)$$

where $\mathbf{x} = [\mathbf{V}_{of}^{\top}, \omega_f^{\top}, \mathbf{\Theta}^{\top}, \mathbf{r}_{of}]^{\top}$ is the (nominal) state vector, $\epsilon_{vl}(t) = (\mathbf{A}(t) - \mathbf{A}_0)\dot{\mathbf{x}}(t) + (\mathbf{G}(t) - \mathbf{G}_0)\dot{\mathbf{u}}(t)$ is the velocity-based linearization error, $\epsilon_l(t) = \int_{t_0}^{t} e^{\mathbf{A}_0(t-\tau)}\epsilon_{vl}(\tau)\mathrm{d}\tau$ is the linearization error, and $\dot{\mathbf{u}}(t)$ is given by Eq.(5.42).

Considering the INDI control law:

$$\mathbf{u}_c(t) = \mathbf{u}_0 + \tilde{\mathbf{G}}_0^{\dagger}(\nu(t) - \mathbf{H}_{sm}\dot{\mathbf{x}}_0) \quad (5.44)$$

the continuous approximation of the close-loop model in the Laplace domain is then:

$$\dot{\omega}_f(s) = \mathbf{H}_{sm}\dot{\mathbf{x}}_0 = \mathbf{H}_{sm}\Big(s\mathbf{I} - \mathbf{A}_0 + \mathbf{G}_0\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{F}(s)\mathbf{H}_{sm}\Big)^{-1}\Big(\mathbf{G}_0\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^{\dagger}\nu(s) + \epsilon_l(s)\Big) \tag{5.45}$$

$$= \Big(\mathbf{I} + \mathbf{P}_0(s)\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^{\dagger}\mathbf{F}(s)\mathbf{H}_{sm}\Big)^{-1}\Big(\mathbf{P}_0(s)\tilde{\mathbf{K}}(s)\tilde{\mathbf{G}}_0^{\dagger}\nu(s) + \mathbf{H}_{sm}\epsilon_l(s)\Big) \tag{5.46}$$

$$= \mathbf{H}_{sm}(\mathbf{I} + \mathbf{\Delta}_a(s))\Big\{\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^{\dagger}\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^{\dagger}\Big\}(\mathbf{I} + \mathbf{\Delta}_d(s))\nu(s) + \epsilon_t(s) \tag{5.47}$$

where $\tilde{\mathbf{K}}(s) = \Big(\mathbf{I} + {}^1\!/\!_s(\mathbf{I} - \hat{\mathbf{F}}(s))\mathbf{K}\Big)^{-1}\mathbf{K} = \Big(\mathbf{I} - \hat{\mathbf{F}}(s)\mathbf{H}_{act}(s)\Big)^{-1}s\mathbf{H}_{act}(s) \approx (\mathbf{I} + T_s\mathbf{K})^{-1}\mathbf{K}$ is the effective bandwidth, $\mathbf{P}_0(s) = \mathbf{H}_{sm}(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0$ is the plant, $\mathbf{F}(s) = e^{-sT_s}\mathbf{I}$ is the delay (with $\hat{\mathbf{F}}(s) = \mathbf{F}(s)$ when delays are synchronized), $\mathbf{\Delta}_a$ and $\mathbf{\Delta}_d$ are the model multiplicative uncertainties due to plant dynamics and delay compensation that are applied on the idealized INDI closed-loop model, and $\epsilon_t(s)$ is an additive tracking error due to the linearization error.

In order for this control law to achieve approximate feedback linearization, the INDI applicability conditions [73] must be met. Since the system is considered as a perturbed rigid aircraft during the controller design, the rigid model is used for the applicability tests, with the assumption that the perturbations will not destabilize the aircraft. An example of a detailed applicability assessment for rigid aircraft is available in [74], where mitigation techniques suggested in [73] had to be used due to less-than-ideal hardware. In this paper, however, the choice of actuators, delays, and sampling rate ensures that most applicability tests are met without issue, while the robust tuning of the LC stage will mitigate the impact of $\mathbf{\Delta}_a$ and $\mathbf{\Delta}_d$ multiplicative uncertainties on performance [74].

During controller design, flexible modes are considered to be singular perturbations that will not destabilize the aircraft as long as their singular parameter is smaller than the SPM of the nominal system (rigid model). As such, only the smallest flexible modes, which have the highest singular parameter values, need to be suppressed ("combed out") by the notch filters when they cause the system to be closed-loop unstable.

In order to assess how many notch filters may be required, the frequency response of the unfiltered closed-loop of the FL stage is required. While the full plant dynamics are necessarily included to capture the flexible modes' spillover, the control law is simplified as continuous and delay-free ($\mathbf{\Delta}_a \neq \mathbf{0}$ and $\mathbf{F}(s) = \mathbf{I} \Rightarrow \mathbf{\Delta}_d = \mathbf{0}$). Figure 5.6 presents the frequency responses of the unfiltered closed-loop FL stage for all design cases where 2 peaks, associated with flexible modes, can be seen crossing the 0 dB line (highlighted in red), which motivates the use of 2 notch filters in the considered ASE filtering. As such, the ASE filter is given by

$$F_{ase}(s) = \frac{(s^2 + a_1(2\zeta_1\omega_1)s + \omega_1^2)(s^2 + a_2(2\zeta_2\omega_2)s + \omega_2^2)}{(s^2 + (2\zeta_1\omega_1)s + \omega_1^2)(s^2 + (2\zeta_2\omega_2)s + \omega_2^2)} \tag{5.48}$$

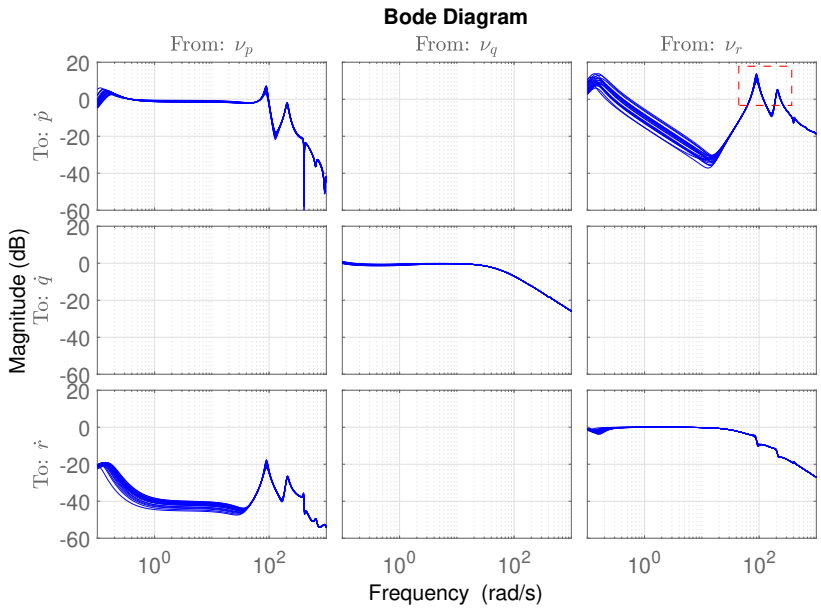

Figure 5.6 Frequency responses of the unfiltered closed-loop FL stage (flexible aircraft linear models)

where a_1, ζ_1, ω_1 and a_2, ζ_2, ω_2 are the tuning parameters of the first and second notch filters. While this expression is in the continuous domain, an equivalent discrete filter will be implemented on the nonlinear model.

Unfortunately, adding a filter to a measurement signal will require the same filter to be added on all other measurement signals to ensure synchronization of delays. This will result in a greater loss of performance (and greater, now frequency-dependent, synchronized delay) than for traditional INDI control applications.

Furthermore, the presence of multiplicative uncertainties warrants the use of a robust tuning technique to ensure robustness, while the additive tracking error ϵ_t should be rejected by INDI's inherent robustness to additive perturbations. This robust tuning will include the added ASE filters to compensate its adverse effect on performance and stability margins.

5.4.2 Linear-controller stage - Angular rate loop

The linear controller selected (see Fig. 5.5) is a PID controller using 3x3 full-matrix tunable gains instead of using 3 distributed univariate (SISO) controllers [74]. This change is to ensure better decoupling of the closed-loop as the FL stage may struggle to enforce it.

A RM is included to help limit the commanded actuator rate and to provide the reference derivative signal without using a pseudo-derivation filter. It is implemented as a first-order

discrete transfer function corresponding to $K_{rm}/(s+K_{rm})$. The RM gain K_{rm} will be selected to match the requirement for settling time.

The contribution from $\dot{\omega}_{rm}$ is equivalent to a feedforward term, which could be combined with the FL-stage's angular acceleration feedback to highlight an error derivative. As such, the $\mathbf{K}_d$ gain is used for regulation and is not applied on the signal $\dot{\omega}_{rm}$.

5.4.3 Robust tuning - Structured $\mathcal{H}_\infty$ Synthesis

The robust tuning used here follows the same approach as [74], with the addition of the ASE filter parameter tuning. While both can be tuned in a single synthesis, these 2 tuning problems are considered separately for computer efficiency concerns given that the ASE filter tuning will require linear flexible models while the controller tuning uses rigid models.

The balanced disk margin (see Fig. 5.7) is used to enforce gain and phase margins within the $\mathcal{H}_\infty$ framework. The minimum disk radius imposed is determined as $\alpha_{dm} = 2\max((GM-1)/(GM+1), \tan(PM/2))$ with "GM" and "PM" denoting the desired Gain and Phase Margins (expressed in absolute units and radians).

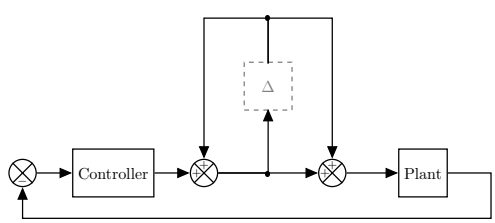

Figure 5.7 Balanced disk margin assessment diagram

5.4.3.1 ASE filter tuning

The requirements considered for the ASE filter parameter tuning process are presented in Table 5.3, where $\alpha_{dm,ase}$ is the minimal disk radius that corresponds to ±4 dB and ±30°. The uncertainty block for the disk margin assessment (see Fig. 5.7) is injected at the plant (aircraft) input, which is considered to be the actuator position for the FL stage. All weighting filters are applied to their relevant output signals going into the associated uncertainty blocks.

The synthesis is carried out in a sequential manner, starting with a single case at flight condition # 11 and expanding progressively to 9 and then 21 cases. The full flexible linear model is used during this tuning process, which was obtained using the `linmod` function in

Table 5.3 Requirements - ASE filter tuning

Requirement	Type	Description	Weighting filter	Frequency range (rad/s)
Minimize observation spillover	Soft	FL stage's input-to-output gain should not exceed 1 and should be at most −8 dB outside the actuator bandwidth (as per [3])	$\frac{s+50}{0.3981s+50}$	[10,1000]
Maintain tracking performance (angular acceleration)	Soft	FL tracking error should not exceed 50%	$\frac{1}{0.5}$	[1,10]
Maintain sufficient bandwidth	Hard	Filter must not attenuate signals more than 10% within the controlled bandwidth	$\frac{1}{0.1}$	[0.001,35]
Maintain sufficient stability margins	Hard	Gain and phase margins of the FL stage must be 4 dB and 30° at worst	$\alpha_{dm,ase} = 0.5359$	[1,1000]

MATLAB, while neglecting delays. This matches the continuous and delay-free simplified control law used in Section 5.4.1, reduces the computational effort involved in the synthesis using linear flexible models, and prevents the tuning algorithm from trying to compensate the delay-induced loss of phase in an attempt to fulfill the stability margin requirement. The algorithm successfully converged to a solution with costs shown in Table 5.4, yielding the following tuned ASE filter:

$$F_{ase}(s) = \frac{(s^2 + 0.0760(2 * 0.8146 * 80.8725)s + 80.8725^2)}{(s^2 + (2 * 0.8146 * 80.8725)s + 80.8725^2)} \cdot \frac{(s^2 + 0.3602(2 * 0.2865 * 223.5552)s + 223.5552^2)}{(s^2 + (2 * 0.2865 * 223.5552)s + 223.5552^2)} \tag{5.49}$$

The time and frequency responses of the filtered FL stage are presented in Fig. 5.8, where it can be observed that the problematic peaks of Fig. 5.6 have been effectively suppressed and that no significant observation spillover remains. Furthermore, the time responses are close to the behavior of the rigid model's FL stage and no longer diverges due to flexible modes being forced unstable.

Table 5.4 Synthesis results - ASE filter tuning

Goal	Type	Cost
Spillover	Soft	0.9851
Tracking	Soft	0.8339
Stability margin	Hard	0.8668
Bandwidth	Hard	0.9999

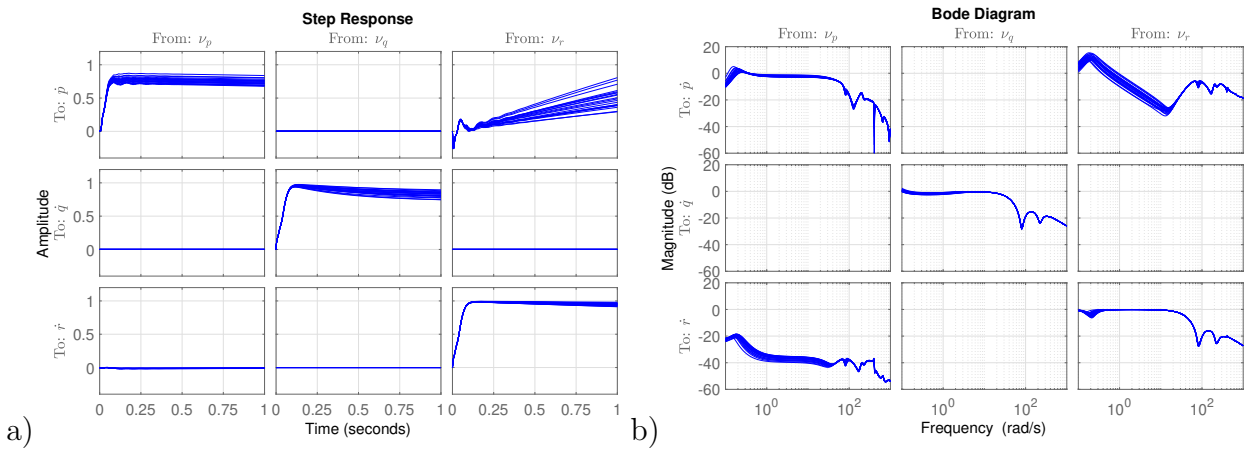

Figure 5.8 Filtered closed-loop FL stage (flexible aircraft linear models):

a) Time responses, and b) frequency responses

5.4.3.2 Controller gains tuning

The requirements considered for the LC gain tuning process are presented in Table 5.5, where α_{dm} is the minimal disk radius that corresponds to ± 6 dB and $\pm 45°$. The uncertainty block for the disk margin assessment (see Fig. 5.7) is injected at the virtual plant (closed-loop FL stage) input, which is considered to be the pseudo-command for the LC stage. The relevant frequency band used during synthesis is $[0.001, 1000]$ rad/s for all tuning objectives. Weighting filters are applied to their relevant output signals going into the associated uncertainty blocks. While the tracking weighting filter is shown as a matrix and implemented as such in the virtual models, their corresponding input/output pairs are treated separately as 9 individual objectives. The RM gain is selected as $K_{rm} = {}^{-\ln(5\%)}/_{5} \approx 0.6$ to match the 5 s settling time (within 5%) requirement.

