Politecnico di Milano
Department of Aerospace Science and Technology
Doctoral Programme in Aerospace Engineering

Learning-Based Control For Aerospace Systems: Methods and applications

Doctoral Dissertation of:
Salvatore Meraglia

Supervisor:
Prof. Marco Lovera

Tutor:
Prof. Pierluigi Di Lizia

The Chair of the Doctoral Program:
Prof. Pierangelo Masarati

2022 – XXXV

Salvatore Meraglia
Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano
Via Giuseppe La Masa, 34
20156 Milano, Italy
E-mail: `salvatore.meraglia@polimi.it`

Abstract

Nowadays, researchers strive to design systems that can operate autonomously in safety-critical applications, such as, *e.g.*, unmanned aerial vehicles and space systems. To ensure a predictable system response and safe operation, reliable control of these systems needs not only to meet performance specifications under nominal conditions, but also to accommodate graceful performance degradation when underlying assumptions are violated. This work mainly aims to design control algorithms for aerospace systems that integrate the experience gathered (*data-driven* knowledge) into a classical (*model-based*) control framework in a systematic way. The thesis is structured so that the problem statements and proposed control solutions are addressed from a methodological and general point of view. Indeed, these solutions are specialized in the light of the particular case study only at the end. Firstly, novel adaptive control architectures with stability, performance, and robustness guarantees are proposed. Then, harmonic control algorithms for disturbance attenuation are presented, focusing on aerospace applications that experience periodic disturbances with a known source and period, but whose amplitude is uncertain. Finally, the dissertation covers *learning-based* methods that do not assume any periodicity of the disturbance, but rely on the repetition of a particular task to improve performance from one trial to the next.

Contents

List of Figures

List of Tables

Introduction

In recent years, researchers have strived to design systems that can operate autonomously in safety-critical applications. Examples include Unmanned Aerial Vehicles (UAVs) [1], self-driving cars [2], and space systems [3]. To ensure a predictable system response and safe operation, reliable control of these systems needs not only to meet performance specifications under nominal conditions, but also to accommodate ”graceful performance degradation” when underlying assumptions are violated [4].

State of the art

Control theory has traditionally involved a *model-based* approach: it leverages a given model of the system dynamics and provides guarantees concerning known operating conditions. Typically, model simplifications of the dynamical system are made during the controller design. This may include making idealized assumptions, linearising a highly nonlinear dynamical system, and neglecting external disturbances and/or unmodelled dynamics. Indeed, a highly complex model could lead to controllers that are too difficult or costly to design, use and maintain. Model identification can be adopted to obtain the system model leveraging measured data from experimental tests on the actual system. However, even if the most advanced identification method is employed, the model always approximates the actual system, *i.e.,* some errors are inevitable. In addition, several unpredictable conditions could introduce uncertainty in a real system, *e.g.*, an aircraft suffering structural damage or a spacecraft grasping an unknown object. As a result, system uncertainties unavoidably exist between the control model and the actual dynamical system. Robust control theory was born to deal with these problems, assuming bounds on noise and/or model uncertainties [5]. Robust control design techniques, such as robust H_∞ and H_2 control design, yield suitable controllers for the modelled disturbances/uncertainties and keep the controller unchanged after the initial design. However, being designed to operate under the worst-case condition, a robust controller may use excessive actions to regulate the process and fail to achieve satisfactory performance over the entire range in which system characteristics may vary. To overcome all these limitations, *learning-based* (or *data-driven*) control approaches represent a viable solution. In recent years, these approaches have also gained popularity due to the success of machine learning in other fields, such as computer vision [6] and natural language processing [7]. *Learning-based* algorithms (almost) completely omit the modelling part and can quickly adapt to new situations by leveraging the experience collected during the operation. However, these methods typically do not provide stability or robustness guarantees. As a consequence, despite a large number of ongoing research projects, the application of *learning-based* methods in safety-critical applications is very limited. In conclusion, the ”best” control solution should involve not only *model-* or *learning-based* methods, but a combination of both approaches that works best in maintaining closed-loop stability, enforcing robustness to uncertainties, and delivering the desired performance in the presence of unexpected events [8].

Approach

This dissertation has the main purpose of designing control algorithms for aerospace systems that integrate in a systematic way the experience gathered (*data-driven* knowledge) into a classical (*model-based*) control framework to have safety guarantees. The control design problem is tackled in three different domains: *time-*,*frequency-* and *trial-domain*.

- *Time-domain*: Some innovative adaptive control architectures are proposed. These controllers are tuned to the physical system in real-time, not the worst-case scenario. Thus, adaptive controllers have the natural capability to estimate and suppress the effect of system uncertainties, without sacrificing performance [9]. This advantage is obtained learning during the operation, *i.e.*, measurements are leveraged to adjust online the control parameters.

- *Frequency-domain*: Several modifications of the harmonic control algorithms are presented considering the disturbance acting on the system being tonal or multi-tonal with a known spectrum. Since the harmonic controller is designed to deal only with the harmonics of output and control input components, it is considered to actually operate in the *frequency-domain* [10].

- *Trial-domain*: Iterative Learning Control (ILC) methods are considered that take advantage of the repetition of a specific task to improve performance from one trial to the next [11]. The *learning* phase occurs offline, *i.e.*, data collected from each experiment/trial are recorded and used in a batch fashion between closed-loop operation trials. The proposed algorithms are computationally efficient, allowing a fast re-tuning of the controller when the system performance is reduced (*e.g.*, components degradation) or when the operating conditions change (*e.g.*, different payloads). Furthermore, the proposed methods guarantee safe operation not only for the optimized controller, but also during the *learning* phase to avoid costly hardware failures.

Structure and contributions

The dissertation is structured into three main Parts and ten Chapters. Part I is dedicated to presenting novel adaptive control architectures with stability, performance, and robustness guarantees, with the goal of developing a methodological approach to design adaptive control systems in various aerospace applications. Part II is instead devoted to presenting and discussing harmonic control algorithms for disturbance attenuation to overcome some of the issues and limitations of classical control laws. This study area is relevant to many aerospace applications that experience periodic disturbances with a known source and period, but whose amplitude is uncertain. On the other hand, Part III describes ILC methods that do not assume any periodicity of the disturbance or reference signals, but rely on the repetition of a particular task to improve performance from one iteration to the next. The thesis is structured so that the problem statements and proposed control solutions are addressed from a methodological and general point of view. Indeed, these solutions are specialized in the light of the particular case study only at the end. As for each Chapter, the contributions are summarized in the following.

- Chapter 1 presents a historical overview of adaptive control while describing the basic concepts and features of adaptive algorithms in view of the methodological approaches discussed in the following.

- The design and stability analysis of an adaptive position controller for UAVs are presented in Chapter 2. The novel aspect of this work is the definition of a systematic design process for a position controller based on Model Reference Adaptive Control, taking into account not-fast closed-loop attitude dynamics in a hierarchical control scheme. After having reformulated the problem considering the attitude dynamics as a pseudo-actuator, an existing Linear Matrix Inequality-based hedging framework has been designed such that the presence of actuator dynamics has no effects on the performance of the adaptation. Simulation and experimental results are provided to illustrate the performance of the proposed control scheme.

- In the previous Chapters, different approaches to designing an adaptive controller with constant unknown parameters have been presented. Unfortunately, these controllers cannot handle time-varying dynamics that is common for aerospace systems to exhibit. Chapter 3 addresses this issue by developing a continuous adaptive controller for nonlinear dynamical systems with linearly parametrizable uncertainty involving time-varying uncertain parameters. Specifically, a modification of *congelation of variable* method is designed to achieve trajectory tracking of a scalar nonlinear system.

- Chapter 4 focuses on the attitude control problem of a spacecraft affected by time-varying inertia variation. Although adaptive control schemes for spacecraft attitude control are numerous in the literature, few have been designed to ensure consistent performance for a spacecraft with both rigid and non-rigid (time-varying) inertia components. In view of this, novel adaptive attitude control schemes are proposed that take advantage of the approach presented in Chapter 3. Detailed derivations of the control laws are provided, along with a thorough analysis of the associated stability and error convergence properties. In addition, numerical simulations are presented to highlight the performance benefits compared with classical control schemes that do not account for inertia time variations[1].

- Chapter 5 gives a brief overview of the harmonic control algorithms starting from the definition of the T-matrix for linear systems which can be used as basis for the control methodologies developed in the following Chapters.

- Chapter 6 presents the design of an active balancing system for rotating orbital devices, motivated by recent space applications for spacecraft endowed with rotating payloads. After having formulated the problem as compensating for a periodic disturbance of a known frequency, a harmonic control algorithm has been designed to command the positions of the actuated masses so that the effects of rotor unbalance are significantly reduced. After extensive numerical simulations, a dedicated breadboard has been developed, and experimental validation of the control law has been carried out.

- Chapter 7 describes a second application of harmonic control, focused on a slightly different paradigm. In particular, the harmonic control framework is specialized in the helicopter rotor-induced vibration problem. Several techniques have been developed to solve this problem, broadly divided into passive and active vibration control techniques. Despite the large amount of research produced in

[1] This work, along with the theory developed in Chapter 3, has been carried out by the Author under the supervision of Prof. Maruthi Akella at the *University of Texas* in Austin (Texas), USA.

the last decades, the interaction between active and passive vibration control techniques has not been analysed yet. The main contribution of Chapter 7 is the analysis of the interaction between the active control of structural response and a mast vibration absorber. Specifically, it is shown that an *ad hoc* design of the control algorithm is needed to avoid instabilities and/or degraded performance caused by the interaction between the two devices.