The synthesis is carried in the same sequential manner as with the ASE filter tuning. Rigid

Table 5.5 Requirements - LC gains tuning

Requirement	Type	Description	Weighting filter
Maximize tracking performance (angular rate) and minimize coupling	Soft	Settling time (5%) should be 5 s for all axes, Tracking error should not exceed 10%, Pitch coupling should not exceed 1%, Roll-to-yaw coupling should not exceed 5%, Yaw-to-roll coupling should not exceed 20%.	$\begin{bmatrix} \frac{s+2/5}{1.1s+0.05*2/5} & \frac{1}{0.01} & \frac{1}{0.2} \\ \frac{1}{0.01} & \frac{s+2/5}{1.1s+0.05*2/5} & \frac{1}{0.01} \\ \frac{1}{0.05} & \frac{1}{0.01} & \frac{s+2/5}{1.1s+0.05*2/5} \end{bmatrix}$
Minimize control effort	Soft	Pseudo-command input should not amplify the commanded angular rate by more than 10.	$\frac{1}{10}$
Maintain sufficient stability margins	Hard	Gain and phase margins of the LC stage must be 6 dB and 45° at worst.	$\alpha_{dm} = 0.8284$

linear models with ASE filters are used during this tuning process, which was obtained using the `linmod` function in MATLAB with the flexible modes and unsteady aerodynamics disabled and with time delays represented by a Padé approximant of order 8. The algorithm successfully converged to a solution with costs shown in Table 5.6, yielding the following tuned gains:

$$\mathbf{K}_p = \begin{bmatrix} 3.3217 & 0.1101 & 0.0000 \\ 0.0000 & 3.1319 & 0.2116 \\ 1.6979 & 0.0319 & 4.5551 \end{bmatrix}, \quad \mathbf{K}_i = \begin{bmatrix} 0.2354 & 0.2222 & 0.0000 \\ 0.0445 & 0.3855 & 0.0352 \\ 0.0000 & 0.1370 & 0.0000 \end{bmatrix},$$

$$\mathbf{K}_d = \begin{bmatrix} 0.3050 & 0.0009 & 0.0546 \\ 0.0007 & 0.0939 & 0.0036 \\ 0.0587 & 0.0033 & 0.1518 \end{bmatrix}$$

Figure 5.9 presents the time and frequency responses of the flexible aircraft linear models at the considered design flight conditions. Most notable is the significant yaw-to-roll coupling which the controller cannot reduce any further while maintaining sufficient stability margins, and its impact on the yaw response which is slower to settle than the other axes. While the steady-state error of the roll axis has not been completely eliminated, the roll rate tracking remains satisfactory and the overall closed-loop behavior should be sufficiently uniform and performant to ensure that well-tuned outer loops will be able to fulfill their maneuver-based requirements.

5.5 Verification

This section will seek to verify the results obtained from linear models. First, the time responses of the nonlinear flexible aircraft model are evaluated at the design flight conditions to confirm that they match the linear responses obtained. Then, Monte Carlo simulation campaigns will be carried out over the entire flight envelope to ensure that the controller fulfills the tracking and stability margin requirements for flight conditions other than those selected for design.

Table 5.6 Synthesis results - LC Gains

Goal	Type	Cost
Tracking (and Decoupling)	Soft	$\begin{bmatrix} 1.7081 & 0.9834 & 1.7081 \\ 1.0263 & 0.9185 & 1.7081 \\ 1.2711 & 1.7081 & 1.0595 \end{bmatrix}$
Control effort	Soft	0.1300
Stability margin	Hard	1.0000

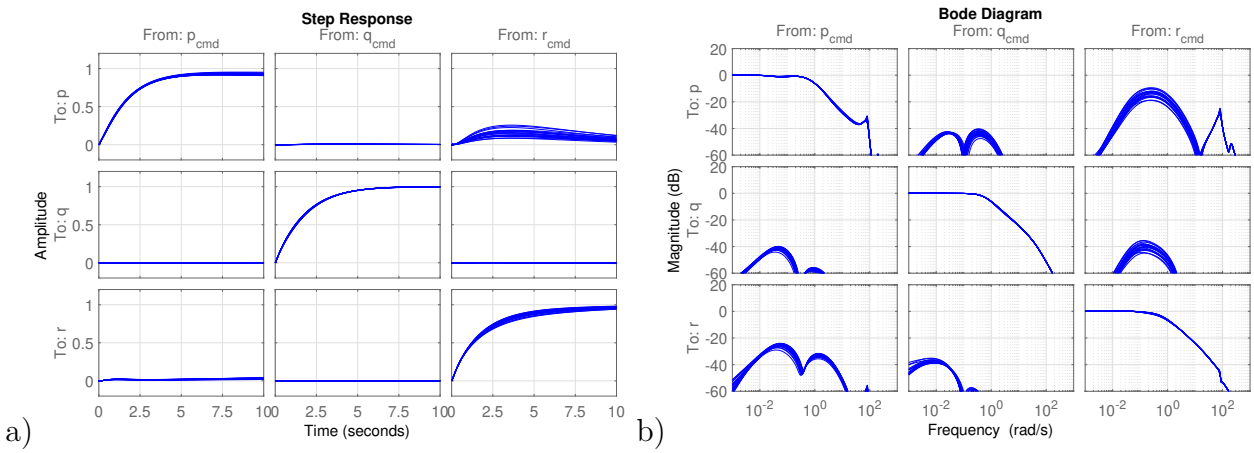

Figure 5.9 Flexible aircraft linear responses at design flight conditions:

a) Time responses, and b) frequency responses

5.5.1 Nonlinear responses

Figure 5.10 presents the time response of the nonlinear flexible aircraft model at design flight conditions for a step command of 1°/s. Comparing with the linear responses shown in Fig. 5.9, the closed-loop linear models do match their nonlinear counter-parts at the design points for small commands. This confirms that, for the specific flight conditions used during design, the problematic flexible modes have been suppressed by the ASE filters as intended and that the controller designed on the rigid aircraft successfully stabilize the flexible aircraft as expected.

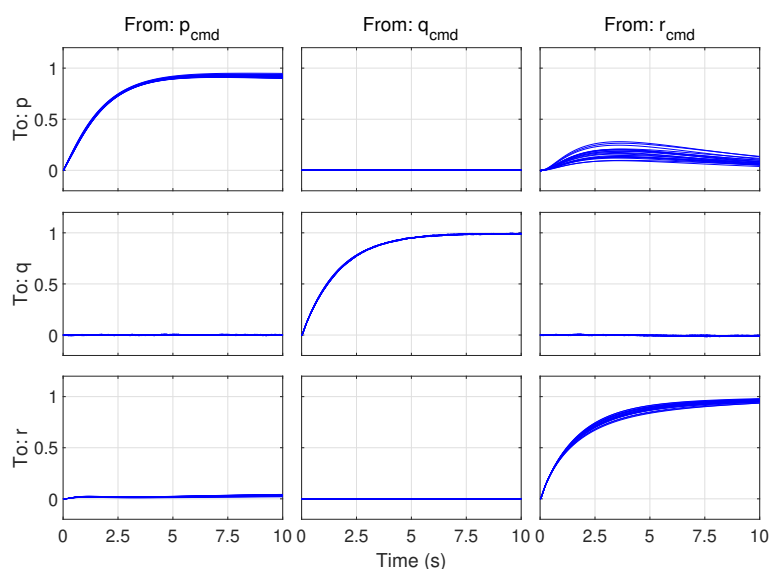

Figure 5.10 Flexible aircraft nonlinear time responses at design flight conditions

5.5.2 Monte Carlo simulation campaigns

As was done in [74], a separate Monte Carlo simulation campaign is considered for each requirement being evaluated on the nonlinear model, which are the tracking (and decoupling) and stability margins requirements (see Table 5.5). Each campaign will use a randomized flight condition and inject a swept sine chirp signal [98] at the relevant input (see Fig. 5.7 for the Stability Margin requirement). The simulation output is processed using Fast-Fourier Transforms in order to retrieve a frequency response, which is then evaluated using the same weighting filters (and $\mathcal{H}_\infty$ norms) as during controller tuning. Due to practical and implementation constraints, the frequency range evaluated for the nonlinear responses are reduced to $[0.1, 50]$ Hz, the pitch coupling requirement was relaxed to 5% error, and the number of randomized trials used is 200 for both campaigns.

To generate randomized flight condition with uniform probability, an altitude variable with uniform probability over the interval $[1000, 12000]$ and normalized speed variable with uniform probability over $[0, 1]$ are used. The randomized flight condition's Mach number is then obtained through linear interpolation between the minimum and maximum Mach number at the given randomized altitude. The minimum and maximum Mach number are obtained through linear interpolations from the flight conditions presented in Table 5.2, which corresponds to the limits visible in Fig. 5.4.

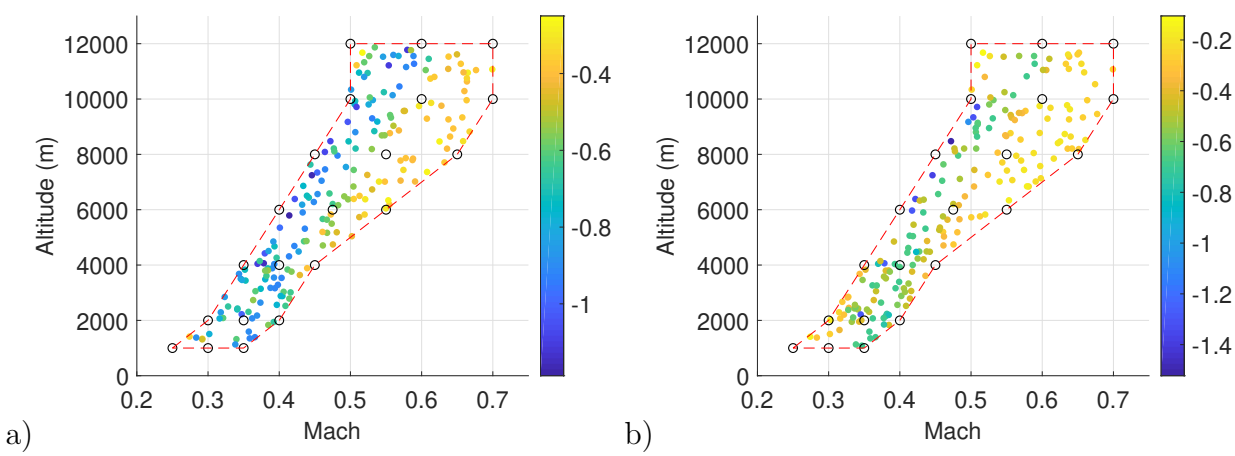

Figure 5.11 Monte Carlo campaigns result: a) Tracking, and b) Stability Margins

Figure 5.11 presents the results of the Monte Carlo campaigns for both requirements. The tracking performance appears better than the linear ones, which can be credited to the relaxed pitch coupling requirement and the reduced frequency band of interest. On the other hand, stability margins appear to have deteriorated slightly due to nonlinear and neglected dynamics effects, such as aerodynamic lags (see Section 5.3.2), for cases near the edge of the

flight envelope (at low speeds and altitudes in the upper troposphere). Furthermore, both stability margins and tracking performance improve with speed, which would indicate that the impact of unsteady effects on the closed-loop system is greater at lower speeds. As flexible modes would be more sensitive to being perturbed by the controller at higher speeds, these results confirm that the ASE filters have successfully suppressed them as intended throughout the flight envelope.

Note that this application example aimed to showcase the capabilities of INDI control for flexible aircraft using a common notch filtering technique. That objective has been fulfilled successfully as the achieved nonlinear performance is deemed sufficient to track the outer loop commands in order to accomplish standard maneuvers.

5.6 Conclusion

This paper demonstrates the feasibility of INDI control for flexible aircraft using notch filters and extends the work presented in [74] for rigid aircraft. The flexible aircraft nonlinear model used is based on [4, 8, 39, 106], which includes numerical values and thus allow for comparisons. A classic INDI feedback-linearization stage with identical ASE filters on all axes was used along with a full-matrix gain PID controller and a reference model for implementation concerns. The Structured $\mathcal{H}_\infty$ Synthesis was selected to tune both the ASE filters and the linear controller gains to ensure robustness to model variations over the flight envelope, as the feedback linearization is only approximately achieved given the considered hardware [73]. Nonlinear performance was assessed through Monte Carlo simulation campaigns, which use swept sine chirps and Fast-Fourier Transforms to generate frequency responses that can be evaluated using the same $\mathcal{H}_\infty$ norms as those used during the controller synthesis.

Future work includes the consideration of higher-order actuator dynamics to allow the inclusion of hinge moment dynamics as in [106], which would increase model fidelity and provide an additional demonstration of the realistic actuator mitigation proposed in [73,74]. Further advances in singular perturbation margins could provide potential innovation for more efficient and systematic ASE filter design. Synthesis of the outer loops, responsible for maneuvers, would also provide interesting insights into the achievable performance and the potential for eventual certification.

CHAPTER 6 GENERAL DISCUSSION

This chapter provides a general discussion of the work presented in the previous three chapters, exploring their interrelationships, how they address the research objectives outlined in Chapter 2, and how they collectively contribute to fulfilling the main research objective of this thesis.

6.1 Summary

In Chapter 3, an alternative model for INDI was proposed, utilizing a frozen-time velocity-based linearization instead of the traditional truncation of a Taylor series expansion. This alternative model offers a smaller linearization error that is easier to estimate, resulting in improved accuracy in describing the system dynamics. This led to further investigations into the causes of undesirable phenomena, such as loss of tracking performance and intersample rippling. Notably, it was observed that neglected plant dynamics form a multiplicative error that is not rejected by the INDI scheme alone. Additionally, the sampling rate and actuator bandwidth combine to form an "effective bandwidth" that is limited by both, and this limitation cannot be overcome by arbitrarily increasing the sampling rate beyond the actuator bandwidth. Furthermore, the sensitivity to signal synchronization was understood as a consequence of the well-known Smith Predictor, which was identified as the underlying scheme for delay compensation. The investigation also demonstrated how performance deteriorates on a rigid aircraft as various hardware parameters worsen, providing valuable insights into the applicability of INDI in limit cases where the system's hardware may underperform relative to the simplifying assumptions, such as when realistic actuators, which are non-identical and non-minimal phase, are used.

These findings led to the refinement of INDI's qualitative and interdependent simplifying assumptions into quantitative applicability tests. These tests, coupled with proposed mitigation strategies in case of failure, will facilitate practical implementation in industrial settings, where there is often limited flexibility to make significant hardware changes at the late development stages, when control laws are typically designed. As such, these tests can serve as assessment tools to determine whether this control approach is suitable for a given application, considering the existing hardware. Additionally, they help relax certain assumptions, such as the requirement for identical first-order actuators. These contributions ultimately enhance the Technological Readiness Level of INDI control.

Chapter 4 presented a demonstration of the proposed INDI applicability tests, alongside the use of Structured $\mathcal{H}_\infty$ Synthesis with INDI on a rigid aircraft with non-ideal hardware. This demonstration spanned the full flight envelope of a rigid aircraft, both with and without the modified INDI scheme proposed in Chapter 3 to mitigate the effects of non-identical actuators with non-minimal phase. The effectiveness of the proposed mitigation technique was demonstrated, as well as the ability of Structured $\mathcal{H}\infty$ Synthesis to robustly tune both the modified and unmodified versions of INDI, in addition to its well-established capability to efficiently tune gain-scheduled controllers. Furthermore, it was suggested that the Worst-Case Analysis Toolbox could serve as a valuable tool to enhance verification beyond traditional Monte Carlo simulations. This toolbox allows for the reuse of $\mathcal{H}_\infty$ norms defined during synthesis as a performance metric and can identify the most problematic (worst-case) scenario, which can then be targeted for improvement before launching a more comprehensive verification campaign.

While this demonstration was not exhaustive, it still showed that INDI can be applied to a rigid aircraft using realistic hardware and that Structured $\mathcal{H}_\infty$ Synthesis can be used to tune the controller more effectively. We note that additional requirements and uncertainties can be added to the synthesis, making it a powerful tool.

Chapter 5 addressed the control of flexible aircraft directly. The integrated modeling approach (see Chapter 2) was employed for the flexible aircraft, providing a foundation that allows its modular components to be updated or replaced with more accurate versions in the future (e.g., finite-element models). It was demonstrated that the use of ASE notch filters, a widely used industry technique, effectively mitigates the impact of neglected flexible modes, thereby enabling the use of controllers designed for a rigid version of the aircraft. This approach eliminates the need for novel sensors when applying INDI to flexible aircraft, as previously suggested in the literature [39]. As in Chapter 4, the Structured $\mathcal{H}_\infty$ Synthesis was used to tune the linear controller, making it robust to linearization errors and capturing the variability in the aircraft's aerodynamics. This approach also mitigates the performance loss caused by the inclusion of ASE filters.

Thus, it is demonstrated that the control design process outlined in Chapter 4 can be extended to flexible aircraft, without the need for additional sensors, since the standard mitigation technique of notch filtering can be employed.

6.2 Synthesis

Taken together, the contributions from Chapters 3, 4, and 5 provide the elements to support the proposal of a new design process for flexible aircraft control.

The current commonplace practice, the so-called "traditional" process, consist of designing the plane itself, modeling it, generating a collection of linear models at selected "design" flight conditions, tuning several PIDs (for each axis, for both the inner and outer loops) for each and every linear model, fine-tuning the gains to ensure a smooth scheduling function, and then carrying out a Monte-Carlo nonlinear simulation campaign at randomized flight conditions. This process proves tedious when the underlying nonlinear model is updated frequently, as all subsequent steps need to be re-done.

The proposed alternative process would inevitably start off the same with the aircraft design, but already differs when it comes to modeling as an integrated approach is suggested as a preliminary flexible aircraft model that can be improved at a later time, such as when new aerodynamics data become available thanks to flight testing. A collection of linear models is still required, with the added step of computing the control effectiveness matrix ($\mathbf{G}_0$), with the help of parameter sweeps over the angles of attack and sideslip, to populate a look-up table. At this stage of development, the applicability tests are performed and relevant mitigation components are added, such as ASE filters. Then, the Structured $\mathcal{H}_\infty$ Synthesis is used to tune the single unique controller in order to fulfill the requirements for all design cases at once, including the ASE filters if any are present. Finally, the tuned controller is verified via a Worst-Case analysis before launching the Monte-Carlo simulation campaign, as to identify & fix potential issues without needing to conduct the Monte-Carlo campaign more than once. Furthermore, the outer loop can be designed in parallel during this time; the inner loop can be replaced by the reference model it is meant to track to allow a preliminary design of the outer loop.

The benefits of the proposed solution over the common practice are as follows.

- Automated tuning, eliminating the need for tedious manual re-tuning;
- Replaces gain-scheduling of the controller with a deterministic look-up table derived from a more accurate control effectiveness matrix;
- Nonlinear assessment using worst-case figures of merit ($\mathcal{H}\infty$ norms), reducing the need for lengthy simulation campaigns and providing additional options for post-processing analysis;

- Robust tuning that corrects minor performance issues from the Feedback-Linearization stage;
- Decouples the design of the inner loop and outer loop controllers, enabling the use of fixed-gain controllers in the outer loop that can be tuned based on the reference model the inner loop is meant to track (which could serve as the starting point for robust tuning using the closed inner loop linear models);
- Future-proofing, as better hardware will directly improve the performance of the Feedback-Linearization stage, and the outer loop controllers can be reused when designing a new aircraft;
- Reduces development costs for new aerospace platforms (including novel configurations like blended wings) by ensuring that both INDI and Structured $\mathcal{H}_\infty$ Synthesis are "generic" (platform agnostic), with their control architecture independent of the plant, thanks to the Control Effectiveness Matrix and added robustness to plant uncertainties.

In addition, the proposed control design process consists of modular improvements upon the common practice, allowing for a gradual transition as industries become familiar with robust tuning first, followed by the modified INDI control design, and eventually addressing more advanced issues of flexible aircraft control. Figure 6.1 shows diagrams representing the intermediate processes, each of which forms a valid control design process that improves upon the previous one. These can be described as follows:

1. *Common practice*: This is the starting point, the current status quo.
2. *Adoption of robust tuning methods*: This allows for familiarization with the $\mathcal{H}_\infty$ framework while retaining the familiar control architecture.
3. *Inclusion of robust analysis methods*: This further familiarizes the team with robust theory and enables quicker iterations of tuning before carrying out the Monte Carlo campaign.
4. *Adoption of INDI*: This is a significant step that overhauls the attitude control loop architecture and should only be undertaken once the control design team is comfortable with the intended robust tuning methods. Furthermore, it is recommended to implement INDI on a familiar aircraft (and model) before addressing novel aircraft designs or modeling approaches.

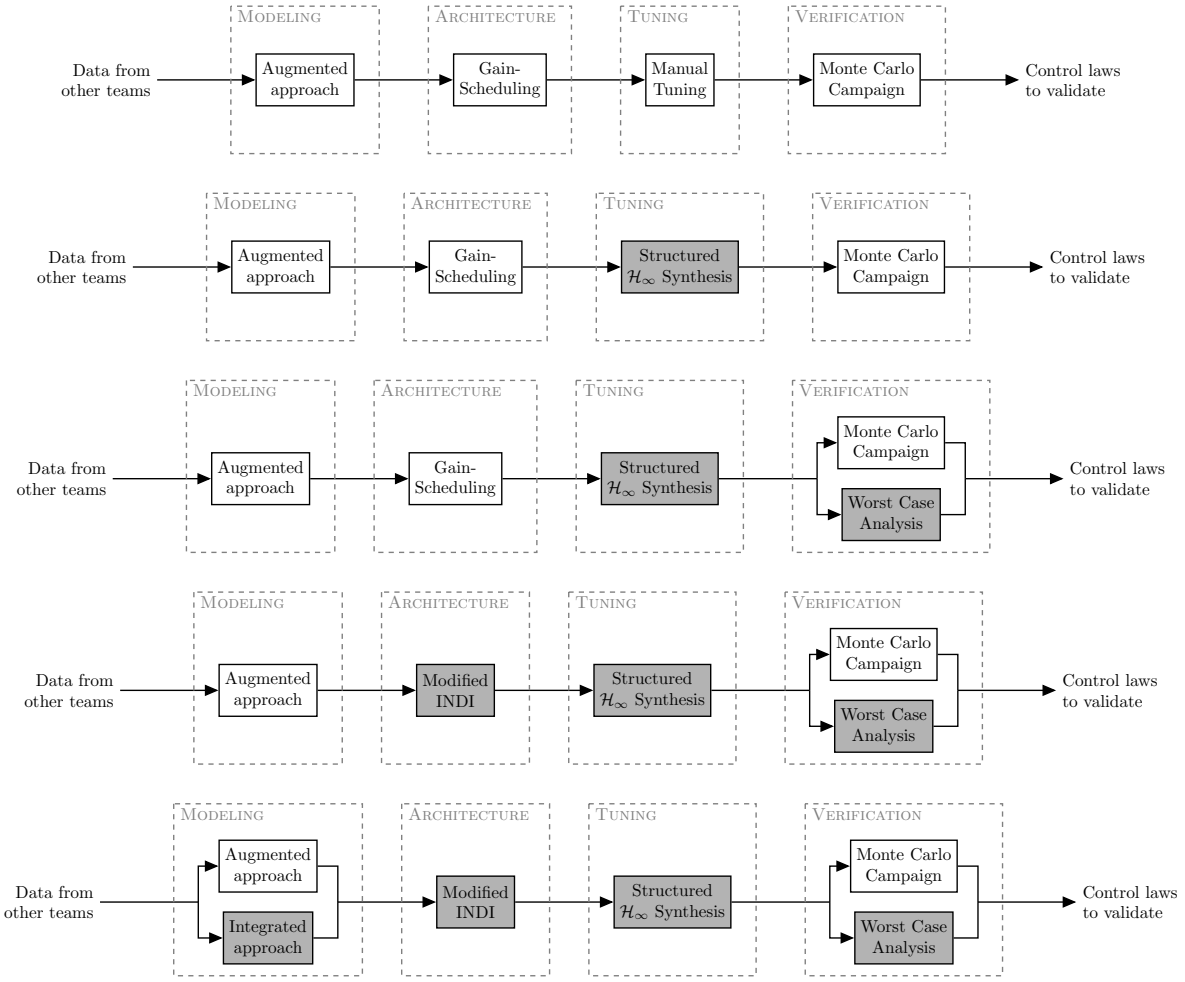

Figure 6.1 Diagrams of the gradual transition from the common practice to the proposed solution to control design process

5. *Inclusion of integrated modeling*: This final step is particularly useful when data is scarce, such as during the early stages of developing a novel aircraft design. Practitioners should first familiarize themselves with all the other suggested modules.

Taken together, these steps form a coherent and reliable transition towards the recommended solution, which will facilitate its adoption by the industry. Each step provides an opportunity to master a new tool and address specific practical issues, establishing a solid foundation for the next phase. To further assist with this transition, quick-reference guides for practitioners are included at the end of this thesis. A user guide for the tuning goals of Structured $\mathcal{H}_\infty$ Synthesis is provided in Appendix A (originally published in the author's Master's thesis [109], and translated for the reader's convenience). Additionally, Appendix B includes the INDI applicability tests and associated mitigation strategies.

Given these advantages, the proposed design process offers a promising alternative to current practices, expected to reduce development costs for the next generation of aircraft, thus fulfilling the main research objective outlined in Chapter 2.

CHAPTER 7 CONCLUSION

To conclude this thesis, this chapter provides a summary of the works presented, their limitations, and the potential future research they enable.

7.1 Summary of Works

Chapter 3 relaxed the conditions of applicability for INDI by providing quantitative tests and proposing modifications to the architecture to mitigate any failed tests.

Chapter 4 demonstrated the use of several suggested approaches on a rigid aircraft, such as applicability tests, mitigating modifications to the architecture, tuning through the Structured $\mathcal{H}_\infty$ Synthesis, and verification through worst-case analysis.

Chapter 5 demonstrated the use of INDI for flexible aircraft with the help of ASE filters, which mitigate the problematic singular perturbations caused by low-frequency flexible modes.

Chapter 6 explained how the works contained in this thesis address the stated research objectives and, when taken collectively, form a proposed design process that can be adopted modularly and progressively by the industry to improve their development productivity.

In sum, the Technology Readiness Level of INDI has been improved, and the main research objective has been fulfilled. It is hoped that this work inspires reforms in the industry to accelerate the current technological transfer efforts from partnerships with academia.

7.2 Limitations

While the demonstrations presented in this thesis are sufficient to support the stated research objectives, they are all confined to the inner loop of the flight control laws, thus regulating only the angular rates of the aircraft.

Moreover, all sensors have been idealized as having negligible noise and measurement errors, with only measurement delays modeled. Consequently, some performance degradation is expected during real-life flight tests, which could be mitigated by incorporating a suitable sensor model during the robust tuning of the controller, as was done with the ASE filters.

Since the demonstrations are intended as proofs of concept, the controller verification campaigns presented were limited in both duration and scope compared to what would typically be done in industry. For example, Monte Carlo campaigns usually involve tens, if not hun-

dreds, of thousands of randomized simulation trials. This volume of simulations generally requires high-performance computing, capable of running tasks in parallel, thus significantly speeding up the process. Lacking access to such equipment, a compromise was made by reducing the number of trials while still ensuring the relevance of the results.

As mentioned in Chapter 5, ASE filter tuning would benefit from an easy-to-use Singular Perturbation Margin assessment tool. In the absence of such a tool, manual identification of the problematic flexible modes appears inevitable. While this may be a missed opportunity to further reduce development costs, it reflects the current practice of ASE filter design, with the only exception being the need to duplicate filters to maintain synchronization across measurement signals.

The aircraft models used for demonstration purposes were based on publicly available data, which necessitated simplifications during implementation. As discussed in Chapters 2 and 5, finite-element modeling approaches are far more common in practice, as they offer greater accuracy and leverage data and models already produced by other teams involved in the aircraft's development. We suggested in Chapter 6 that an integrated modeling approach could allow control designers to provide a preliminary design before refining it as FEM data becomes available from other teams.

7.3 Future Research

The demonstrations presented in this thesis were sufficient as proofs of concept for the proposed approaches but would benefit from further reproduction and expansion, considering real-world constraints. For instance, [57] incorporated outer-loop control, realistic sensor models (accounting for characteristics such as bias, noise, delay, resolution, and sampling frequency), parametric uncertainties of aerodynamic coefficients and moments of inertia, saturation issues (mitigated with a Pseudo-Control Hedging (PCH) module), and various additional requirements. A similar demonstration on a flexible aircraft is now feasible, with several mitigation techniques available to address potential issues. Ideally, the entire suggested design process should be demonstrated in collaboration with an industrial partner.