- The fundamental ideas and aspects regarding ILC algorithms are discussed in Chapter 8, focusing on the *optimization-based* framework used in the following Chapters.

- Chapter 9 describes the novel *Smoother-based* ILC algorithm, highlighting common points and differences concerning the state of the art described in the previous Chapter. The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance to improve learning performance. Furthermore, the algorithm has been specialized to achieve high-performance trajectory tracking with UAVs, and an experimental campaign involving a small quadrotor has been carried out.

- The ILC approach presented in the previous Chapter has been capable of reaching exceptional tracking performance. However, the learning phase needed to apply such a technique is related to each specific system, thus making the application of ILC poorly scalable. To overcome this limitation, Chapter 10 presents a novel H_∞-optimization-based definition of a transfer map that allows transforming the input signal learnt on a source system to the input signal needed for a target system to execute the same task. Experimental results demonstrate the effectiveness of the proposed approach in improving the tracking performance of a UAV transferring knowledge between different-scale UAVs.

Research works and M.Sc. theses

Part of the material presented in this dissertation has been published or submitted for review in the following works[2].

Conference papers:

1. S. Meraglia, D. Invernizzi, M. Lovera. *Active balancing systems for rotating orbital devices*. XXV AIDAA International Congress, Roma, Italy, 2019;

2. R. Rubinacci, S. Meraglia, M. Lovera, *Continuous time adaptive Higher Harmonic Control*, 6th CEAS Conference on Guidance, Navigation and Control, Berlin, Germany, 2022;

3. S. Meraglia, M. Lovera, *Smoother-based Iterative Learning Control for UAV Trajectory Tracking*, 2022 American Control Conference, Atlanta (Georgia), USA, 2022;

4. V. Donatone, S. Meraglia, M. Lovera, *H-infinity-based Transfer Learning for UAV Trajectory Tracking*, 2022 International Conference on Unmanned Aircraft Systems, Dubrovnik, Croatia, 2022;

5. G. Roggi, S. Meraglia, M. Lovera. *Leonardo Drone Contest 2021: Politecnico di Milano team architecture*, 2022 International Conference on Unmanned Aircraft Systems, Dubrovnik, Croatia, 2022;

6. S. Meraglia, M. Lovera. *The role of closed-loop attitude dynamics in adaptive UAV position control*. 22nd IFAC Symposium on Automatic Control in Aerospace, Mumbai, India, 2022.

Journal papers:

1. S. Meraglia, M. Lovera, *Smoother-based Iterative Learning Control for UAV Trajectory Tracking*, IEEE Control Systems Letters, Vol. 6, pp. 1501-1506, 2022;

2. G. Roggi, S. Meraglia, M. Lovera, *Leonardo Drone Contest autonomous drone competition: overview, results, and lessons learned*

[2] Other works are under preparation for submission.

from Politecnico di Milano team, Journal of Intelligent & Robotic Systems [Under review];

3. R. Rubinacci, S. Meraglia, M. Lovera, *Integrated Active and Passive Rotorcraft Vibration Control*, Journal of the American Helicopter Society [Under review];

4. S. Meraglia, D. Invernizzi, M. Lovera, T. Mohtar, A. Bursi, *Design of an active balancing system for rotating orbital devices*, Journal of Guidance, Control, and Dynamics [Under review].

Moreover, the Author has been involved in a number of *M.Sc.* theses as co-advisor. Much of the work presented in this dissertation has been obtained as the result of the joint effort between the Author and the master students, and has been the basis for further developments. The Author's contributions are clarified in the following.

1. R. Rubinacci. *Integrated active and passive rotorcraft vibration control*. Aeronautical Engineering, *M.Sc.*, 2020;

2. L. Bendinelli. *Model reference adaptive control strategies for tiltable propellers UAVs*. Space Engineering, *M.Sc.*, 2020;

3. F. Carloni. *A norm-optimal Kalman ILC for precise UAV trajectory tracking*. Aeronautical Engineering, *M.Sc.*, 2020;

4. M. Padovani. *A deep reinforcement learning-based algorithm for exploration planning*. Aeronautical Engineering, *M.Sc.*, 2021;

5. A. Della Libera. *Adaptive control for position-attitude cascade controller for UAVs*. Aeronautical Engineering, *M.Sc.*, 2021;

6. V. Donatone. *H-infinity-based transfer learning for UAV trajectory tracking*. Aeronautical Engineering, *M.Sc.*, 2021;

7. G. Sala. *Learning-based control for UAV trajectory tracking*. Aeronautical Engineering, *M.Sc.*, 2021;

8. G. Lizza. *Dynamic scaling for model identification of multirotor UAVs*. Aeronautical Engineering, *M.Sc.*, 2021;

9. M. Pepellin. *$\mathcal{L}_1$ adaptive control for tiltable propellers UAVs*. Aeronautical Engineering, *M.Sc.*, 2022;

10. M. Labori. *Adaptive control strategies for high-performance attitude control*. Aeronautical Engineering, *M.Sc.*, 2022;

11. E. Colli. *Fault detection and isolation using machine learning for spacecraft attitude sensors*. Space Engineering, *M.Sc.*, 2022;

12. L. Shang. *Adaptive position control for multirotor UAVs*. Aeronautical Engineering, *M.Sc.*, 2022;

Notation

This is a brief collection of the most used notations throughout the dissertation. $\mathbb{R}\,(\mathbb{R}_{>0})$ denotes the set of (positive) real numbers, $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space, and $\mathbb{R}^{m\times n}$ the set of $m \times n$ real matrices. Given $A \in \mathbb{R}^{n\times n}$, we use the compact notation $A \in \mathbb{R}^{n\times n}_{>0}$ to represent a positive definite matrix. The i-th vector of the canonical basis in $\mathbb{R}^n$, *i.e.*, the vector with a 1 in the i-th coordinate and 0 elsewhere, is denoted as e_i and the identity matrix in $\mathbb{R}^{n\times n}$ is denoted as $I_n := [e_1 \cdots e_i \cdots e_n]$. Given vectors $x, y \in \mathbb{R}^n$, the standard inner product is defined as $\langle x,y\rangle := x^\top y$. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is $\|x\| := \sqrt{\langle x,x\rangle}$. For A and $B \in \mathbb{R}^{n\times n}$, the standard inner product is given by $\langle A,B\rangle := \operatorname{tr}\left(A^T B\right) = \operatorname{vec}(A)^T \operatorname{vec}(B)$, where $\operatorname{vec}(A)$ is the column obtained by staking the columns of A one after the other. The norm induced by this inner product is the Frobenius norm, which is defined as $\|A\|_F := \sqrt{\operatorname{tr}(A^T A)}$. The minimum and maximum eigenvalues are denoted as $\lambda_m(A)$ and $\lambda_M(A)$, respectively, and skew $(A) := \frac{A-A^T}{2}$ is the skew-symmetric part of A. The n-dimensional unit sphere is denoted as $\mathbb{S}^n := \left\{x \in \mathbb{R}^{n+1} : \|x\| = 1\right\}$. The set $\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} : R^\top R = I_3, \det(R) = 1\}$ denotes the three-dimensional Special Orthogonal group. Given $\omega \in \mathbb{R}^3$, the map $S(\cdot) : \mathbb{R}^3 \to \mathfrak{so}(3) := \{\Omega \in \mathbb{R}^{3\times 3} : \Omega = -\Omega^\top\}$ is such that $S(\omega)y = \omega \times y$, $\forall y \in \mathbb{R}^3$, where $\times$ represents the cross product in $\mathbb{R}^3$. The inverse of the map S is denoted as $S(\cdot)^{-1} : \mathfrak{so}(3) \to \mathbb{R}^3$.

Part I

Adaptive Control

CHAPTER 1

Adaptive control preliminaries

Adaptive control has been the subject of extensive research over the last decades. As a popular control methodology of increasing interest for applications in engineering and science fields, adaptive control has unique capabilities to accommodate system parametric, structural, and environmental uncertainties caused by, *e.g.*, payload variations, component failures, and external disturbances. Indeed, adaptive control, under some generic design conditions, can tolerate significant uncertainties to ensure desired system asymptotic tracking performance and system stability. These characteristics are crucial for resilient control systems whose performance is required to be recoverable under system uncertainties and faults, such as aircraft control. This Chapter provides a historical overview while describing the basic concepts and features of adaptive algorithms.

1.1 Introduction

The control designer rarely has a precise knowledge of the plant parameters when controlling a real plant. Additionally, the environment in which the system operates may undergo unanticipated changes, and the characteristics of the plant may alter over time as a result of numerous factors. It may be difficult to achieve satisfactory performance over the full range in which system characteristics may vary using conventional control theory tools with fixed parameters. A sophisticated feedback controller should be able to learn changes in parameters through processing and use appropriate gains to adapt them because the output response contains information about the state and parameters of the system [12]. This debate produced a feedback control structure that serves as the foundation for adaptive control. The controller structure consists of a feedback loop and a controller with real-time adjustable gains, as shown in Figure 1.1. The way controller gains are updated in response to changes in system dynamics and disturbances distinguishes one adaptive scheme from another [9].

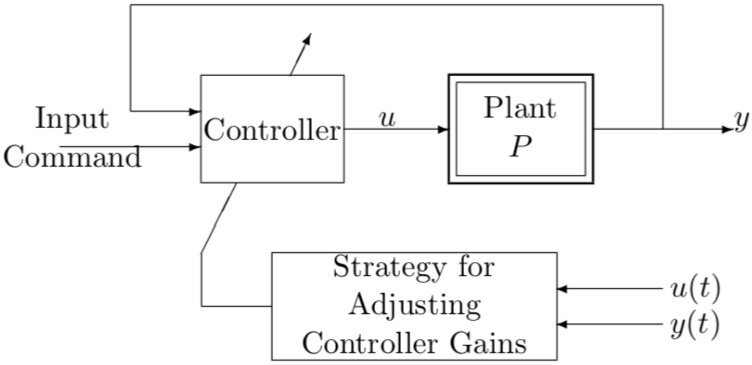


Figure 1.1: *General structure of an adaptive controller [9].*

The definition of an *adaptive control system* continues to be multifaceted, making it impossible to define them in a single and concise sentence [9]. The term *adaptation* is defined in biology as “an advantageous conformation of an organism to changes in its environment.” Inspired by this definition, Drenick and Shahbender introduced the term *adaptive system* in control theory to represent "control systems that monitor their own performance and adjust their parameters in the direction of better performance" [13]. Another popular definition of an *adaptive control system* has been given in [14], which describes it as "a feedback control system intelligent enough to adjust its characteristics in a changing environment to operate in an optimum manner according to some specified criterion."

Remark 1. *In addition to their adaptive properties, adaptive controllers can be designed to learn [15]. Learning is the process of identifying patterns and taking action based on memory. A simple example of a learning controller is a tracking error integrator in the feedback loop. It accumulates and integrates errors based on previous and current data. Adaptive controllers can be seen as nonlinear extensions of feedback integrators [16]. Compared to a standard learning process that is primarily concerned with maximizing parameters knowledge, adaptive control is a type of learning that minimizes the error with respect to a specific target output. Although the objective may differ, standard learning algorithms and adaptive control share common research interests [17].*

1.2 Historical background

Since the idea of adaptation and feedback are so closely related, the history of adaptive control systems is almost as long as the entire field of control systems. As a result, adaptive control has been investigated since the *1950s* and is still an area of intense research activity. As in [17], we categorize various developments in this area into three periods (*1950-1965*, *1965-1980*, and *1980-2000*) that are outlined below.

1.2.1 1950-1965: Early research

Since the early *1950s*, the design of autopilots for high-performance aircraft has motivated intensive research on adaptive control [18, 19]. Indeed, constant-gain feedback control methods cannot handle effectively when the aircraft flies from one operational point to another and their dynamics changes dramatically. An advanced controller, such as an adaptive controller, that can learn and adapt to changes in aircraft dynamics was needed. Whitaker *et al.* in [20] proposed the Model Reference Adaptive Control (MRAC) to solve the autopilot control problem. The design of adaptive laws for the various proposed control schemes involved the use of the sensitivity method [21] and the MIT rule [22]. Kalman proposed in [23] an adaptive pole placement scheme based on an optimal linear quadratic problem. This period is called the *brave era* because "there was a very short path from idea to flight test with very little analysis in between" [24]. The tragic flight test of the X-15 confirms this view [25].

1.2.2 1965-1980: Stability framework

It was soon realized that the MIT rule could lead to instability, especially when there is a sufficient phase lag between the error measurement and the parameters update [26]. Several authors contributed to the formulation of a stability framework for analysing and synthesizing adaptive systems [27, 28, 29]. Instead of using a gradient descent approach as in [22], Lyapunov's method was proposed, ultimately becoming the basis for the stability of adaptive systems. The work done over these 15 years set the foundations for stable adaptation in dynamic systems, both stochastic and deterministic, focusing mainly on parametric uncertainties. The overall goal was to ensure a well-behaved closed-loop system that asymptotically met control goals such as tracking and regulation. Furthermore, conditions under which learning, *i.e.*, accurate parameter estimation, can take place were formalized, deriving both necessary and sufficient conditions [30]. Finally, the concurrent advancement of computers and electronics, which has made possible the implementation of complex controllers, such as adaptive ones, increased interest in adaptive control applications [9].

1.2.3 1980-2000: Robust adaptive control

The debates concerning the practicality of adaptive control immediately followed the accomplishments of the *1970s*. It was noted in [31] that the classical adaptive strategies could become unstable in the presence of small disturbances. Other examples of instability were later published, demonstrating a lack of robustness in the presence of unmodelled dynamics or bounded disturbances [32]. Several approaches have been developed to ensure that adaptive control systems provide adequate robustness to withstand non-parametric uncertainties, such as external disturbances, time-varying parameters, and unmodelled dynamics [33, 34, 35]. As a result, a body of research called *robust adaptive control* was produced. In particular, an adaptive controller is defined as *robust* if it guarantees signal boundedness in the presence of "reasonable" classes of unmodelled dynamics and bounded disturbances [9].

1.3 Adaptive control problem formulation

We consider the dynamics of a plant process $P(\theta)$ given by

$$\dot{x} = f(x, \theta, u, t) \tag{1.1}$$

$$y = g(x, \theta, u, t) \tag{1.2}$$

where $x(t) \in \mathbb{R}^n$ represents the system state, $y(t) \in \mathbb{R}^p$ represents all measurable system outputs. $\theta \in \mathbb{R}^s$ represents the system parameters, and $f(\cdot)$ and $g(\cdot)$ describe the system dynamics. The plant $P(\theta)$ is subject to various perturbations and modelling errors due to environmental changes, complexities in the underlying mechanisms, and anomalies. Hence both f and g are not fully known. Most adaptive control schemes have adopted a parametric approach to distinguish known from unknown parts. Specifically, it is assumed that f and g are known functions parametrized with respect to the parameter θ, which is unknown.

Remark 2. *It is an idealization to assume that the only unknown in equations* (1.1)-(1.2) *is the parameter θ and that the functions, f and g, are exactly known. Several departures from this assumption can take place in the form of unmodelled dynamics, time-varying parameters, and disturbances. In these cases, robust modifications need to be designed to ensure that the underlying signals remain bounded, with errors that are proportional to the size of these perturbations. These modifications either rely on properties of persistent excitation (PE) of the exogenous command signals [36] or on appropriately modifying the adaptive law [9].*

1.3.1 Adaptive control objective

The adaptive control objective is to design the input signal $u(t)$ so that the system output $y(t)$ tracks a desired command signal $y_c(t)$ by including an adaptive component that attempts to estimate the parameters online[1]. Typically, the control objective is to ensure that $\lim_{t\to\infty} e(t) = 0$, where $e(t) = y(t) - y_c(t)$. Furthermore, the control input must lead to a closed-loop dynamic system that has bounded solutions at all time t [12].

1.3.2 Adaptive control solutions

Most of the existing adaptive control solutions are built upon the conventional certainty equivalence (CE) principle [9]. Precisely, CE-based solutions determine a controller structure that leads to an optimal solution when the parameters are known, and then replace the parameters in the controller with their estimates. A typical solution of the adaptive controller takes the form

$$u = F_u\left(\theta_c, \phi, t\right) \tag{1.3}$$

$$\theta_c = F_\theta\left(\theta_c, \phi, t\right) \ , \tag{1.4}$$

where $\theta_c(t)$ is an estimate of a control parameter that is intentionally varied as a function of time, and $\phi(t)$ represents all available data at time t. The non-autonomous nature of F_u and F_θ is due to the presence of exogenous signals, *e.g.,* set-points and command signals. The functions F_u and F_θ are deterministic constructions, and make the overall closed-loop system nonlinear and non-autonomous. The adaptive control solutions involve suitable functions F_u and F_θ to learn the unknown parameter $\theta_c^*(t)$, and achieve the control objective described in the previous Section.

Remark 3. *Since adaptive controllers synthesized through the CE principle are designed to guarantee the elimination of tracking error rather than accurate parameter estimation, their performance can generally only at most equal the performance of corresponding controllers for deterministic cases. Moreover, unless the reference signals satisfy specific PE conditions, CE-based adaptive controllers cannot guarantee convergence of parameter estimation errors to zero [9]. These facts lead to potential performance degradation of CE-based adaptive controllers in many applications compared to the underlying deterministic controllers [38, 39]. Therefore, research efforts in the field of adaptive control have begun to*

[1] For details, the readers are referred to the survey papers [17, 37] and references therein.

explore deviations from the CE philosophy. The Immersion and Invariance (I&I) formulation is a notable example of such efforts [40].

1.3.3 Adaptive control architecture

An adaptive controller includes an online parameter estimator (1.4), also known as *adaptation law*, and a control law (1.3). In this Section, we describe the different approaches classified according to how the parameter estimator is combined with the control law [9].

Indirect adaptive control

The first approach, known as *indirect adaptive control*, estimates the system parameters online and modifies the control parameters in accordance with these estimates. This architecture has also been referred to as *explicit identification* in the literature [12]. In particular, the system model $P(\theta^*)$ is parametrized with respect to an unknown parameter vector θ^* and, by analysing the system input $u(t)$ and output $y(t)$, an online parameter estimator provides an estimate $\theta(t)$ of $\theta^*(t)$ at each time instant. Then, the controller parameters are computed by solving an algebraic equation $\theta_c(t) = F(\theta(t))$ using the parameter vector estimate $\theta(t)$. The block diagram of the *indirect adaptive control* scheme is shown in Figure 1.2.