Further research into the INDI architecture could yield new mitigation techniques or innovative variants. For example, the modified INDI architecture uncovers the presence of an underlying Smith Predictor loop that relies on a relatively poor approximation of the plant. Consequently, performance could be enhanced by using a more accurate model of the plant, which could be achieved by implementing proper plant models or incorporating an adaptive law.

REFERENCES

[1] M. Waszak, "Modeling the benchmark active control technology wind-tunnel model for application to flutter suppression," in *21st Atmospheric Flight Mechanics Conference*, 1996, p. 3437.

[2] Ü. Gülıat, *Fundamentals of Modern Unsteady Aerodynamics*. Berlin: Springer, 2010.

[3] P. Cheng and T. Hirner, "Automated procedures for aircraft aeroservoelastic compensation," in *Astrodynamics Conference*. American Institute of Aeronautics and Astronautics, Aug. 1992, p. 4606.

[4] L. Meirovitch and I. Tuzcu, "Integrated approach to the dynamics and control of maneuvering flexible aircraft," National Aeronautics and Space Administration (NASA), Tech.Rep. CR-2003-211748, 2003.

[5] H. Lhachemi, D. Saussié, and G. Zhu, "Performance enhancement of a self-scheduled longitudinal flight control system via multi-objective optimization," in *2014 American Control Conference*, IEEE. Institute of Electrical & Electronics Engineers (IEEE), jun 2014, pp. 1377–1383. [Online]. Available: http://dx.doi.org/10.1109/ACC.2014.6859147

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

[7] V. Mukhopadhyay, "Historical perspective on analysis and control of aeroelastic responses," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 5, pp. 673–684, 2003.

[8] T. Theodorsen, "General theory of aerodynamic instability and the mechanism of flutter," National Advisory Committee for Aeronautics (NACA), techreport 496, 1949.

[9] D. Gaukroger, E. Chapple, and A. Milln, "Wind-tunnel flutter tests on a model delta wing under fixed and free root conditions," Tech. Rep., 1955. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.226.9375&rep=rep1&type=pdf

[10] M. Patil, D. Hodges, and C. Cesnik, "Limit cycle oscillations of a complete aircraft," in *41st Structures, Structural Dynamics, and Materials Conference and Exhibit*, 2000, p. 1395.

[11] M. J. Patil, D. H. Hodges, and C. E. Cesnik, "Limit-cycle oscillations in high-aspect-ratio wings," *Journal of fluids and structures*, vol. 15, no. 1, pp. 107–132, 2001.

[12] M. J. Patil, D. H. Hodges, and C. E. S. Cesnik, "Nonlinear aeroelasticity and flight dynamics of high-altitude long-endurance aircraft," *Journal of Aircraft*, vol. 38, no. 1, pp. 88–94, 2001.

[13] M. Patil, "Limit cycle oscillations of aircraft due to flutter-induced drag," in *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2002, p. 1409.

[14] M. J. Patil, "From fluttering wings to flapping flight: The energy connection," *Journal of Aircraft*, vol. 40, no. 2, pp. 270–276, 2003.

[15] D. Thompson, Jr and T. Strganac, "Store-induced limit cycle oscillations and internal resonances in aeroelastic systems," in *41st Structures, Structural Dynamics, and Materials Conference and Exhibit*, 2000, p. 1413.

[16] D. E. Thompson and T. W. Strganac, "Nonlinear analysis of store-induced limit cycle oscillations," *Nonlinear Dynamics*, vol. 39, no. 1-2, pp. 159–178, 2005.

[17] D. Berggren, "Investigation of limit cycle oscillations for a wing section with nonlinear stiffness," *Aerospace Science and Technology*, vol. 8, no. 1, pp. 27–34, 2004.

[18] T. W. Strganac, J. Ko, and D. E. Thompson, "Identification and control of limit cycle oscillations in aeroelastic systems," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 6, pp. 1127–1133, 2000.

[19] G. Platanitis and T. W. Strganac, "Control of a nonlinear wing section using leading- and trailing-edge surfaces," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 52–58, 2004.

[20] S. Gujjula, S. N. Singh, and W. Yim, "Limit cycles and domain of stability in unsteady aeroelastic system," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 4, pp. 728–732, 2004.

[21] L. Abbas, Q. Chen, P. Marzocca, and A. Milanese, "Non-linear aeroelastic investigations of store (s)-induced limit cycle oscillations," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 222, no. 1, pp. 63–80, 2008.

[22] K. S. Dawson and D. L. Maxwell, "Limit cycle oscillation flight-test results for asymmetric store configurations," *Journal of Aircraft*, vol. 42, no. 6, pp. 1588–1595, 2005.

[23] S. Fazelzadeh, A. Mazidi, and H. Kalantari, "Bending-torsional flutter of wings with an attached mass subjected to a follower force," *Journal of Sound and Vibration*, vol. 323, no. 1-2, pp. 148–162, 2009.

[24] K. W. Lee and S. N. Singh, "Immersion-and invariance-based adaptive control of a nonlinear aeroelastic system," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1100–1110, 2009.

[25] D.-H. Lee and P. Chen, "Studies of aerodynamic influence of under-wing stores on flutter characteristics of F-16," in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, 2009, p. 2463.

[26] K. Lee and S. Singh, "Non-certainty-equivalent adaptive control of a nonlinear aeroelastic system," *International Journal of Electronics and Telecommunications*, vol. 56, no. 4, pp. 463–471, 2010.

[27] K. W. Lee and S. N. Singh, "Adaptive control of multi-input aeroelastic system with constrained inputs," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 2337–2350, 2015.

[28] A. A. Paranjape, J. Guan, S.-J. Chung, and M. Krstic, "Pde boundary control for flexible articulated wings on a robotic aircraft," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 625–640, 2013.

[29] B. J. Bialy, I. Chakraborty, S. C. Cekic, and W. E. Dixon, "Adaptive boundary control of store induced oscillations in a flexible aircraft wing," *Automatica*, vol. 70, pp. 230–238, 2016.

[30] R. M. Bennett, C. V. Eckstrom, J. A. Rivera Jr, B. E. Dansberry, M. G. Farmer, and M. H. Durham, "The benchmark aeroelastic models program: description and highlights of initial results," 1991.

[31] R. M. Bennett, R. C. Scott, and C. D. Wieseman, "Computational test cases for the benchmark active controls model," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 922–929, 2000.

[32] R. C. Scott, S. T. Hoadley, C. D. Wieseman, and M. H. Durham, "Benchmark active controls technology model aerodynamic data," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 914–921, 2000.

[33] D. E. Biskri, R. M. Botez, N. Stathopoulos, S. Therien, M. Dickinson, and A. Rathe, "New mixed method for unsteady aerodynamic force approximations for aeroservoelasticity studies," *Journal of aircraft*, vol. 43, no. 5, pp. 1538–1542, 2006.

[34] R. M. Botez, A. D. Dinu, I. Cotoi, N. Stathopoulos, S. Therien, M. Dickinson, and A. Rathé, "Improved method for creating time-domain unsteady aerodynamic models," *Journal of Aerospace Engineering*, vol. 20, no. 3, pp. 204–208, 2007.

[35] H. Dunn, "An analytical technique for approximating unsteady aerodynamics in the time domain," National Aeronautics and Space Administration (NASA), Tech.Rep. NASA Technical Paper 1738, 1980.

[36] J. R. Wright and J. E. Cooper, *Introduction to aircraft aeroelasticity and loads*. John Wiley & Sons, 2008, vol. 20.

[37] V. Mukhopadhyay, "Transonic flutter suppression control law design and wind-tunnel test results," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 930–937, 2000.

[38] T. Lieu, C. Farhat, and M. Lesoinne, "Reduced-order fluid/structure modeling of a complete aircraft configuration," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 41-43, pp. 5730–5742, 2006.

[39] X. Wang, E. Van Kampen, Q. Chu, and R. De Breuker, "Flexible aircraft gust load alleviation with incremental nonlinear dynamic inversion," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1519–1536, 2019.

[40] H. Lhachemi, D. Saussié, and G. Zhu, "Explicit hidden coupling terms handling in gain-scheduling control design via eigenstructure assignment," *Control Engineering Practice*, vol. 58, pp. 1 – 11, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066116301964

[41] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[42] H. Lhachemi, D. Saussié, and G. Zhu, "Boundary feedback stabilization of a flexible wing model under unsteady aerodynamic loads," *Automatica*, vol. 97, pp. 73–81, 2018.

[43] M. J. Balas, "Active control of flexible systems," *Journal of Optimization Theory and Applications*, vol. 25, no. 3, pp. 415–436, 1978.

[44] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard h/sub 2/and h/sub infinity/control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, 1989.

[45] M. G. Safonov, D. Limebeer, and R. Chiang, "Simplifying the h/sub infinity/ theory via loop-shifting, matrix-pencil and descriptor concepts," *International Journal of Control*, vol. 50, no. 6, pp. 2467–2488, 1989.

[46] P. Apkarian and D. Noll, "Nonsmooth h_infty synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[47] P. Gahinet and P. Apkarian, "Structured H_∞ synthesis in MATLAB," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1435–1440, jan 2011. [Online]. Available: http://dx.doi.org/10.3182/20110828-6-IT-1002.00708

[48] ——, "Automated tuning of gain-scheduled control systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 2740–2745. [Online]. Available: http://pierre.apkarian.free.fr/papers/GainScheduling_cdc2013.pdf

[49] P. Apkarian, P. Gahinet, and C. Buhr, "Multi-model, multi-objective tuning of fixed-structure controllers," in *2014 European Control Conference (ECC)*. IEEE Publ. Piscataway, NJ, 2014, pp. 856–861. [Online]. Available: http://pierre.apkarian.free.fr/papers/MultiModReq.pdf

[50] P. Apkarian, M. N. Dao, and D. Noll, "Parametric robust structured control design," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1857–1869, 2015. [Online]. Available: https://arxiv.org/pdf/1405.4202.pdf

[51] H. Lhachemi, "Synthèse et validation d'un système de commandes de vol robuste et autoséquencé," Master's thesis, École Polytechnique de Montréal, 2013.

[52] É. Jacquet, "Commande structurée h-infinie de la dynamique rigide et aéro-élastique d'un avion civil," Master's thesis, École Polytechnique de Montréal, 2018.

[53] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[54] G. Puyou, C. Berard, G. Ferreres, and P. Menard, "Conception multi-objectifs de lois de pilotage pour un avion de transport civil," *Journal Européen des Systèmes Automatisés (JESA)*, vol. 4, no. 9-1, pp. 1019–1051, 2006.

[55] H. Lhachemi, D. Saussié, and G. Zhu, "Handling hidden coupling terms in gain-scheduling control design: Application to a pitch-axis missile autopilot," in *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 0365.

[56] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

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

[58] F. Lavergne, F. Villaume, M. Jeanneau, S. Tarbouriech, and G. Garcia, "Nonlinear robust autoland," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, p. 5848.

[59] P. Smith, "A simplified approach to nonlinear dynamic inversion based flight control," in *23rd Atmospheric Flight Mechanics Conference*, 1998, p. 4461.

[60] J. F. Buffington, "Modular control law design for the innovative control effectors (ICE) tailless fighter aircraft configuration 101-3," Air Force Research Lab Wright-Patterson AFB OH Air Vehicles Directorate, Tech. Rep., jun 1999.

[61] P. Smith and A. Berry, "Flight test experience of a non-linear dynamic inversion control law on the vaac harrier," in *Atmospheric Flight Mechanics Conference*, 2000, p. 3914.

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

[63] R. van't Veld, E.-J. Van Kampen, and Q. P. Chu, "Stability and robustness analysis and improvements for incremental nonlinear dynamic inversion control," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1127.

[64] S. Sieberling, Q. Chu, and J. Mulder, "Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1732–1742, 2010.

[65] Z. Wang, J. Zhao, Z. Cai, Y. Wang, and N. Liu, "Onboard actuator model-based incremental nonlinear dynamic inversion for quadrotor attitude control: Method and application," *Chinese Journal of Aeronautics*, vol. 34, no. 11, pp. 216–227, nov 2021.

[66] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, may 2021.

[67] P. Simplício, M. Pavel, E. Van Kampen, and Q. Chu, "An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion," *Control Engineering Practice*, vol. 21, no. 8, pp. 1065–1077, 2013.

[68] C. Cakiroglu, E.-J. Van Kampen, and Q. P. Chu, "Robust incremental nonlinear dynamic inversion control using angular accelerometer feedback," in *2018 AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, jan 2018, p. 1128.

[69] R. Steffensen, A. Steinert, and E. J. J. Smeur, "Non-linear dynamic inversion with actuator dynamics: an incremental control perspective," *Journal of Guidance, Control and Dynamics*, Jan. 2022.

[70] W. Wang, P. Menon, D. Bates, S. Ciabuschi, N. Paulino, E. Sotto, A. Bidaux, A. Garus, A. Kron, S. Salehi *et al.*, "Verification and validation of autonomous rendezvous systems in the terminal phase," in *AIAA Guidance, Navigation, and Control Conference*, 2012, p. 4925.

[71] A. Marcos, H. Garcia De Marina, V. Mantini, C. Roux, and S. Bennani, "Optimization-based worst-case analysis of a launcher during the atmospheric ascent phase," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 4645.

[72] A. Marcos and S. Bennani, "Lpv modeling, analysis and design in space systems: Rationale, objectives and limitations," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5633.

[73] F. Laliberté and D. Saussié, "Relaxing the applicability conditions for Incremental Nonlinear Dynamics Inversion," *Aerospace Science and Technology*, 2025, (Submitted, not yet published).

[74] ——, "Robust tuning of the modified Incremental Nonlinear Dynamics Inversion for a rigid aircraft," *Journal of Guidance, Control, and Dynamics*, (not yet published).

[75] ——, "Robust control for flexible aircraft using Incremental Nonlinear Dynamics Inversion with notch filters," *Journal of Guidance, Control, and Dynamics*, 2025, (Submitted, not yet published).

[76] R. van't Veld, "Incremental nonlinear dynamic inversion flight control: Stability and robustness analysis and improvements," Master's thesis, Delft University of Technology, the Netherlands, 2016.

[77] H. Chang, Y. Liu, Y. Wang, and X. Zheng, "A modified nonlinear dynamic inversion method for attitude control of uavs under persistent disturbances," in *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2017, pp. 715–721.

[78] I. Matamoros and C. C. de Visser, "Incremental nonlinear control allocation for a tailless aircraft with innovative control effectors," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1116.

[79] J. Zhang, P. Bhardwaj, S. A. Raab, S. Saboo, and F. Holzapfel, "Control allocation framework for a tilt-rotor vertical take-off and landing transition aircraft configuration," in *2018 Applied Aerodynamics Conference*, 2018, p. 3480.