Direct adaptive control

On the other hand, in the *direct adaptive control* architecture, no effort is made to identify the system parameters. As a result, control parameters are adjusted directly to improve a certain performance index. This architecture is also referred to as *implicit identification* [12]. Specifically, the system model $P(\theta^*)$ is parametrized in terms of the unknown controller parameter $\theta_c^*(t)$, for which the controller $F_u(\theta_c^*)$ meets the performance requirements to obtain the system model with the same input/output characteristics as $P_c(\theta^*)$. The adaptation law provides an estimate $\theta_c(t)$ at each time instant, which is then used to update the controller without intermediate calculations. The block diagram of the *direct adaptive control* scheme is shown in Figure 1.3.

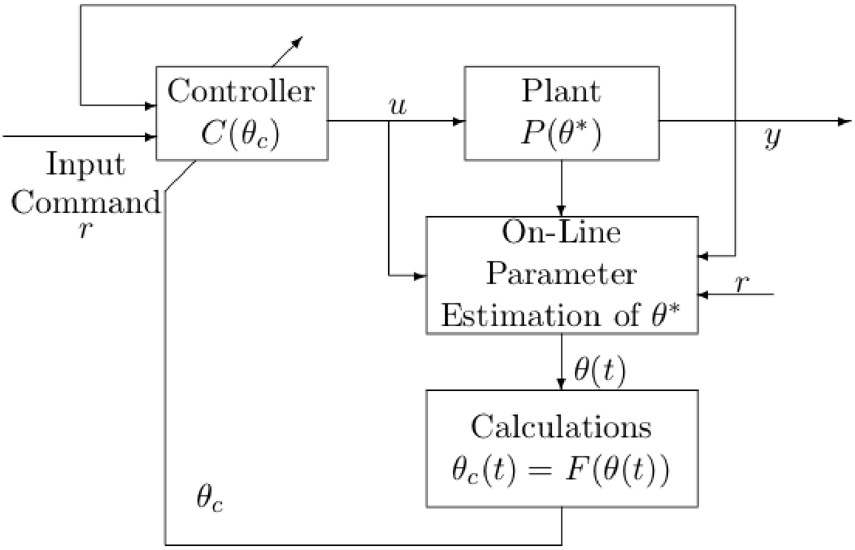


Figure 1.2: *Indirect adaptive control architecture. [9]*

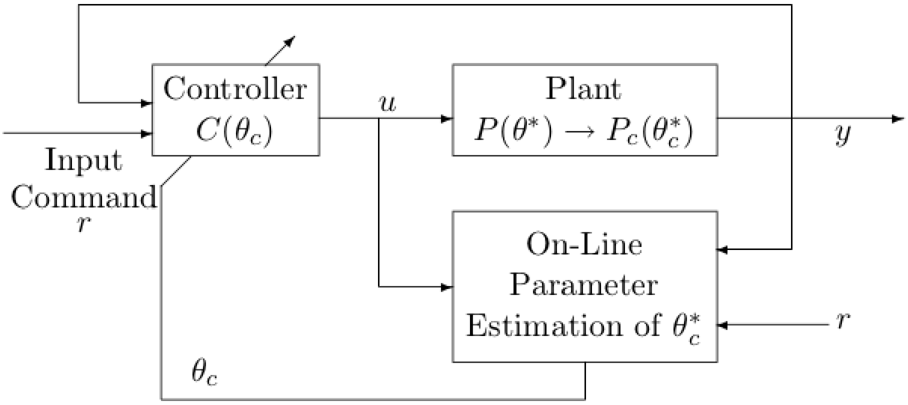


Figure 1.3: *Direct adaptive control architecture. [9]*

1.3.4 Model Reference Adaptive Control

A well-known procedure for determining the structure of the functions F_u and F_θ is denoted as *Model Reference Adaptive Control* (MRAC), originally proposed in 1958 by Whitaker et *al.* at MIT [20]. The main idea behind this concept is to specify the desired command-to-output performance of a servo-tracking system using a differential equation (the *reference model*) that defines the ideal response $y_m(t)$ of the system due to external commands. MRAC was derived from the Model Reference Control (MRC) problem [16]. The objective of MRC is to find a feedback control law that modifies the system dynamics, so that its input/output properties are exactly the same as those of the *reference model*. In the same way, MRAC designs the controller structure $F_u\left(\theta_c^*, \phi, t\right)$ to ensure $\lim_{t\to\infty} y(t) - y_m(t) = 0$ assuming known the ideal (unknown) parameter $\theta_c^*(t)$ such that $\theta_c(t) = \theta_c^*(t)$ in equation (1.3).

Being $\theta_c^*(t)$ unknown, the obtained controller cannot be implemented. One way of dealing with the unknown parameter case is to use the CE-based approach to replace the unknown $\theta_c^*(t)$ in the control law with its estimate $\theta_c(t)$ obtained using the *direct* or *indirect* approach [9]. Indeed, after having determined the controller F_u, MRAC approach focuses on finding the adaptive law F_θ such that output following takes place with the closed-loop system trajectories remaining bounded. A block diagram of a system operating under MRAC controller is shown in Figure 1.4.

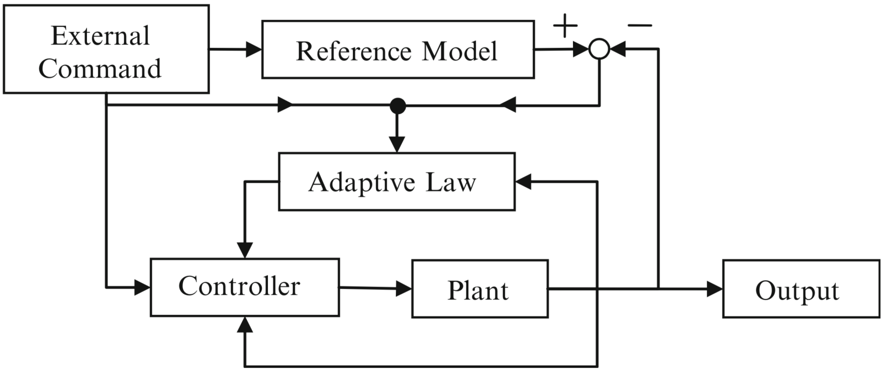


Figure 1.4: *MRAC closed-loop block diagram [16].*

1.3.5 $\mathcal{L}_1$ Adaptive Control

In adaptive control, the nature of the adaptation process plays a central role in both robustness and performance. Ideally, one would like the adaptation to respond correctly to all variations in initial conditions, reference inputs and uncertainties by quickly identifying a set of control parameters that provide a satisfactory response to the system. This goal, of course, requires fast estimation schemes with high adaptation rates and, consequently, leads to the fundamental question of determining the upper limit of the adaptation rate that does not result in poor robustness characteristics [4]. The lack of analytical quantification of the relationship between the rate of adaptation, the transient response, and the robustness margins led to gain-scheduled designs of adaptive controllers, examples of which are the successful flight tests of the late *1990s* by the Air Force and Boeing [16]. These flight tests relied on intensive Monte Carlo analysis to determine the best rate of adaptation for various flight conditions. It was evident that fast adaptation led to high frequencies in the control signals and increased sensitivity to time delays. The fundamental issue therefore came down to determining an architecture that would allow rapid adaptation without losing robustness. It was clear that such an architecture could reduce the amount of gain scheduling and possibly eliminate it, since rapid adaptation in the presence of guaranteed robustness should be able to compensate for the detrimental effects of rapidly varying uncertainties on system response. The $\mathcal{L}_1$ adaptive control theory precisely addressed this issue by setting an architecture for which adaptation is decoupled from robustness, which enables arbitrarily fast adaptation without sacrificing robustness. [41]. In fact, in $\mathcal{L}_1$ adaptive control architectures, the rate of the adaptation loop can be set arbitrarily high, subject only to hardware limitations (computational power and high-frequency sensor noise), while the trade-off between performance and robustness can be addressed through conventional methods from classical and robust control. This separation between adaptation and robustness is achieved by explicitly building the robustness specification into the problem formulation, with the understanding that the uncertainties in any feedback loop can be compensated for only within the bandwidth of the control channel. From an architectural perspective, this modification of the problem formulation leads to the inclusion of a bandwidth-limited filter in the feedback path, which ensures that the control signal remains in the desired frequency range (see Figure 1.5). With $\mathcal{L}_1$ adaptive controller in the feedback loop, the response of the closed-loop system can be predicted a priori, thus significantly reducing the amount of Monte Carlo analysis required for the verification and validation of such systems. These features of $\mathcal{L}_1$ adaptive control theory were verified, consistently with the theory, in many flight tests and high-fidelity simulation environments [4].

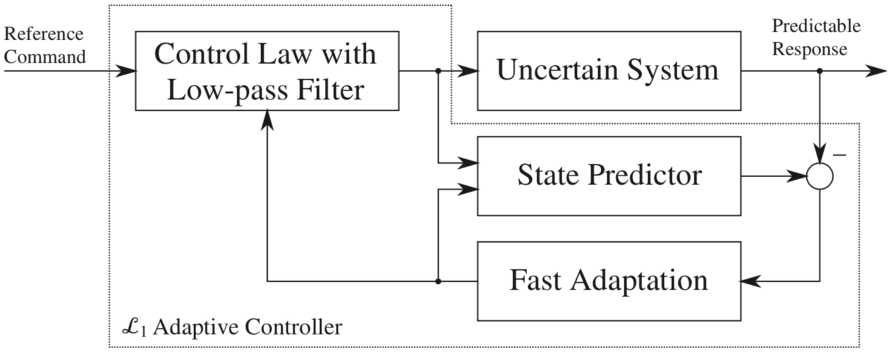


Figure 1.5: *$\mathcal{L}_1$ Adaptive control closed-loop block diagram [41].*

CHAPTER 2

The role of closed-loop attitude dynamics in adaptive UAV position control

This Chapter presents the design and the stability analysis of an adaptive position controller for Unmanned Aerial Vehicles (UAVs). Considering a hierarchical control scheme, the novelty of this work is the definition of a systematic approach to design a position controller based on Model Reference Adaptive Control (MRAC) theory taking into account not-fast closed-loop attitude dynamics. After having reformulated the problem considering the attitude dynamics as a pseudo-actuator, we exploit an existing Linear Matrix Inequality (LMI) based hedging framework designed such that the adaptation performance is not affected by the presence of actuator dynamics. Results from simulations and from experiments on a platform designed to replicate the longitudinal motion of quadrotors are provided to illustrate the performance of the proposed control scheme.