[80] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 703–711, 2002.

[81] D. J. Leith and W. Leithead, "Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families," *International Journal of Control*, vol. 70, no. 2, pp. 289–317, 1998.

[82] M. D. Adams, *Signals and Systems (Edition 3.0)*. Michael Adams, 2020.

[83] D. Gaudette and D. Davison, "Performance limitations imposed by sensor time delays in a general 2-DOF control scheme," in *2006 American Control Conference*. IEEE, 2006, pp. 6 pp.–.

[84] D. E. Davison and R. Tonita, "Performance limitations in control systems with sensor time delays," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 225–230, 2005.

[85] N. Abe and K. Yamanaka, "Smith predictor control and internal model control - a tutorial," in *SICE 2003 Annual Conference (IEEE Cat. No.03TH8734)*, vol. 2, 09 2003, pp. 1383 – 1387 Vol.2.

[86] M. Sandström, "Investigation of and compensation for time-delays in driveline control systems," Master's thesis, KTH, Optimization and Systems Theory, 2014.

[87] G. Puyou, "Multi-objective design of piloting laws for a civil transport aircraft," Ph.D. dissertation, National School of Aeronautics and Space, Toulouse, France, 2005.

[88] J. Boiffier, *The Flight Dynamics: The Equations*. John Wiley and Sons, 1998.

[89] MIL-STD-1797B, "Military and government specs & standards (NPFC): Flying qualities of piloted aircraft," Feb. 2006. [Online]. Available: https://global.ihs.com/doc_detail.cfm?document_name=MIL-STD-1797&item_s_key=00286990

[90] R. Alkhatib and M. F. Golnaraghi, "Active structural vibration control: A review," *The Shock and Vibration Digest*, vol. 35, no. 5, pp. 367–383, sep 2003.

[91] Y. Kumtepe, T. Pollack, and E.-J. V. Kampen, "Flight control law design using hybrid incremental nonlinear dynamic inversion," in *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, jan 2022, p. 1597.

[92] Y. Zhou, H. W. Ho, and Q. Chu, "Extended incremental nonlinear dynamic inversion for optical flow control of micro air vehicles," *Aerospace Science and Technology*, vol. 116, p. 106889, sep 2021.

[93] Y. Li, X. Liu, P. Lu, Q. He, R. Ming, and W. Zhang, "Angular acceleration estimation-based incremental nonlinear dynamic inversion for robust flight control," *Control Engineering Practice*, vol. 117, p. 104938, dec 2021.

[94] International Organization for Standardization (ISO), "International Standard Atmosphere (ISA) [ISO 2533:1975]," 1975.

[95] P. Seiler, A. Packard, and P. Gahinet, "An introduction to disk margins [lecture notes]," *IEEE Control Systems Magazine*, vol. 40, no. 5, pp. 78–95, 2020.

[96] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[97] P. Menon, E. Prempain, I. Postlethwaite, D. Bates, and S. Bennani, "Nonlinear worst-case analysis of an LPV controller for approach-phase of a re-entry vehicle," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5638.

[98] I. H. Chan, "Swept sine chirps for measuring impulse response," *(White paper)*, 2010.

[99] E. Albano and W. P. Rodden, "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows." *AIAA Journal*, vol. 7, no. 2, pp. 279–285, 1969.

[100] T. Bailey and J. E. Hubbard, "Distributed piezoelectric-polymer active vibration control of a cantilever beam," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 605–611, 1985.

[101] M. J. Balas, "Singular perturbations in stable feedback control of distributed parameter systems," *IFAC Proceedings Volumes*, vol. 15, no. 5, pp. 241–267, 1982.

[102] P. Friedmann, D. Guillot, and E. Presente, "Adaptive control of aeroelastic instabilities in transonic flow and its scaling," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 6, pp. 1190–1199, 1997.

[103] M. Balas and S. Harvey, "Residual mode filter/model reduction compensation and application to f-16 dynamic models," in *2007 U.S. Air Force T & E Days*. American Institute of Aeronautics and Astronautics, Feb. 2007, p. 1650.

[104] E. Smeur, G. de Croon, and Q. Chu, "Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection," *Control Engineering Practice*, vol. 73, pp. 79–90, apr 2018.

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

[106] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear incremental control for flexible aircraft trajectory tracking and load alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 1, pp. 39–57, Jan. 2022.

[107] E. Mooij, "Robust control of a conventional aeroelastic launch vehicle," in *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2020, p. 1103.

[108] X. Yang and J. J. Zhu, "Singular perturbation margin and generalised gain margin for nonlinear time-invariant systems," *International Journal of Control*, vol. 89, no. 3, pp. 451–468, Sep. 2015.

[109] F. Laliberté, "Modélisation et commande d'un drone hélicoptère tandem," Master's thesis, École Polytechnique de Montréal, 2017.

[110] P. C. Young, *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer Science & Business Media, 2011.

APPENDIX A USER GUIDE: STRUCTURED $\mathcal{H}_\infty$ SYNTHESIS

This appendix aims at describing the features of the `systune` function in MATLAB as to help exploit its potential. This function uses the `TuningGoal` class of objects to translate requirements into the appropriate $\mathcal{H}_\infty/\mathcal{H}_2$ norms and attempts to fulfill them all at once. Requirements can be specified either as an objective (to be optimized as much as possible) or a constraint (to respect a given threshold without optimizing further), which are referred to as "soft" and "hard" tuning goals in MATLAB documentation.

This appendix contains three sections: the first presents how to model a system for use with `systune`; the second describes the different types of requirement that are supported by the `TuningGoal` class; and the last is about the application of the different types of requirement, which have been divided between the topics of trajectory tracking and disturbance rejection.

A.1 Modeling

To use the different requirements (needed in order to use `systune`), all points of interest in the system must be named. This is caused by the constructor functions for requirements that only accept names (string of characters) to designate the different points of interest. The interest points that can be supplied as argument in requirements (`TuningGoal`) are:

- System's inputs and outputs (`InputName` and `OutputName` model properties)
- Analysis points (`AnalysisPoint` elements must be included in the model at the desired location)
- Tunable SIMULINK blocks

In order to use `systune`, the model provided to this function must contain all points of interest used in the requirements and must be one of the following types of dynamic system models:

- A generalized state-space `genss`
- A SIMULINK model interfaced by `slTuner`

The specified model must have tunable elements and may contain uncertain elements.

It is to note that the analysis points are meant to designate loops:

- for a requirement that applies on a loop; the constructor function of the requirement adds an output before and an input after the analysis point and then analyze the transfer function between the two (e.g. sensibility function).
- for any requirement on a system with multiple loops; `TuningGoal` have properties that allows the opening of loops for a given requirement (e.g. ignoring the outerloop for an innerloop requirement).

For example, the controller output and the feedbacks (when multiple loops are present) are often analysis points.

A.1.1 Build a generalized state-space model `genss` for use with `systune`

- **Define tunable parameters**. This can be done through predefined structures (blocks of class `TunablePID`, `TunableTF`, `TunableSS`, or `TunableGain`) or through a custom structure containing `realp` parameters.
- **Name all inputs and outputs of subsystems**.
- **Interconnect the fixed and tunable elements**. This is done with the help of the `feedback` and `connect` functions. Use the `AnalysisPoint` function to create "analysis point" elements and include them during connection (e.g. `feedback`(G*X1*F,1) where X1 is an analysis point between systems F and G).

The advantage of this approach is that a vector of generalized state-space models can also be supplied to the `systune` function.

A.1.2 Build a SIMULINK model to be interfaced with `slTuner` for use with `systune`

- **Build a SIMULINK model**.
- **Name all blocks and signals of interest**.
- **Create a `slTuner` interface**. The `slTuner` function returns a `slTuner` interface object and takes as argument the name of the SIMULINK model to interface and a cell array containing the name of all tunable blocks (e.g. the name of the Gain or PID blocks mean to be part of the controller).

- **Designate the analysis points**. The `addPoint` function must be used to add analysis point, which takes as argument the `slTuner` interface and a cell array containing all points of interests (to be added to the list of analysis points). Note that the inputs and outputs of the system must also be added this way.

The advantage of this approach is that it is easier to build the connections between subsystems and that it is easier to simulate the final result after synthesis (the simulator can be used directly to create the interface), and that it is possible to use SIMULINK block names as argument in `TuningGoal` constructor functions.

A.2 `TuningGoal`

In order to specify the objectives and constraints of the optimization with the help of `systune`, the `TuningGoal` class must be used. The different types of goals are grouped into the following five categories:

- Time domain requirements
- Frequency domain requirements
- Loop shapes and stability margins
- Passivity
- System dynamics

Note that every goal can be used as either an objective (function to minimize) or a constraints (function to be brought under 1).

Furthermore, it is possible to add scaling factors and to open certain loops (using analysis points) for the majority of these goal types. Opening loops allows, for instance, to enforce innerloop requirements by opening the outerloops.

For ease of reading, whenever a frequency-domain function refers to "the system", it is implied that this means the closed-loop system's transfer function between the specified input and output (which may be multivariate).

A.2.1 Time domain

The goals that can be expressed in the time domain are:

- `StepTracking` - Tracking of a reference model's time response to a unit step input

- `StepRejection` - Tracking of a reference model's time response to a unit step disturbance
- `Transient` - Matching with a reference model's transient time response
- `LQG` - Goal framed as a cost function to be optimized

A.2.1.1 `StepTracking`

This requirement allows the specification of a tracking goal. This requirement can be expressed in three ways:

- Provide the desired time constant, thus the response time aimed for. The reference model used by the algorithm will be of order 1.
- Provide the desired time constant and the desired overshoot percentage. The reference model used by the algorithm will be of order 2, deducing the desired damping ratio from the desired overshoot.
- Provide the reference model directly (can be multivariate (MIMO)).

The algorithm used for this requirement consists in minimizing (or bring under 1) the following function:

$$f(x) = \frac{\left\| \frac{1}{s}(T(s,x) - R(s)) \right\|_2}{0.1 \left\| \frac{1}{s}(R(s) - I) \right\|_2}$$

where x is the vector of tunable parameters, T is the system, R is the reference model, and where the 0.1 coefficient is customizable (`RelGap` property).

This function is a measure of the matching error between the system and the reference model. This function is equal to 1 if the time response of the system is at most 0.1 (or at another value if `RelGap` was changed), which is the maximal tolerable matching error. This function is equal to zero if the system's response is identical to the reference model's. The algorithm thus seeks to ensure that the maximal tolerable matching error is respected and to minimize the (worst-case) matching error.

Note that given the nature of the algorithm[1], `systune` will throw an error if it cannot force the controller's feedthrough term to zero (if the matrix $\mathbf{D}$ of the controller's state-space formulation cannot be brought to zero or if the transfer function of the controller is not strictly proper).

[1]This algorithm uses the energy of the signal ($\mathcal{H}_2$ norm) which is infinite if the controller as a non-zero feedthrough term.

A.2.1.2 `StepRejection`

This requirement allows the specification of a perturbation rejection goal. It distinguish itself from `StepTracking` by its customizable parameters and optimization function. This requirement can be expressed in three ways:

- Provide the desired maximal amplitude and response time. The reference model used by the algorithm will be of order 1.
- Provide the desired maximal amplitude, response time and damping ratio. The reference model used by the algorithm will be of order 2.
- Provide the reference model directly (can be multivariate (MIMO)).

The algorithm used for this requirement consist in minimizing (or bring under 1) the following function:

$$f(x) = \left\| \frac{T(s,x)}{R(s)} \right\|_\infty$$

where x is the vector of tunable parameters, T is the system, and R is the reference model.

This function is a measure of the maximal amplitude of the system relative to the reference model. This function is equal to 1 if the system's time response to the perturbation is identical to the reference model's, or if the system's peak amplitude, response time and/or damping ratio matches the reference model's (and that the other characteristics are better: smaller amplitude peak, faster response, higher damping). This function is equal to 0 if the system is insensitive to the considered perturbation. The algorithm thus seeks to minimize the maximal amplitude and response time, and to maximize the damping ratio, of the system following a step perturbation, and thus minimize its sensitivity to the perturbation. The reference model is meant to indicate the tolerable maximal amplitude, maximal response time, and minimal damping ratio.

Note that the actual MATLAB implementation of this goal adjusts the weighting filter $\mathbf{W} = \frac{1}{\mathbf{R}(s)}$ to regularize it for numerical conditioning reasons.

A.2.1.3 `Transient`

This requirement can be seen as a generalization of `StepTracking`. This requirement consists in following a specified reference model's output to a specified input. The specified input can be an impulse (default), a step, a ramp or a custom signal (provided as a transfer function).

The algorithm used for this requirement consists in minimizing (or bring under the `RelGap`

property value) the following function:

$$f(x) = \frac{\|y(t,x) - y_{ref}(t)\|_2}{\|1 - y_{ref}(t)\|_2}$$

where x is the vector of tunable parameters, y is the system's response, y_{ref} is the reference model's response, and $1 - y_{ref}(t)$ designates the transient portion of the reference model's response (gap relative to the steady state value).

Note that `systune` will throw an error if the feedthrough term of the controller cannot be brought to zero for the same reason as with the use of `StepTracking`.

A.2.1.4 `LQG`

This requirement express an objective (or constraint) through a LQG cost function. This function has the following format:

$$J = E(z(t)'\mathbf{Q}_z z(t))$$

where $z(t)$ is the system's time response to a white noise input (whose covariance matrix $\mathbf{Q}_w$ must be provided), $\mathbf{Q}_z$ is the weighting matrix to be provided, and $E(.)$ is the expectation (mean of a random variable, similar to a weighted integral).

Note that `systune` will throw an error if the feedthrough term of the controller cannot be brought to zero for the same reason as with the use of `StepTracking`.

A.2.2 Frequency domain

The goals that can be expressed in the frequency domain are:

- `Gain` - Maximum gain in function of frequency
- `Variance` - Limits the amplification of white noise
- `Tracking` - Limits the tracking error
- `Overshoot` - Limits the overshoot (but does not optimize it)
- `Rejection` - Minimum attenuation factor of a perturbation
- `Sensitivity` - Limits sensitivity to perturbations
- `WeightedGain` - Custom weighting filter ($\mathcal{H}_\infty$ norm)
- `WeightedVariance` - Custom weighting filter ($\mathcal{H}_2$ norm)

A.2.2.1 `Gain`

This requirement allows the specification of a maximal limit on amplitudes in function of frequency. This requirement can be expressed in two ways:

- Provide the maximum tolerable gain for all frequencies (the considered frequency band can be restricted with the `Focus` property).
- Provide the maximum error profile (in function of frequency).

Application examples include:

- enforce a gain smaller than 1 on all frequencies to ensure disturbance rejection;
- enforce a custom attenuation profile on a desired frequency band.