2.1 Introduction

In recent years, the study of UAVs has received increasing attention thanks to their wide range of applications. In particular, quadrotor UAVs have attracted commercial and research interest thanks to their high level of manoeuvrability and simple mechanical structure. Various controllers have been proposed to track a predefined trajectory or a path for quadrotors. Fixed-gain linear or nonlinear controllers often serve to address the problem satisfactorily when dealing with nominal operation (see, *e.g.*, [42] for a comprehensive survey). Specifically, several control systems have been developed based on robust control [43], backstepping [44], sliding mode controller [45], or hybrid control architecture [46]. In [47] and [48], Iterative Learning Control (ILC) was used to update the reference signal to multirotor systems subject to repetitive disturbances to achieve high-performance tracking. However, repetitive operational conditions and controlled environments are essential for these data-based control algorithms. If more challenging scenarios such as, *e.g.*, actuator degradation and faults, severe external disturbances, and parameter uncertainties have to be considered, then approaches capable of learning whilst operating are needed. Adaptive control is an attractive candidate for dealing disturbances and uncertainties for fault-tolerant or reconfigurable unmanned flight [16]. One well-known architecture is the MRAC architecture which has been used multiple times in multirotor control achieving impressive results. The interested reader is referred to [49] as an example of numerous references on adaptive multirotor control. In particular, in recent years the $\mathcal{L}_1$ adaptive control approach has been successfully applied across various UAV applications [41, 4].

Considering a hierarchical control scheme, the contribution of this Chapter is the design and stability analysis of a position controller based on MRAC theory taking systematically into account non-fast closed-loop attitude dynamics. In fact, actuator dynamics pose a significant obstacle to the design and implementation of standard adaptive controllers [16]. In particular, if the actuator dynamics have sufficiently wide bandwidth, then they can be neglected in the design of MRAC [50]. However, if the actuator dynamics do not have sufficiently wide bandwidth or the control system is used for safety-critical applications, a systematic approach must be followed to determine if the actuator bandwidth is large enough to not affect the adaptation performance maintaining the closed-loop dynamical system stable. After having reformulated the problem considering the closed-loop attitude dynamics as pseudo-actuator, the effects of these dynamics on the adaptation performance are analysed by exploiting the LMI-based hedging framework presented in [51]. To the best knowledge of the authors, no theoretical analysis of the attitude dynamics effects on the MRAC position controller has been performed so far for a hierarchical control scheme of multirotor UAVs. This Chapter contributes by evaluating these effects in such an application.

2.2 Problem statement

In this Section, we first show the dynamical model of vectored-thrust[1] UAV and present a hierarchical control law capable of stabilizing simultaneously the UAV position and heading direction. Then, the interactions between attitude and position loops are analysed. Finally, the UAV model is linearised in near hover conditions to fit the MRAC architecture.

2.2.1 Mathematical model

The configuration of a rigid UAV can be identified with the motion of a body-fixed frame $F_B := (O_B, \{b_1, b_2, b_3\})$ with respect to a reference frame $F_I := (O_I, \{i_1, i_2, i_3\})$, where b_j and i_j for $j \in \{1,2,3\}$ are unit vectors forming right-handed orthogonal triads and O_B, O_I are the origins of the body and reference frame, respectively (see Figure 2.1). In the following, the position vector from O_I to O_B, resolved in F_I, is denoted as $p = \begin{bmatrix} p_x \; p_y \; p_z \end{bmatrix}^\top \in \mathbb{R}^3$ while the rotation matrix describing the attitude of the UAV is denoted as $R = \begin{bmatrix} b_1 \; b_2 \; b_3 \end{bmatrix} \in \mathrm{SO}(3)$, where b_i is the *i-th* body axis resolved in F_I. The UAV dynamical model can be described by [52]:

$$\dot{R} = RS(\omega) \qquad J\dot{\omega} = -S(\omega)J\omega + \tau_e + \tau_c \tag{2.1}$$

$$\dot{p} = v \qquad m\dot{v} = -mgi_3 + T_c R i_3 + f_e, \tag{2.2}$$

[1] Multirotor UAVs with coplanar propellers are known as *vectored-thrust* UAVs, because their propulsive system can deliver a control force only along a fixed direction within the airframe [42].

where $J = J^\top \in \mathbb{R}^{3\times 3}_{>0}$ is the UAV inertia matrix with respect to O_B, $m \in \mathbb{R}_{>0}$ is the UAV mass, $g = 9.81\,\mathrm{m/s^2}$ is the gravitational acceleration, $\omega \in \mathbb{R}^3$ is the body angular velocity, $v \in \mathbb{R}^3$ is the inertial translational velocity, $T_c \in \mathbb{R}_{>0}$ and $\tau_c \in \mathbb{R}^3$ are the overall thrust and the torque applied by the propellers, respectively, and $(f_e, \tau_e) \in \mathbb{R}^6$ is the disturbance wrench including, *e.g.*, aerodynamic effects. $S(\cdot)$ is the matrix representation of linear cross-product operation such that for any $a, b \in \mathbb{R}^3$ $S(a)b = a \times b$.

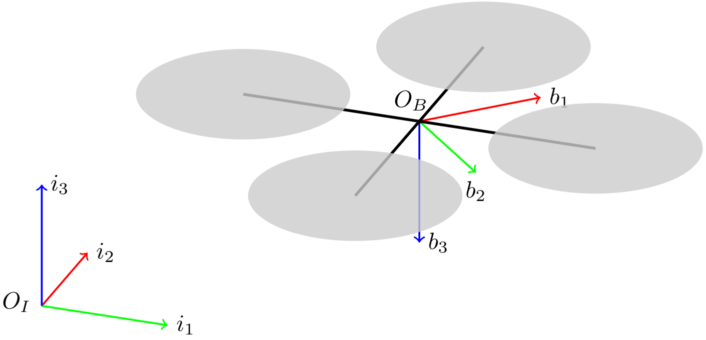


Figure 2.1: *Reference frames.*

2.2.2 Cascade control design for position-yaw stabilization

By relying on the differential flatness property of the dynamics (2.1)-(2.2) with respect to the position vector p and to the rotation about the b_3 axis [53], several control strategies have been proposed in the literature to deal with the nonlinear and underactuated nature of the quadrotor dynamics. In this work, the objective of the control design is to stabilize the UAV at a constant position set-point $p^d = [\,p_x^d \;\; p_y^d \;\; p_z^d\,]^\top \in \mathbb{R}^3$ with a desired yaw angle $\psi^d \in \mathbb{R}$. To tackle the underactuated nature of vectored-thrust UAVs, we follow a hierarchical control strategy in which the attitude dynamics (inner loop) is used to stabilize the translational one (outer loop). Each loop considers the translation and rotation dynamics of the multirotor system separately, hence, reducing the complexity of the control design problem [54]. Specifically, the control architecture involves an adaptive position controller, an attitude extraction block, and a cascade P/PID attitude controller similar to the one implemented in the PX4 autopilot [55]. The overall architecture (see Figure 2.2) is described in the following.

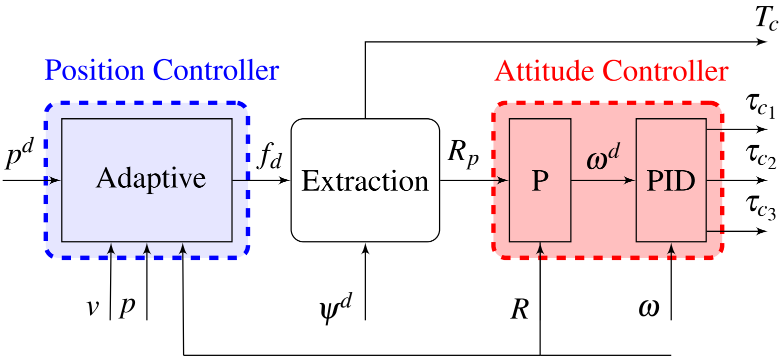


Figure 2.2: *Block diagram of the proposed controller architecture.*

Position Controller

Since the control force in the inertial frame ($T_c R i_3$) cannot be delivered instantaneously in a desired direction, a widely adopted strategy is to introduce a virtual control variable $f_d = \left[f_{d_x}\ f_{d_y}\ f_{d_z}\right]^\top \in \mathbb{R}^3$ in equation (2.2)

$$m\dot{v} = -mgi_3 + f_e + f_d - (f_d - T_c R i_3), \tag{2.3}$$

where f_d should be selected so that the desired set-point ($p = p^d$, $v = 0$) defines an asymptotically stable equilibrium point of the translational dynamics. In this work, we consider for the altitude stabilization a linear cascade controller as:

$$f_{d_z} := PI_z(s)\left(k_p^z(p_z^d - p_z) - v\right) - D_z(s)v_z + mgi_3\ , \tag{2.4}$$

where $PI_z(s) := k_p^{z/i} + \frac{k_i^{z/i}}{s}$ and $D_z(s) := k_d^{z/i}\frac{sT^z}{s+T^z}$ are continuous transfer functions defining, respectively, a proportional-integral and (filtered) output-derivative actions , while k_p^z, $k_p^{z/i}$, $k_i^{z/i}$ and $k_d^{z/i} \in \mathbb{R}_{>0}$ are scalar gain and $T^z \in \mathbb{R}_{>0}$ is the filter time constant. Instead, we consider an adaptive controller for the xy-plane control (described in the following Section) counteracting the unmodelled forces (whose effect in the z-axis is considered negligible in this work) to ensure that the UAV tracks a user-specified command with desired transient performance requirements.

Attitude extraction

The idea behind the hierarchical approach is to find a reference attitude R_p and a control thrust T_c^d such that

$$f_d - T_c^d R_p i_3 = 0. \tag{2.5}$$

Then, by exploiting the full actuation of the rotational dynamics (2.1), the control torque τ_c is designed so that the reference attitude R_p is asymptotically tracked. In this way, the mismatch term $f_d - T_c R i_3$ will converge to zero. To solve equation (2.5), the reference attitude R_p is selected with the third axis aligned with the force required for position stabilization f_d and the rotation about this axis is assigned as a function of a desired yaw angle ψ^d through the unit vector $b_{p_1} := \begin{bmatrix} \cos(\psi^d) & \sin(\psi^d) & 0 \end{bmatrix}^\top$, which represents the heading direction. Thus, a solution to equation (2.5) is $T_c^d = ||f_d||$ and

$$R_p := \begin{bmatrix} \frac{b_{p_3} \times b_{p_1}}{\|b_{p_3} \times b_{p_1}\|} \times b_{p_3} & \frac{b_{p_3} \times b_{p_1}}{\|b_{p_3} \times b_{p_1}\|} & b_{p_3} \end{bmatrix}, \quad b_{p_3} := \frac{f_d}{\|f_d\|}\,. \tag{2.6}$$

Attitude controller

In this Chapter, we consider the following nonlinear cascade controller for attitude stabilization (similar to the one implemented in the commercial PX4 autopilot [55]):

$$\tau_c^d := PI_R(s)\left(\gamma_R(K_p^R R_p^\top R) - \omega\right) - D_R(s)\omega\,, \tag{2.7}$$

where $PI_R(s) := K_p^i + K_i \frac{1}{s}$ and $D_R(s) := K_d \frac{sT}{s+T}$ are continuous transfer functions defining, respectively, a proportional-integral and (filtered) output-derivative actions, while K_p^R, K_p^i, K_i and $K_d \in \mathbb{R}_{>0}^{3\times 3}$ are diagonal gain matrices and $T \in \mathbb{R}_{>0}$ is the filter time constant. The term $\gamma_R(K_p^R R_p^\top R) := -\frac{1}{2} S^{-1}(K_p^R R_p^\top R - R^\top R_p K_p^R)$ is a nonlinear proportional stabilizer computing the reference angular velocity that must be tracked by the inner loop through a PID controller.

2.2.3 Interactions between loops

The classical way to design control laws for multirotor UAVs consists in assuming that the controllers will be tuned such that the attitude dynamics would converge faster than translational dynamics. This time-scale separation allows one to neglect the perturbation acting on the translational (outer) loop due to the dynamics of attitude (inner) loop. Namely, if the perturbation term (defined as $\sigma := \frac{1}{m}T_c(R - R_p)i_3$) vanishes rapidly[2], the complete closed-loop system will be stable in practice [42].

In the literature, different authors quantify how much faster the attitude control loop should be to ensure the closed-loop stability of the whole system (see, *e.g.*, [56] where the stability analysis has been addressed by singular perturbation theory). On the other hand, a hierarchical controller robust to perturbation σ is proposed in [46], where the closed-loop stability of the whole system is ensured combining the Input-to-State Stability (ISS) properties of the position error system with the global asymptotic stability of the attitude error subsystem. Similarly, in [57] the stability of the interconnection between the attitude and position loops is studied within the framework of differential inclusions.

Another way to ensure the stability of the whole system is to use standard tools considering the linearised system and to determine when the closed-loop attitude bandwidth (that can be seen as an actuator of the translational dynamics) is too narrow and instability may occur [58]. However, for adaptive control of uncertain dynamical systems, these tools can no longer be used to determine how wide the actuator (closed-loop attitude) bandwidth needs to be to ensure stability.

One way to address this issue is to reduce the aggressiveness of the adaptive controller (degrading the tracking performance) by trial-and-error. In contrast, we propose a systematic approach that exploits the hedging-based MRAC architecture, initially presented in [59], that allows the adaptation performance to not be affected by the presence of actuator dynamics and exploits a LMI-method proposed in [51] to determine if the actuator bandwidth is large enough.

[2] It is assumed that motor dynamics is negligible with respect to the rigid body dynamics of the UAV, the desired values T_c^d and τ_c^d are considered to be instantaneously reached by $T_c = T_c^d$ and $\tau_c = \tau_c^d$.

2.2.4 Linearised system

To implement the proposed adaptive architecture, the UAV model in (2.1)-(2.2) must be linearised. Assuming near hovering conditions (*i.e.*, $p \approx \bar{p}$, $v \approx 0$, $R \approx I_3 + S(\Delta\alpha)$ with $\Delta\alpha := \left[\phi \;\; \theta \;\; \psi\right]^\top$ being small rotation angles, $\omega \approx 0$) we obtain:

$$\begin{aligned} \Delta\dot{\alpha} &= \Delta\omega, & \Delta\dot{p} &= \Delta v && (2.8)\\ J\Delta\dot{\omega} &= \Delta\tau_c, & m\Delta\dot{v} &= mgS(\Delta\alpha)i_3 + \Delta T_c i_3, && (2.9) \end{aligned}$$

where $\Delta T_c := T_c - mg$, $\Delta\tau_c := \tau_c$, and $\Delta(\cdot)$ represent deviation variables. Similarly, the control law can be linearized by recognizing that in near hovering conditions $\|f_d\| \approx mg$ and $\gamma_R(K_p^R R_p^\top R) \approx K_p^R \left[\phi_p{-}\phi \;\; \theta_p{-}\theta \;\; \psi^d{-}\psi\right]^\top$, so as to obtain:

$$\Delta T_c := PI_z(s)\left(k_p^z(p_z^d - p_z) - v\right) - D_z(s)v_z, \tag{2.10}$$

$$\Delta\tau_c := PI_R(s)(K_p^R\left[\phi_p{-}\phi \;\; \theta_p{-}\theta \;\; \psi^d{-}\psi\right]^\top - \omega) - D_R(s)\omega, \tag{2.11}$$

where the virtual roll and pitch angles are

$$\phi_p := +\tfrac{1}{mg}\, f_{d_x} = \; MRAC(p_y^d, p_y, v_y, \phi), \tag{2.12}$$

$$\theta_p := -\tfrac{1}{mg}\, f_{d_y} = \; MRAC(p_x^d, p_x, v_x, \theta), \tag{2.13}$$

while ψ^d is the desired yaw angle. Consider only the dynamics in the xy-plane, the corresponding system state-space form can be written as:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0\\ -g & 0\\ 0 & 0\\ 0 & g \end{bmatrix} w \tag{2.14}$$

where $x = \left[p_x \;\; v_x \;\; p_y \;\; v_y\right]^\top \in \mathbb{R}^4$ $w := \left[\theta \;\; \phi\right]^\top \in \mathbb{R}^2$ is the output of the closed-loop attitude dynamics given by[3]

$$w(s) = \begin{bmatrix} G_\theta(s) & 0\\ 0 & G_\phi(s) \end{bmatrix} u(s) \tag{2.15}$$

[3] For the sake of simplicity, we interchangeably use time-domain and frequency-domain representations of signals (*e.g.*, $x(t)$ and $x(s)$ denote the function of time and its Laplace transform, respectively).

where $u := \begin{bmatrix} \theta_p & \phi_p \end{bmatrix}^\top \in \mathbb{R}^2$ is the control input given by the adaptive controller, and $G_\theta(s)$ and $G_\phi(s)$ are the continuous transfer functions defining the closed-loop pitch and roll dynamics respectively. These transfer functions can be either retrieved analytically or identified from experimental data. In this Chapter, we focus on the position controller design with an assumption that a well-designed attitude controller has been applied. For the sake of simplicity, in this work $G_\theta(s)$ and $G_\phi(s)$ are approximated by first order dynamical systems[4]. In the following Sections, we refer to closed-loop attitude dynamics as actuator dynamics.