The algorithm used for this requirement consist in minimizing (or bring under 1) the following function:

$$f(x) = \left\| \frac{T(s,x)}{gain_{max}} \right\|_\infty$$

where x is the vector of tunable parameters, T is the system, and $gain_{max}$ is the maximum gain (or maximum gain profile in function of frequency).

This function is a normalized measure of the maximum amplitude of the system. This function is equal to 1 if the system's amplitude reaches the specified limit (that can vary with frequency) at one point in the considered frequency band (limit is reach for at least once frequency in the band). This function equals zero if the system has a null transfer. The algorithm thus seeks to ensure that the specified limit on amplitude is respected and to minimize the amplitude of the system.

A.2.2.2 `Variance`

This requirement allows the specification of a maximum limit on the amplification of a white noise. The algorithm used for this requirement consists in minimizing (or bring under 1) the following function:

$$f(x) = \left\| \frac{T(s,x)}{amplification_{max}} \right\|_2$$

where x is the vector of tunable parameters, T is the system, and $amplification_{max}$ is the maximum tolerable amplification.

This function is a normalized measure of the total energy of the impulse response of the system, which is a normalized measure of noise amplification. This function equals 1 if noise

amplification by the system reaches the imposed limit. This function equals zero if the output is deemed insensitive to the considered white noise. The algorithm thus seeks to minimize noise amplification and to ensure that the amplification limit is respected.

Note that if the considered noise is not white, the `WeightedVariance` requirement should be used instead.

A.2.2.3 `Tracking`

This requirement allows the specification of a maximal limit of the tracking error in function of frequency. This requirement can be expressed in two ways:

- Provide the maximum steady-state error, maximum peak (overshoot), and maximum settling time.
- Provide the maximum tracking error profile (in function of frequency).

The algorithm used for this requirement consists in minimizing (or bring under 1) the following function:

$$f(x) = \left\| \frac{(T(s,x) - I)}{error_{max}} \right\|_\infty$$

where x is the vector of tunable parameters, T is the system, and $error_{max}$ is the maximum error (or maximum error profile in function of frequency).

This function is a normalized measure of the maximum tracking error of the system. This function equals 1 if the tracking error reaches the specified limit (that can vary with frequency) at one point in the considered frequency band (limit is reach for at least once frequency in the band). This function equals zero if the system has a unit transfer function, which implies perfect tracking (output = input). The algorithm thus seeks at ensuring that the specified limit on tracking error is respected and that the tracking error of the system is minimized.

A.2.2.4 `Overshoot`

This requirement allows the enforcement of a limit on overshoot. The algorithm used approximates[2] the overshoot by the norm $\|T(s,x)\|_\infty$ and cannot enforce an overshoot smaller than 5%.

[2]This norm yields the exact overshoot for a second-order system and is only an approximation for higher-order systems.

A.2.2.5 `Rejection`

This requirement allows the specification of a minimum attenuation factor of a given perturbation. The attenuation faction is the ratio between the system's open-loop sensitivity to the considered perturbation and the system's closed-loop sensitivity to the same perturbation (the ratio must be larger than 1 to have attenuation). The algorithm minimizes the ratio

$$\max_{\omega} \|W_f(j\omega)S(j\omega, x)\|_\infty$$

where S is the system's closed-loop sensitivity function to the considered perturbation, and W_f is a weighting function computed from the provided minimum attenuation factor. A ratio of 1 indicates that the minimum factor is reached and a ratio below 1 indicates an attenuation factor above the specified limit. The ratio equals 0 if the closed-loop system is insensitive to the given perturbation (the attenuation factor then becomes infinite if the open-loop is not insensitive to this perturbation).

A.2.2.6 `Sensitivity`

This requirement allows the specification of a maximum sensitivity to a given perturbation. The algorithm minimizes the following function:

$$f(x) = \left\| \frac{S(s, x)}{Sensitivity_{max}} \right\|_\infty$$

where x is the vector of tunable parameters, S is the system's closed-loop sensitivity function to the considered perturbation, and $Sensitivity_{max}$ is the maximum tolerable sensitivity (may vary with frequency). This function equals 1 if the sensitivity limit is reached (but not exceeded) and is nul if the system is insensitive to the given perturbation.

This requirements distinguish itself by its absolute criterion, where `Rejection` uses a relative criterion.

A.2.2.7 `WeightedGain`

This requirement allows the use of custom weighting filters at the input and output to represent various goals as a $\mathcal{H}_\infty$ norm to minimize.

The algorithm for this requirement minimizes the norm

$$\|W_o(s)T(s, x)W_i(s)\|_\infty$$

where x is the vector of tunable parameters, T is the system, W_o is the output weighting

filter, and W_i is the input weighting filter.

This requirement thus allows the direct use of $\mathcal{H}_\infty$ Synthesis requirements.

A.2.2.8 `WeightedVariance`

This requirement allows the use of custom weighting filters at the input and output to represent various goals as a $\mathcal{H}_2$ norm to minimize.

The algorithm for this requirement minimizes the norm

$$\|W_o(s)T(s,x)W_i(s)\|_2$$

where x is the vector of tunable parameters, T is the system, W_o is the output weighting filter, and W_i is the input weighting filter.

As this norm allows the minimization of the energy of the system's response to a given input, it is possible to use this requirement to represent LQG performance criteria, or to represent rejection criteria for colored noises or non-uniform perturbations (such as gusts).

A.2.3 Loop shapes and stability margins

The following requirements concerns the gains of a given loop (unlike the requirement of the two previous categories that considered the gains of given transfer functions):

- `MinLoopGain` - Minimum gain (of the loop) in function of frequency
- `MaxLoopGain` - Maximum gain (of the loop) in function of frequency
- `LoopShape` - Limits the shape of the loop (equivalent to using `MinLoopGain` for low frequencies and `MaxLoopGain` for the high ones simultaneously)
- `Margins` - Minimum stability margins (gain and phase)

A.2.3.1 `MinLoopGain`

This requirement allows to impose a lower limit (may vary with frequency) on a given loop's gain, or on the singular values of a given loop if the system is multivariate (MIMO).

The algorithm minimizes the following function:

$$\|W_s S(s,x)\|_\infty$$

where x is the vector of tunable parameters, S is the sensitivity function of the given loop, and

where W_s is a weighting filter computed from the specified limit (may vary with frequency). Since the sensitivity function of a loop is inversely proportional to its gain[3], minimizing the sensitivity allows the maximization of the loop gain. This function is below 1 when the loop gain is above the limit.

Due to the approximations[4] used by the algorithm, it proves better suited for limits imposed on low frequencies.

A.2.3.2 `MaxLoopGain`

This requirement allows to impose an upper limit (may vary with frequency) on a given loop's gain, or on the singular values of a given loop if the system is multivariate (MIMO).

The algorithm minimizes the following function:

$$\|W_t T(s,x)\|_\infty$$

where x is the vector of tunable parameters, T is the complementary sensitivity function of the given loop, and where W_t is a weighting filter computed from the specified limit (may vary with frequency). Since the complementary sensitivity function of a loop is proportional to its gain[5], minimizing the complementary sensitivity allows the minimization of the loop gain. This function is below 1 when the loop gain is under the limit.

Due to the approximations[6] used by the algorithm, it proves better suited for limits imposed on high frequencies.

A.2.3.3 `LoopShape`

This requirement allows to impose a gain profile in function of frequency for an open-loop response.

The gain profile can be specified in the following ways:

- Provide the gain profile in function of frequency directly (using the `frd` command) with or without a tolerance on the cutoff frequency
- Provide the desired cutoff frequency, with or without tolerance (the desired profile will be assumed to match an integrator, i.e. ω_c/s)

[3]By definition, $S = \frac{1}{1+L}$ where L is the (open-loop) transfer function of the given loop. The amplitude of L is what is designated as "loop gain".

[4]Note that the algorithm uses the following approximation: $|S| \approx |L|$ when $|L| >> 1$.

[5]By definition, $T = 1 - S = \frac{L}{1+L}$ where L is the (open-loop) transfer function of the given loop. The amplitude of L is what is designated as "loop gain".

[6]Note that the algorithm uses the following approximation: $|T| \approx |L|$ when $|L| << 1$.

The algorithm then uses the algorithms of `MinLoopGain` for lower frequnecies and of `MaxLoopGain` for the higher frequencies, while computing the weighting filters (W_s and W_t) for both from the given gain profile and tolerance on the cutoff frequency (0.1 by default). The function to minimize is thus:

$$f(x) = \left\| \begin{matrix} W_s S(s,x) \\ W_t T(s,x) \end{matrix} \right\|_\infty$$

The terms "lower frequencies" and "higher frequencies" are used relative to the cutoff frequency.

A.2.3.4 `Margins`

This requirement allows the specification of a minimum gain margin and a minimum phase margin for a given loop.

The algorithm minimizes the following function:

$$\|2\alpha S(s,x) - \alpha I\|_\infty$$

where x is the vector of tunable parameters, S is the sensitivity function of the given loop, and α is a parameter computed from the specified margins. The norm used by the algorithm is in fact the inverse of the "disk margin". This type of stability margins regroups both gain and phase margins into a single value, and is notably used to represent the margins of multivariate systems. The α parameter is in fact the disk margin equivalent of the desired margins[7], which is

$$\alpha = \max\left[\frac{GM-1}{GM+1}, \tan(PM/2)\right]$$

where GM is the desired minimum gain margin (in absolute unit) and PM is the desired phase margin (in rad).

A.2.4 Passivity

The requirements concerning passivity are[8]:

- `Passivity` - Constraints the system to be passive

[7]The value of α can be verified by computing the disk margin $DM = \frac{1+\alpha}{1-\alpha}$ and using the command `dmplot`(DM). This last commandd generates a plot showing the equivalent gain and phase margins (in dB and °).

[8]This user guide was first created before the addition of `ConicSector`, and the associated improvement of the MATLAB environment. Please refer to https://www.mathworks.com/help/control/ug/sector-bounds-and-sector-indices.html for further details about conic sectors.

- `WeightedPassivity` - Constraints the weighted system to be passive

A passive system is a system that cannot generate energy by itself; a greater output signal must require a greater input signal. A system G is called passive if its frequency response is real and positive, thus if it fulfills the following relation:

$$G(j\omega) + G(j\omega)^H > 0$$

The interesting properties of a passive system are:

- All passive systems are stable.
- The inverse of a passive system is passive.
- The interconnection in parallel of passive systems is passive.
- The feedback interconnection of passive systems is passive.

If a system is passive, the use of a passive control law in the feedback loop can ensure closed-loop stability (even if the regulated system contains unknown or varying elements).

A.2.4.1 `Passivity`

This requirement allows the addition of passivity constraint on the system between the specified input and output. It is also possible to add a lower limit on excess of passivity[9] relative to the input and relative to the output. The inner workings of this algorithm involves the computing of conic sector indexes[10] and falls outside the scope of this document.

A.2.4.2 `WeightedPassivity`

This requirement allows the addition of passivity constraint on the system between the specified input and output. This requirement is essentially identical to `Passivity`, except that it allows the addition of weighting filters (functions of frequency) t the input and output of the system.

A.2.5 System dynamics

The requirements concerning the systems dynamics (i.e. its poles) are:

[9] Use the command `getPassiveIndex` to obtain the relative passivity index.

[10] Please refer to https://www.mathworks.com/help/control/ug/sector-bounds-and-sector-indices.html for further details about conic sectors.

- `Poles` - Restrict the system's closed-loop poles in the complex plane
- `ControllerPoles` - Restrict a tunable sub-system's poles in the complex plane

A.2.5.1 `Poles`

This requirement allows the definition of an acceptable region of the complex plane for the poles of the closed-loop system. This acceptable region is defined through the following criteria:

- Minimum frequency (or minimum decay rate)
- Maximum frequency
- Minimum damping ratio

A.2.5.2 `ControllerPoles`

This requirement allows the definition of an acceptable region for the poles of a tunable sub-system. This acceptable region is defined in the same ways as for `Poles`, but the SIMULINK block to be constraned must be specified (and must contain tunable elements). For example, a multivariate system containing 4 PID controllers can have one specific PID constrained by a `ControllerPoles` goal.

A.3 Applications

The different types of requirements allows the enforcement of various constraints during controller synthesis with `systune`. The two most common tuning goals are concerned with tracking performance and disturbance rejection performance.

Table A.1 presents the types of requirement that can be used to enforce objectives (and constraints) related to tracking.

Table A.2 presents the types of requirement that can be used to enforce objectives (and constraints) related to the rejection of noises and perturbations.

Reminder that the algorithm based on the $\mathcal{H}_2$ norm must be able to cancel the feedthrough term of the controller (between specified input and output) to be able to convergence.

Table A.1 Requirements suitable for tracking

Requirement	Tunable parameters	Advantages	Algorithm	Application
`StepTracking`	Settling time, damping ratio; reference model, and max relative error	Attempts to meet targets	$\mathcal{H}_2$ norm	I/O transfer
`Tracking`	Settling time, steady-state error, max overshoot; max error profile	Attempts to exceed targets	$\mathcal{H}_\infty$ norm	I/O transfer
`Transient`	Reference model, Max relative error.	Like `StepTracking`, but free to specify the input	$\mathcal{H}_2$ norm	I/O transfer
`LQG`	Weighting, noise variance	Customizable optimization	$\mathcal{H}_2$ norm	I/O transfer
`Gain`	Max gain, frequency band	Customizable frequency profile	$\mathcal{H}_\infty$ norm	I/O transfer
`WeightedGain`	Input & Output weighting filters	Customizable frequency profile	$\mathcal{H}_\infty$ norm	I/O transfer
`WeightedVariance`	Input & Output weighting filters	Customizable frequency profile, can express a weighting similar to a LQG objective	$\mathcal{H}_2$ norm	I/O transfer
`MinLoopGain`	Min gain, frequency band; min gain at a specific frequency (for an integral profile)	Customizable frequency profile	$\mathcal{H}_\infty$ norm	Loop
`LoopShape`	Desired frequency profile, tolerance (between min/max gains); desired cutoff frequency (assuming a desired integral frequency profile), tolerance	Like `MinLoopGain`, but combined with `MaxLoopGain` (with a tolerance between the 2)	$\mathcal{H}_\infty$ norm	Loop
`Poles`	Min damping ratio, min & max frequencies	Customizable region of the complex plane	N/A	Loop, entire system

Table A.2 Requirements suitable for disturbance rejection

Requirement	Tunable parameters	Advantages	Algorithm	Application
`StepRejection`	Max amplitude, max settling time, min damping ratio; reference model	Attempts to rejects perturbations better than the reference model	$\mathcal{H}_\infty$ norm	I/O transfer
`Gain`	Max gain, frequency band	Customizable frequency profile	$\mathcal{H}_\infty$ norm	I/O transfer
`Rejection`	Min attenuation factor, frequency band	Allows the enforcement of a non-degradation of open-loop performance (indirect sensitivity criterion)	$\mathcal{H}_\infty$ norm	Loop
`Sensitivity`	Max sensitivity, frequency band	Directly limits the gains of the sensitivity function (absolute criterion)	$\mathcal{H}_\infty$ norm	Loop
`Variance`	Max limit on the amplification of white noise, white noise variance	Designed for white noises	$\mathcal{H}_2$ norm	I/O transfer
`WeightedVariance`	Input & Output weighting filters	Customizable frequency profile, can be used to reject colored noises	$\mathcal{H}_2$ norm	I/O transfer
`WeightedGain`	Input & Output weighting filters	Customizable frequency profile	$\mathcal{H}_\infty$ norm	I/O transfer
`MaxLoopGain`	Max gain, frequency band; max gain at a specific frequency (for an integral profile)	Customizable frequency profile	$\mathcal{H}_\infty$ norm	Loop
`LoopShape`	Desired frequency profile, tolerance (between min/max gains); desired cutoff frequency (assuming a desired integral frequency profile), tolerance	Like `MinLoopGain`, but combined with `MaxLoopGain` (with a tolerance between the 2)	$\mathcal{H}_\infty$ norm	Loop

APPENDIX B APPLICABILITY TESTS AND MITIGATIONS FOR INDI

This appendix aims at providing a quick reference for practitionners who intend to use the applicability test and mitigation techniques for INDI presented in this thesis. As such, all information contain in this appendix is simply a summary of what has been presented in Chapter 3, which should be referred to for further details along with Chapter 4 as it provides an application example.