2.3 MRAC with actuator dynamics

This Section provides a concise overview of [51] and [60]. In particular, we consider the uncertain dynamical system subject to actuator dynamics

$$\dot{x} = Ax + Bw, \quad x(0) = x_0, \tag{2.16}$$

where $x \in \mathbb{R}^n$ is the state vector available for feedback, $A \in \mathbb{R}^{n\times n}$ is an unknown system matrix, $B \in \mathbb{R}^{n\times m}$ is a known input matrix, and the pair (A,B) is controllable. $w \in \mathbb{R}^m$ is the actuator output of the actuator dynamics given by

$$\begin{aligned} \dot{x}_\mathrm{c} &= -Mx_\mathrm{c} + u, \quad x_\mathrm{c}(0) = x_{\mathrm{c}0}, \\ w &= Mx_\mathrm{c}, \end{aligned} \tag{2.17}$$

where $x_\mathrm{c} \in \mathbb{R}^m$ is the actuator state vector, $M \in \mathbb{R}^{m\times m}_{>0}$ is a diagonal matrix with diagonal entries representing the actuator bandwidth of each control channel, and $u \in \mathbb{R}^m$ is the control input restricted to the class of admissible controls consisting of measurable functions. Next, we consider the *ideal* reference system capturing a desired closed-loop dynamical system performance given by

$$\dot{x}_\mathrm{i} = A_\mathrm{r}x_\mathrm{i} + B_\mathrm{r}c, \quad x_\mathrm{i}(0) = x_{\mathrm{i}0}\ , \tag{2.18}$$

where $x_\mathrm{i} \in \mathbb{R}^n$ is the ideal reference state vector, $c \in \mathbb{R}^m$ is a given uniformly continuous bounded command, $A_\mathrm{r} \in \mathbb{R}^{n\times n}$ is the Hurwitz reference

[4] The proposed architecture can be extended to higher-order dynamics following the approach in [60].

system matrix, and $B_{\mathrm{r}} \in \mathbb{R}^{n\times m}$ is the command input matrix. We now make the following assumption that is standard in the MRAC literature and is known as the *matching condition* (see, *e.g.*, [16, 9]).

Assumption 1. *There exist an unknown matrix $W_x \in \mathbb{R}^{m\times n}$ and a known matrix $K_r \in \mathbb{R}^{m\times m}$ such that $A_r = A - BW_x^\top$ and $B_r = BK_r$ hold.*

The objective of the MRAC problem is to construct an adaptive feedback control law $u(t)$ such that the state vector $x(t)$ asymptotically follows the reference state vector $x_{\mathrm{i}}(t)$. The standard MRAC formulation can suppress the effect of any model uncertainty to achieve desirable tracking performance requirements in the absence of actuator dynamics [16]. In the presence of actuator dynamics, the standard MRAC formulation based on the (pre-chosen) *ideal* reference model given by equation (2.18) does not allow the uncertain dynamical system to track the reference model trajectories asymptotically.

A remedy to this problem is given by the hedging method (see [59] and [61] for more details). This method alters the trajectories of the reference model to allow adaptive controllers to be developed so that actuator dynamics have no effect on their stability. Namely, we consider the following *modified* reference model

$$\dot{x}_{\mathrm{r}} = A_{\mathrm{r}}x_{\mathrm{r}} + B_{\mathrm{r}}c + B(w-u), \quad x_{\mathrm{r}}(0) = x_{\mathrm{r}0}\ , \tag{2.19}$$

where the deficit term $B(w-u)$ is introduced. Let the adaptive feedback control law be given by

$$u = -\hat{W}_x^{\mathrm{T}}x + K_rc\ , \tag{2.20}$$

where $\hat{W}_x \in \mathbb{R}^{n\times m}$ is the estimate of W_x, which is obtained with the adaptation law

$$\dot{\hat{W}}_x = \gamma \mathrm{Proj}\left[\hat{W}_x, xe^{\mathrm{T}}PB\right], \quad \hat{W}_x(0) = \hat{W}_{x0}\ , \tag{2.21}$$

where $\mathrm{Proj}[\cdot,\cdot]$ is the projection operator [16], $\gamma \in \mathbb{R}_{>0}$ being the learning rate, $e \triangleq x - x_{\mathrm{r}}$ being the system error state vector, and $P \in \mathbb{R}^{n\times n}_{>0}$ being the solution of the Lyapunov equation

$$0 = A_r^TP + PA_r + Q \tag{2.22}$$

with $Q = Q^\top \in \mathbb{R}_+^{n\times n}$. In addition, the projection bounds are defined such that $\left|\left[\hat{W}_x\right]_{ij}\right| \leq \hat{W}_{x,\max,i+(j-1)n}$ for $i = 1,\dots,n$ and $j = 1,\dots,m$, where $\left[\hat{W}_x\right]_{ij}$ denotes the ij-th entry of the matrix $\hat{W}_x$ and $\hat{W}_{x,\max,i+(j-1)n} \in \mathbb{R}_{>0}$ are symmetric[5] element-wise projection bounds. In [51], Gruenwald *et al.* proved that the solutions $\left(e,\hat{W}_x,x_\mathrm{r},w\right)$ of the closed-loop system are bounded and $\lim_{t\to\infty} e = 0$ if the following condition is satisfied.

Assumption 2. *Let $\bar{W}_{x_{i_1,\dots,i_l}} \in \mathbb{R}^{n\times m}$ be defined as where $i_l \in \{1,2\}$, $l \in \{1,\dots,mn\}$, such that $\bar{W}_{x_{i_1,\dots,i_l}}$ represents the corners of the hypercube defining the maximum variation of $\hat{W}_x(t)$ ensured by the projection operator. The matrix*

$$\mathcal{A}_{i_1,\dots,i_l} = \begin{bmatrix} A_r + B\bar{W}^\mathrm{T}_{x_{i_1,\dots,i_l}} & BM \\ -\bar{W}^\mathrm{T}_{x_{i_1,\dots,i_l}} & -M \end{bmatrix} \tag{2.23}$$

satisfies the matrix inequality

$$\mathcal{A}^\mathrm{T}_{i_1,\dots,i_l}\mathcal{P} + \mathcal{P}\mathcal{A}_{i_1,\dots,i_l} < 0, \quad \mathcal{P} = \mathcal{P}^\mathrm{T} > 0, \tag{2.24}$$

for all permutations of $\bar{W}_{x_{i_1,\dots,i_l}}$.

Remark 4. *Since Assumption 2 is satisfied for large values of M [51], we can cast* (2.24) *as a convex optimisation problem whose solution M is the minimum actuator bandwidth that satisfies Assumption 2 for the given level of system uncertainty. From a practical standpoint, there should be a basic tradeoff between the permitted system uncertainties and the actuator dynamics, as remarked in [60].*

2.4 Adaptive UAV position control

In this Section, we compare in simulation the proposed control architecture (with hedged reference model) with the standard MRAC. The task is to track a stair sequence in the xy-plane maintaining $\psi_d = 0$. In particular, we use a high-fidelity simulator of quadrotor dynamics in which the altitude and attitude controllers are the ones described in Section 2.2.2 (which

[5] Note that the results of this Section can be extended to the case when asymmetric projection bounds are considered, as in [50].

behave for small errors as Section 2.2.4). The UAV mass is $m = 0.250\,\mathrm{kg}$ and the attitude loops are tuned such that their closed-loop dynamics can be well approximated by first-order systems with $M_\phi = 9\,\mathrm{rad/s}$ and $M_\theta = 5.5\,\mathrm{rad/s}$ as roll and pitch bandwidth, respectively[6]. Furthermore, we consider the body-drag force using the simplified model $f_e := -c_D\|v\|v$ with $c_D = 0.1$ being the body drag coefficient. The adaptive controller synthesis has been carried out by using the linear system (2.14) with zero initial conditions designing independently the $x-$ and $y-$axis. For both the standard MRAC and the hedged-one, we set $Q = I_2$ from (2.22) and select a second-order reference system with zero initial conditions, a natural frequency of $\omega_\mathrm{n} = 4\,\mathrm{rad/s}$, and a damping ratio $\zeta = 0.7$. Using the rectangular projection operator, the bounds on the uncertainty (considered equal for the $x-$ and $y-$axis) are set element-wise such that $|\left[\hat{W}_x\right]_{1,1}| < \omega_n^2/g \leq 1.8$ and $|\left[\hat{W}_x\right]_{2,1}| < 2\zeta\omega_n/g \leq 0.65$. Using these bounds and considering the system dynamics (2.14), we ensure that the uncertain parameters belong to the convex set delimited by the projection operator centred on the origin. Then, using the projection bounds in the LMI optimization problem highlighted in Remark 4, the minimum allowable actuator bandwidth is computed as $M_{(\cdot)}^{\min} = 5.5\,\mathrm{rad/s}$. The adaptation rate is selected as $\gamma_H = 300$ for the proposed architecture and as $\gamma_M = 100$ for the standard MRAC. This choice is dictated by the fact that using a larger adaptation rate $\gamma_M > 100$ for the standard MRAC architecture leads to instability. Figure 2.3 and Figure 2.4 show the results obtained with the standard MRAC controller in the presence of fast (roll dynamics) and not-fast (pitch dynamics). We can notice that, especially in x-direction (with not-fast pitch dynamics), the standard MRAC controller introduces unwanted oscillations and deteriorates the tracking performance. On the other hand, Figure 2.5 and Figure 2.6 show the performance of the proposed adaptive architecture. After the learning transient, the system response becomes almost identical to the *modified* reference one. Finally, we define the following metric to compare the performance achieved by the two controllers:

$$e^P(t) := \sqrt{(p_x(t) - p_x^i(t))^2 + (p_y(t) - p_y^i(t))^2}\,, \qquad (2.25)$$

[6] For the sake of conciseness, only the parameters related to translational dynamics are reported.