Technical background presents the definition of relevant expressions, before the applicability tests are presented in detail. Proposed mitigation techniques are summarized at the end of this appendix.

B.1 Technical background

Considering the following nonlinear system:

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

We define the following expressions:

- $\mathbf{u}$ is the plant input vector (actuator output);
- $\mathbf{x}$ is the plant state vector;
- $\mathbf{u}_c$ is the commanded actuator position (actuator input);
- t_0 is the time of the last available measurement;
- $\tau = t - t_0$ is the (synchronized) delay since the last available measurement (must match the largest of the sensor delays or the sampling period);
- $\mathbf{H}_{act}$ is the actuator multivariate transfer function
- $\mathbf{K}(s) = (\mathbf{I} - \mathbf{H}_{act}(s))^{-1} s\mathbf{H}_{act}(s) \Leftrightarrow \mathbf{H}_{act}(s) = (s\mathbf{I} + \mathbf{K}(s))^{-1}\mathbf{K}(s)$ is the actuator "bandwidth function" (constant for first-order actuators);
- $\mathbf{H}_{sm}$ is the selector matrix that picks state derivative measurement signals from the plant output;
- $\mathbf{A}_0 = \mathbf{A}(\mathbf{x}_0, \mathbf{u}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{t=t_0}$ is the state Jacobian matrix of the plant;

- $\mathbf{G}_0 = \mathbf{G}(\mathbf{x}_0, \mathbf{u}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{t=t_0}$ is the input Jacobian matrix of the plant;
- $\tilde{\mathbf{G}}_0 = \mathbf{H}_{sm}\hat{\mathbf{G}}_0$ is the Control Effectiveness Matrix, which is a square matrix obtained from the selector matrix and the estimation of the input Jacobian matrix;
- $\bar{\mathbf{G}}_0^\dagger = \Big(\mathbf{H}_{sm}\hat{\mathbf{G}}_0\mathbf{K}(0)\Big)^\dagger\mathbf{K}(0)$ is the proposed modified Control Effectiveness Matrix inverse for disparate actuators;
- $\mathbf{F} = e^{s\tau}$ is the pure time delay transfer function (to be synchronized);
- T_s is the controller sampling time;
- $D = \frac{\tau_D}{T_s} = \left\lceil \frac{\tau}{T_s} \right\rceil$ is the discretized (synchronized) delay ($\tau_D = DT_s \geq \tau$);
- $\mathbf{P} = \mathbf{H}_{sm}(s\mathbf{I} - \mathbf{A}_0)^{-1}\mathbf{G}_0$ is the plant linear model's transfer function between $\mathbf{u}$ and $\mathbf{x}$ (or their derivatives $\dot{\mathbf{u}}$ and $\dot{\mathbf{x}}$);
- $\hat{\mathbf{P}} = \mathbf{H}_{sm}\frac{1}{s}\hat{\mathbf{G}}_0$ is the on-board simplified approximation of the plant linear model's transfer function, which neglects its dynamics;
- $\mathbf{M}_\star = e^{\mathbf{A}_\star T_s}$ is the state transition matrix ($\mathbf{\Phi}$) over one sample;
- $\mathbf{y} = \dot{\mathbf{x}}$ is the state derivative variable;
- $\zeta = \mathbf{H}_{sm}\mathbf{y}$ is the state derivative variable meant to track pseudo-command.

Figure B.1 presents the continuous feedback-linearization stage of the classic implementation of the INDI Control Law, defined as

$$\mathbf{u}_c = \mathbf{u}_0 + \tilde{\mathbf{G}}_0^\dagger(\nu - \zeta_0)$$

and whose closed-loop transfer function can be framed as

$$\frac{\zeta(s)}{\nu(s)} = \mathbf{H}_{sm}(\mathbf{I} + \mathbf{\Delta}_A)\Big\{\Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm}\Big)^{-1}\mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\Big\}(\mathbf{I} + \mathbf{\Delta}_D)$$

where

- $\mathbf{\Delta}_A = \Big(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} - \mathbf{A}_0\Big)^{-1}\mathbf{A}_0$ is the plant dynamics uncertainty;
- $\mathbf{\Delta}_D = \mathbf{E}_I\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\Big(\mathbf{I} + (\mathbf{P} + \mathbf{E}_I)\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\Big)^{-1}$ is the implementation uncertainty;
- $\mathbf{E}_I = (\mathbf{I} - \hat{\mathbf{F}})\hat{\mathbf{P}} - (\mathbf{I} - \mathbf{F})\mathbf{P}$ is the implementation error.

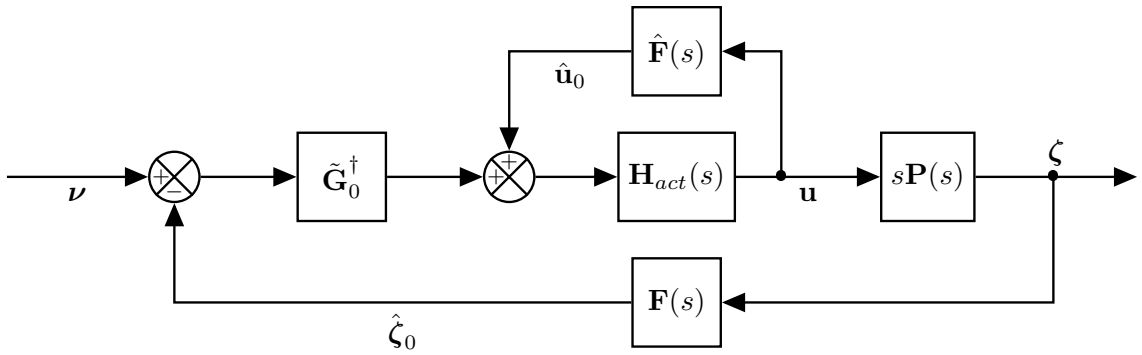

Figure B.1 Classic INDI implementation (continuous feedback-linearization shown)

Note that this closed-loop system represents the Feedback-Linearization stage of INDI, and that a linear controller must be added as an outerloop to complete the control scheme (referred to as the Linear Controller stage).

B.2 Applicability tests

Table B.1 summarizes the applicability tests and their associated proposed mitigation in case of failure. The type of norm to be used for each test is not specified as it should be selected based on the application at hand. For aerospace applications, it is recommended to employ weighting filters and select the $\mathcal{H}$ norm for all tests involving norms; see Appendix A for further details on norm types and weighting filters, or Chapter 4 for examples.

The following subsections will detail the applicability tests, with mitigations being detailed in the next section.

Table B.1 Tests for INDI applicability and proposed mitigation

Test	Expression	Mitigation
Decoupling (Identical actuators)	$\left\|\tilde{\mathbf{G}}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger - \mathbf{K}\right\| < \epsilon_G$	Modified INDI
Stabilizing	$\left\|\left(s\mathbf{I} + \mathbf{G}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} - \mathbf{A}_0\right)^{-1}\right\| < 1$	Nonlinear
Fast Actuation	$\|\mathbf{\Delta}_A\| < \epsilon_A$	Augmented plant Modified INDI
Slow varying (Locally Lipschitz)	$\|\mathbf{A}(t) - \mathbf{A}_0\| \leq L_A\tau$ $\|\mathbf{G}(t) - \mathbf{G}_0\| \leq L_G\tau$	Nonlinear Anti-spillover filtering
Small delay	$\left\|e^{\mathbf{A}_0\tau} - \mathbf{I}\right\| < \epsilon_t$	Modified INDI Predictive filter
Bandwidth ratio	$\tau\mathbf{K}(0) \geq \epsilon_\Omega\mathbf{I}$	Simpler control law (Negligible delay)
Synchronized	$\left\|\mathbf{F} - \hat{\mathbf{F}}\right\| < \epsilon_F$	Online Synchronization
Compensated	$\|\mathbf{\Delta}_D\| < \epsilon_D$	Modified INDI Predictive filter
Intersample rippling	$\left\|e^{\mathbf{A}_0T_s} - \mathbf{I}\right\| < \epsilon_s$	Modified INDI Filter
Discrete delay	$\frac{\tau}{T_s} - D = 0$	New (larger) $\tau = DT_s$ (Must be reassessed)

B.2.1 Decoupling

This test enforces the classic assumption of identical actuators, which requires the following norm

$$\left\|\tilde{\mathbf{G}}_0\mathbf{K}\tilde{\mathbf{G}}_0^\dagger - \mathbf{K}\right\| < \epsilon_G$$

to be negligible ($\epsilon_G \approx 0$ over the appropriate frequency band). If fulfilled, the idealized closed-loop ($\mathbf{\Delta}_A = \mathbf{\Delta}_D = \mathbf{0}$) will simplify down to the (decoupled) actuator dynamics.

Rationale: Simplification of the actuator loop relies on this decoupling, which does not work with the modified Control Effectiveness Matrix $\bar{\mathbf{G}}_0^\dagger$. Leads to actuator-induced coupled dynamics that cannot be rejected at the Feedback-Linearization stage, limiting achievable performance for the linear controller tuning process.

Mitigation: Modified INDI is intended to minimize dynamic coupling due to disparate actuators.

B.2.2 Stabilizing

Neglecting delays and implementation errors ($\mathbf{\Delta}_D = \mathbf{0}$), the Feedback-Linearization stage must be able to stabilize the plant over the relevant frequency band. This can be evaluated through the norm of the denominator of the closed-loop transfer function:

$$\left\| \left(s\mathbf{I} + \mathbf{G}_0 \mathbf{K} \tilde{\mathbf{G}}_0^\dagger \mathbf{H}_{sm} - \mathbf{A}_0 \right)^{-1} \right\| < 1$$

We note that this expression is part of $\mathbf{\Delta}_A$, and normally acts as a (beneficial) low-pass filter. Failure to meet this test would result in the lineraization error being amplified instead of attenuated, further destabilizing the system in uncertain ways that cannot be compensated by outerloops.

Rationale: Slow actuators fail to bound the linearization error; advanced controller scheme required.

Mitigation: None. An advanced nonlinear control law should be considered instead of INDI, due to inappropriate hardware (actuators) for the considered plant.

B.2.3 Fast Actuation

The simplifying assumption of negligible plant dynamics requires sufficiently fast actuation to dominate the closed-loop response. The approximation error from neglected plant dynamics can be framed as an output multiplicative error ($\mathbf{\Delta}_A$), which should be negligible for the plant dynamics to be as well. The test takes the following form:

$$\|\mathbf{\Delta}_A\| < \epsilon_A \quad \text{or} \quad \left\| \left(s\mathbf{I} + \mathbf{G}_0 \mathbf{K} \tilde{\mathbf{G}}_0^\dagger \mathbf{H}_{sm} - \mathbf{A}_0 \right)^{-1} \mathbf{A}_0 \right\| < \epsilon_A$$

where the error vanishes ($\mathbf{\Delta}_A \rightarrow \mathbf{0}$) as the actuator bandwidth is arbitrarily increased ($\mathbf{K} \rightarrow \infty\mathbf{I}$), highlighting how the actuator quasi-instantaneous bandwidth is meant to dominate the response. On the other hand, failing this test means the actuators are too slow to dominate the plant dynamics and that both the plant dynamics and actuator dynamics (including their interaction) need to be taken into account during the linear controller tuning phase.

Rationale: Actuator dynamics must be taken into account if they are too slow to be neglected. Manifests as an output multiplicative error on the closed-loop transfer. Represents the neglected plant dynamics.

Mitigation: The augmented plant approach should be used to capture the convolution between $\mathbf{P}$ and $\mathbf{H}_{act}$, and a robust tuning approach should be used for the Linear Controller stage (such as the Structured $\mathcal{H}_\infty$ Synthesis).

B.2.4 Slow varying

The slow evolution of the plant dynamics is a core assumption for the INDI control law, which is equivalent as requiring it to be (locally) Lipschitz with a small enough Lipschitz constant. The test takes the form:

$$\|\mathbf{A}(t) - \mathbf{A}_0\| \leq L_A \tau \qquad \text{and} \qquad \|\mathbf{G}(t) - \mathbf{G}_0\| \leq L_G \tau$$

where $L_A \leq \left|\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t}\right|$ and $L_G \leq \left|\frac{\mathrm{d}\mathbf{G}}{\mathrm{d}t}\right|$, allowing for estimations based on knowledge of the plant dynamics. As such, a Lipschitz constant estimate provides an upper limit on the tolerable (synchronized) delay.

Rationale: Large Lipschitz constants (over the time interval $[t_0, t]$) prevents the rejection of linearization errors; advanced controller scheme required unless anti-spillover filtering can be used.

Mitigation: Anti-spillover filtering, similar to the ones used on flexible aircraft, are meant to shield a control system from stimulating unmodeled dynamics. Otherwise, advanced nonlinear control should be sought as an alternative to INDI.

B.2.5 Small delay

The simplifying assumption of sufficiently small delays is used to justify the neglecting of plant dynamics when considering the Feedback-Linearization stage's closed-loop response. This is done by approximating the Taylor series expansion of the state transition matrix $\mathbf{\Phi} = e^{\mathbf{A}_0\tau}$ as the identity matrix (implying that $\mathbf{A}_0\tau \approx \mathbf{0}$ for small enough τ). The test is thus:

$$\left\| e^{\mathbf{A}_0\tau} - \mathbf{I} \right\| < \epsilon_t$$

where ϵ_t is the tolerable threshold for various multiplicative uncertainties impacting multiples terms of the equations of motion. For significant delays, the plant dynamics must be properly modeled to allow the Linear Controller stage to be appropriately tuned.

Rationale: Plant dynamics must be taken into account if delay is too large to make it negligible OR delay must be made smaller by using predictive filters. Leads to model mismatch in the effective Smith Predictor loop that increases $\|\mathbf{\Delta}_D\|$ (and thus the difference between the classical INDI equations and the proposed alternative).

Mitigation: Predictive filters to reduce delay OR a robust tuning approach should be used for the Linear Controller stage (such as the Structured $\mathcal{H}_\infty$ Synthesis).

B.2.6 Bandwidth ratio

This test is not related to any simplifying assumption made for classic INDI, but captures the interaction between the actuator bandwidth and the synchronized delay (or sampling rate). The test is simply:

$$\tau\mathbf{K}(0) \geq \epsilon_\Omega \mathbf{I}$$

which contains no norm (no weighting filter required). $\epsilon_\Omega \in (\frac{1}{2}, \frac{1}{5}, \frac{1}{10})$ represents the desired tolerance for neglecting dynamics poles as being dominated, which is usually a factor of 10 or 5 or minimally 2 (according to Nyquist's theorem). Failing this test means that the actuator poles are dominating the effective bandwidth function, and that delay compensation is not necessary and that INDI control is not required for this application; the idealized delay-free system can be used or simpler alternatives can be used such as the traditional PID controller.

Rationale: Delay compensation becomes unnecessary when actuator bandwidth dominates the effective bandwidth (can set $\hat{\mathbf{F}} = \mathbf{I}$ without significant loss of performance). Based on

Nyquist's theorem ($\epsilon_\Omega \in (\frac{1}{2}, \frac{1}{5}, \frac{1}{10})$ depending on desired tolerance). Designer can either use the delay-free model directly or use a different (simpler) control law entirely.

Mitigation: Classic INDI is unnecessary; consider a simplified version of INDI (based on the delay-free model) or a simpler control law (like PID controllers).

B.2.7 Synchronized

Delay synchronization is central to the INDI control law principles, due to its nested Smith Predictor. As such, the actuator position signals must be artificially delayed ($\hat{\mathbf{F}}$) to correspond to the same t_0 as the plant's sensors ($\mathbf{F}$). The test is thus:

$$\left\|\mathbf{F} - \hat{\mathbf{F}}\right\| < \epsilon_F$$

which vanishes when synchronization is achieved ($\epsilon_F \approx 0$ over the relevant frequency band). Improper synchronization will significantly increase the implementation error ($\mathbf{E}_I$) and the implementation uncertainty ($\mathbf{\Delta}_D$). Note that any filtering of the sensors will introduce dephasing that must be taken into account (e.g. such filters should be included in $\mathbf{F}$ and reproduced in $\hat{\mathbf{F}}$).

Rationale: Unknown or varying delays requires an on-board synchronization module [57] to ensure proper compensation.

Mitigation: Online synchronization [57].

B.2.8 Compensated

All simplifying assumptions that leads to a simplification of the implementation of the INDI control law are captured by the implementation uncertainty $\mathbf{\Delta}_D$. This is represented by the following test:

$$\|\mathbf{\Delta}_D\| < \epsilon_D \quad \text{or} \quad \left\|\mathbf{E}_I\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\Big(\mathbf{I} + (\mathbf{P} + \mathbf{E}_I)\mathbf{K}\tilde{\mathbf{G}}_0^\dagger\Big)^{-1}\right\| < \epsilon_D$$

where $\mathbf{E}_I = (\mathbf{I} - \hat{\mathbf{F}})\hat{\mathbf{P}} - (\mathbf{I} - \mathbf{F})\mathbf{P}$ is the implementation error (due to $\hat{\mathbf{F}}$ and $\hat{\mathbf{P}}$), which drives the tracking error (i.e. $\mathbf{\Delta}_D \to \mathbf{0}$ when $\mathbf{E}_I \to \mathbf{0}$). Failure to pass this test would warrant the use of a non-simplified INDI control loop, which is referred to as "Modified INDI" in this thesis (where $\bar{\mathbf{G}}_0\dagger = \tilde{\mathbf{G}}_0^\dagger$ for identical actuators).

Rationale: Plant dynamics must be taken into account if delay compensation is inadequate OR delay must be made smaller by using predictive filters. Manifests as an input multiplicative error on the closed-loop transfer. Represents the mismatch between delays and plant models due to implementation.

Mitigation: Improve delays (see *Small Delay* and *Synchronized* tests) OR reduce implementation error $\mathbf{E}_I$ by improving $\hat{\mathbf{P}}$ within Modified INDI OR use an Augmented plant along with a robust tuning approach for the Linear Controller stage (such as the Structured $\mathcal{H}_\infty$ Synthesis).

B.2.9 Intersample rippling

A clear distinction must be made between the (synchronized) sensor delays (τ) and the sampling period of the control loop (T_s); increasing the sampling rate arbitrarily will not improve the synchronized delay, but a poor sampling frequency may actually drive the delay ($\tau = T_s$ if T_s is larger than all sensor delays). This may indicate that a sampling period matching the synchronized delay would be optimal, but unfortunately this gives rise to the intersample rippling effect. Similarly to the *Small delay* test, we consider:

$$\left\| e^{\mathbf{A}_0 T_s} - \mathbf{I} \right\| < \epsilon_s$$

where ϵ_s is smaller than ϵ_t. In situations where this test passes while the *Small delay* test fails, the additional summation term in the alternative equations of motion can be neglected (i.e. situation where $\mathbf{M} \approx \mathbf{I}$ but not $\mathbf{M}^D$), which itself represents the remainder of the transient responses of previous input commands that occurred since t_0.

Rationale: Low sampling rate may require mitigation to avoid undesired high-frequency oscillations (fatigue). Contributes to implementation errors due to model mismatch (similarly to $\mathbf{\Delta}_D$ for the continuous case).

Mitigation: Increase sampling rate OR add roll-off filtering OR reduce delays (see *Small Delay* test).

B.2.10 Discrete delay

Discretization should ideally be made in a manner that does not alter the closed-loop dynamics. Whenever the synchronized delay is not an integer multiple of the sampling period,

the effective delay after discretization is increased to the next sample (next integer multiple of the sampling period). As such, the following test:

$$\frac{\tau}{T_s} - D = 0$$

should be fulfilled as to avoid increasing the synchronized delay further than required by any sensor.

Rationale: Discretized delay is a multiple of the sampling time rounded up, effective delay increases if the multiple is not an integer or if the sampling period is larger than the measurement delay.

Mitigation: Change the sampling rate OR reduce delays (see *Small Delay* test) OR replace τ by $\tau_D = DT_s$ (and re-assess all delay-related applicability tests).

B.3 Mitigation

Several mitigation techniques were previously mentioned, such as the use of:

- a robust tuning method (see Appendix A or Chapter 4),
- an online synchronization module (see [57]),
- anti-spillover filters (see Chapter 5),
- predictive filters (e.g. Kalman Filters [110]),
- a modified INDI control law, and
- an augmented plant formulation.

These last two were introduced in this thesis (see Chapter 3, or alternatively [73]) and are now concisely presented for practical convenience.

B.3.1 Modified INDI

The modified INDI control law is presented in Fig. B.2 and is defined as

$$\mathbf{u}_c = \Big(\mathbf{I} - (\mathbf{I} - \tilde{\mathbf{G}}_0)\hat{\mathbf{H}}_{act}\Big)^{-1}\bar{\mathbf{G}}_0^{\dagger}\Big(\nu - \zeta_0 + \tilde{\mathbf{G}}_0\mathbf{u}_0\Big)$$

where the classic on-board model approximation $\hat{\mathbf{P}} = \tilde{\mathbf{G}}_0$ (as well as $\hat{\mathbf{F}} = \mathbf{F}$) was used. This alternative control law allows the use of $\bar{\mathbf{G}}_0^{\dagger}$, which mitigates the use of non-identical

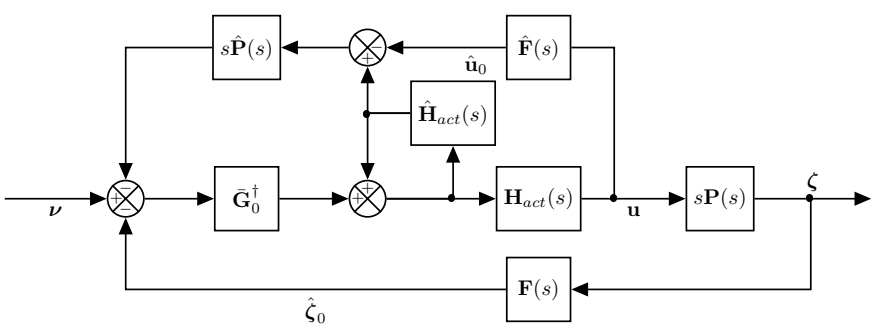

Figure B.2 Modified INDI implementation (continuous feedback-linearization shown)

actuators, in addition of allowing the modification of the on-board model $\hat{\mathbf{P}}$ in order to reduce the implementation error $\mathbf{E}_I$.

B.3.2 Augmented plant

This mitigation approach consists in including the actuator dynamics within the plant, when the transient described by the convolution of the plant and actuator dynamics is not negligible (usually due to insufficient actuator bandwidth). This is meant to provide accurate linear models to be used during the tuning of the Linear Controller stage; the control law and implementation are unchanged, only the way the closed-loop transfer function is expressed is changed to capture the transient dynamics emanating from the convolution between the actuator dynamics and the plant dynamics. As such, the equations of motion can be frame as follows:

$$
\begin{aligned}
\begin{bmatrix}\dot{\mathbf{y}}\\ \ddot{\mathbf{u}}\end{bmatrix} &= \begin{bmatrix}\mathbf{A}_0 & \mathbf{G}_0\\ \mathbf{0} & \mathbf{A}_{act}\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \dot{\mathbf{u}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{act}\end{bmatrix}\dot{\mathbf{u}}_c + \begin{bmatrix}\epsilon_L\\ \mathbf{0}\end{bmatrix}\\
\dot{\xi} &= \mathbf{A}_{aug}\xi + \mathbf{B}_{aug}\dot{\mathbf{u}}_c + \epsilon_{aug}\\
\Rightarrow \xi(t) &= e^{\mathbf{A}_{aug}(t-t_0)}\xi(t_0) + \mathbf{B}_{aug}\mathbf{u}_c(t) - e^{\mathbf{A}_{aug}(t-t_0-T_s)}\mathbf{B}_{aug}\mathbf{u}_c(t_0)\\
&\quad + \sum_{j=1}^{D-1} e^{\mathbf{A}_{aug}((D-j-1)T_s)}\Big(e^{\mathbf{A}_{aug}T_s} - \mathbf{I}\Big)\mathbf{B}_{aug}\mathbf{u}_c(t_0 + jT_s) + \mathbf{E}_{aug}(t)
\end{aligned}
$$

where

$$
e^{\mathbf{A}_{aug}x} = \begin{bmatrix} e^{\mathbf{A}_0 x} & \left(e^{\mathbf{A}_0 x}\right) \circledast \left(\mathbf{G}_0 e^{\mathbf{A}_{act}x}\right)\\ \mathbf{0} & e^{\mathbf{A}_{act}x}\end{bmatrix}
$$

The closed-loop then becomes:

$$
\begin{aligned}
\xi(t) &= \mathbf{M}_{aug}^D\xi(t_0) + \mathbf{B}_{aug}\mathbf{u}_c(t) - \mathbf{M}_{aug}^{D-1}\mathbf{B}_{aug}\mathbf{u}_c(t_0) + \sum_{k=1}^{D-1}\mathbf{M}_{aug}^{k-1}(\mathbf{M}_{aug} - \mathbf{I})\mathbf{B}_{aug}\mathbf{u}_c(t - kT_s)\\
&\quad + \mathbf{E}_{aug}(t)\\
\zeta(t) &= \begin{bmatrix}\mathbf{H}_{sm} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\xi(t)\\
\mathbf{u}_c(t) &= \tilde{\mathbf{G}}_0^\dagger\nu(t) - \begin{bmatrix}\tilde{\mathbf{G}}_0^\dagger & \mathbf{B}_{act}^{-1}\end{bmatrix}\hat{\zeta}(t_0) + \mathbf{u}_c(t_0)\\
&= \tilde{\mathbf{G}}_0^\dagger\nu(t) - \begin{bmatrix}\tilde{\mathbf{G}}_0^\dagger & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{y}}(t_0)\\ \hat{\mathbf{u}}(t_0)\end{bmatrix} = \hat{u}(t_0) + \tilde{\mathbf{G}}_0^\dagger(\nu(t) - \hat{\mathbf{y}}(t_0))\\
\Rightarrow \zeta(z) &= \begin{bmatrix}\mathbf{H}_{sm} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\Big(\mathbf{I} - \Big(\mathbf{M}_{aug}^D - \tilde{\mathbf{Q}}_{aug}(z)\begin{bmatrix}\tilde{\mathbf{G}}_0^\dagger\mathbf{H}_{sm} & \mathbf{B}_{act}^{-1}\end{bmatrix}\Big)z^{-D}\Big)^{-1}\Big\{\tilde{\mathbf{Q}}_{aug}(z)\tilde{\mathbf{G}}_0^\dagger\nu(z) + \mathbf{E}_{aug}(z)\Big\}
\end{aligned}
$$

where $\mathbf{M}_{aug} = e^{\mathbf{A}_{aug}T_s}$ and $\tilde{\mathbf{Q}}_{aug}(z) = (1 - z^{-1})\Big(\mathbf{I} - \mathbf{M}_{aug}^D z^{-D}\Big)(\mathbf{I} - \mathbf{M}_{aug}z^{-1})^{-1}\mathbf{B}_{aug}$.

The open and closed-loop transfer functions are essentially the same as with the classic case,

with the exception that every terms has been replaced by their augmented counterpart. As such, other mitigation techniques can be combined if necessary, such as the Modified INDI control law. All these expressions simplify down to the classical formulations when the actuator position reaches steady-state within the sampling period, a condition which renders the actuator dynamics negligible.

In practice, it is recommended to use the `feedback` command in MATLAB to generate the (continuous time) Feedback-Linearization stage's closed-loop transfer function:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix} = \texttt{FEEDBACK}\left\{ \begin{bmatrix} \mathbf{P}s\mathbf{H}_{act} \\ \mathbf{H}_{act} \end{bmatrix}, \begin{bmatrix} \tilde{\mathbf{G}}_0^{\dagger} & -\mathbf{I} \end{bmatrix} \right\} \tilde{\mathbf{G}}_0^{\dagger}\nu$$

to generate the linear models to be used during the Linear Controller tuning.