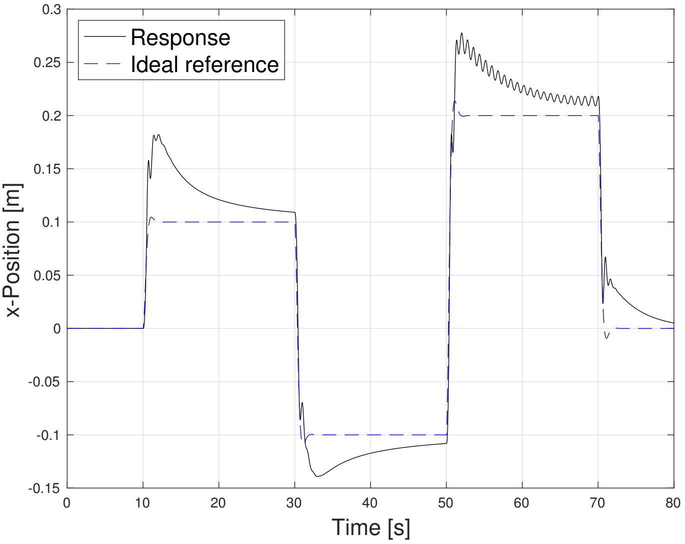


Figure 2.3: *Standard MRAC performance in x-direction.*

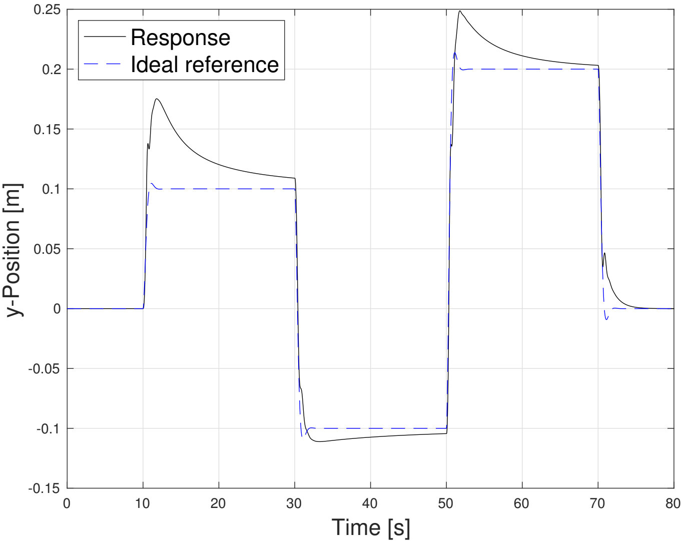


Figure 2.4: *Standard MRAC performance in y-direction.*

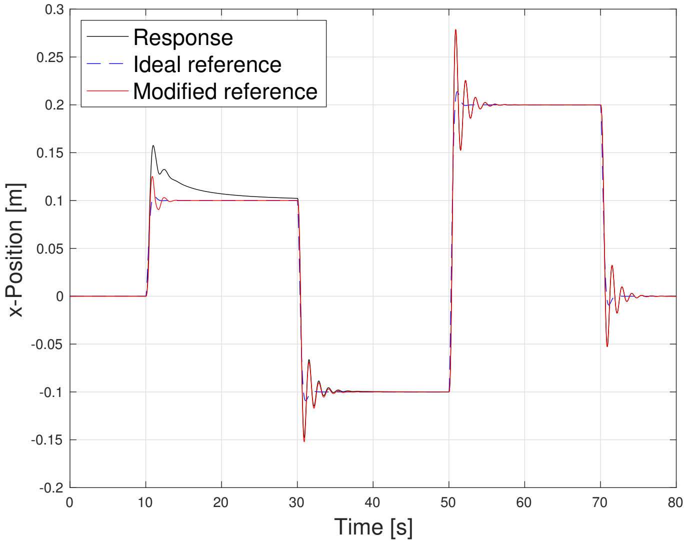


Figure 2.5: *Proposed architecture performance in x-direction.*

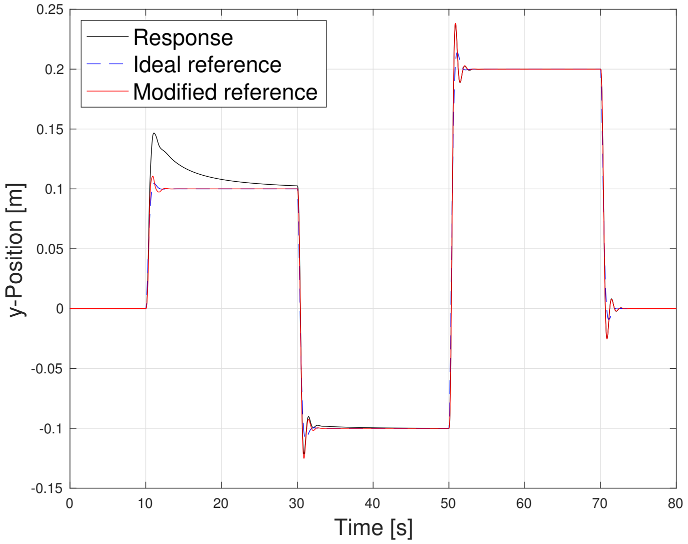


Figure 2.6: *Proposed architecture performance in y-direction.*

where $p_x^i(t)$ and $p_y^i(t)$ are the x- and y-position given by the *ideal* reference system, respectively. We report in Table 10.2 the peak ($e_{PK}^P = \max_t e^P(t)$) and the root mean square (e_{RMS}^P) of the signal $e^P(t)$.

Table 2.1: *Simulation results: performance metrics.*

	Hedging-based	Standard MRAC
Peak e_{PK}^P [m]	0.0805	0.1110
RMS e_{RMS}^P [m]	0.0136	0.0314

We can state that the proposed architecture is effective in improving the UAV tracking performance providing a systematic way to take into account the closed-loop attitude dynamics.

2.5 Experimental results

The experimental tests performed on the ANT-X 2DoF drone (see Figure 2.7) are presented and discussed in this Section. The tests are intended to show and compare the behaviour of the proposed control architecture with the standard MRAC scheme in a realistic scenario. The ANT-X

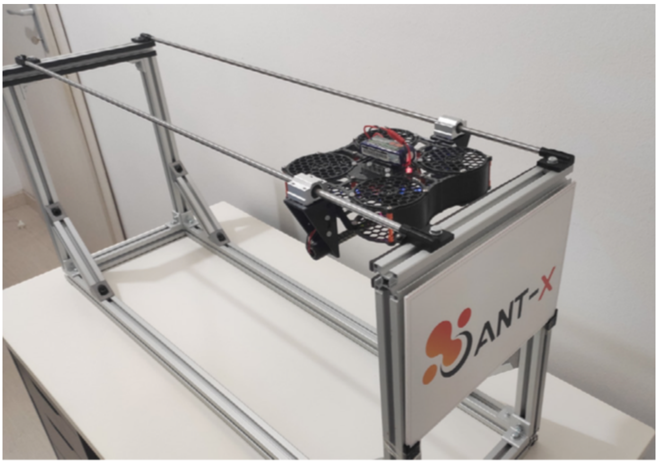


Figure 2.7: *The ANT-X 2DoF drone.*

2DoF drone setup consists of a small quadrotor UAV constrained to operate along two linear guides, allowing only pitch rotation and longitudinal motion [62]. It has been designed to replicate the longitudinal motion of quadrotors, which is described by the following equations:

$$\dot{\theta} = q, \qquad J_\theta \dot{q} = M_{c_x} + M_{e_x} \tag{2.26}$$

$$\dot{p_x} = v_x, \qquad m\dot{v}_x = -T_c \sin\theta + f_{e_x} \tag{2.27}$$

where θ, $q \in \mathbb{R}$ are the pitch angle and rate, respectively, p_x, $v_x \in \mathbb{R}$ are the position and velocity along the x-axis, respectively, $m \in \mathbb{R}_{>0}$ is the quadrotor mass, $J_\theta \in \mathbb{R}_{>0}$ is the pitch inertia moment, $T_c \in \mathbb{R}_{>0}$ is the control thrust (imposed constant during the experiments $T_c = mg$), $\tau_{c_x} \in \mathbb{R}$ is the control torque, while M_{e_x}, $f_{e_x} \in \mathbb{R}$ are torque and force disturbances along the x-axis. Being able to study the longitudinal dynamics is important since it captures all the most relevant challenges associated with the underactuated nature of co-planar multirotors. In fact, the conceptual design behind the ANT-X 2DoF drone was the development of a platform suitable for the testing and validation of flight control concepts for multirotor UAVs (*e.g.*, position and attitude controllers design) in a safe and controlled environment and in a repeatable manner [63]. The task of the experimental campaign is to track a stair sequence in the x-direction. To implement the proposed adaptive architecture the model (2.26)-(2.27) is linearised assuming small rotation angle ($\sin\theta \approx \theta$). Similarly to the previous Section, for both the standard MRAC and the hedged-one, we set $Q = I_2$ from (2.22) and select a second-order reference system with zero initial conditions, a natural frequency of $\omega_\mathrm{n} = 2.3\,\mathrm{rad/s}$, and a damping ratio $\zeta = 0.7$. Using the rectangular projection operator, the bounds on the uncertainty are set element-wise such that $\mid \left[\hat{W}_x\right]_{1,1} \mid < \omega_n^2/g \leq 0.55$ and $\mid \left[\hat{W}_x\right]_{2,1} \mid < 2\zeta\omega_n/g \leq 0.35$. Then, using the bounds on $\hat{W}_x$ in the LMI optimization problem, the minimum allowable pitch closed-loop bandwidth is computed as $M_\theta^{\min} = 2.9\,\mathrm{rad/s}$. The attitude loop is tuned such that the closed-loop pitch dynamics can be well approximated by a first-order system with $M_\theta = 3\,\mathrm{rad/s}$ as bandwidth that is close to the allowable limit $M_\theta^{\min}$. Figure 2.8 and Figure 2.9 show the position response of the standard MRAC architecture and of the proposed one using $\gamma_H = \gamma_M = 100$.

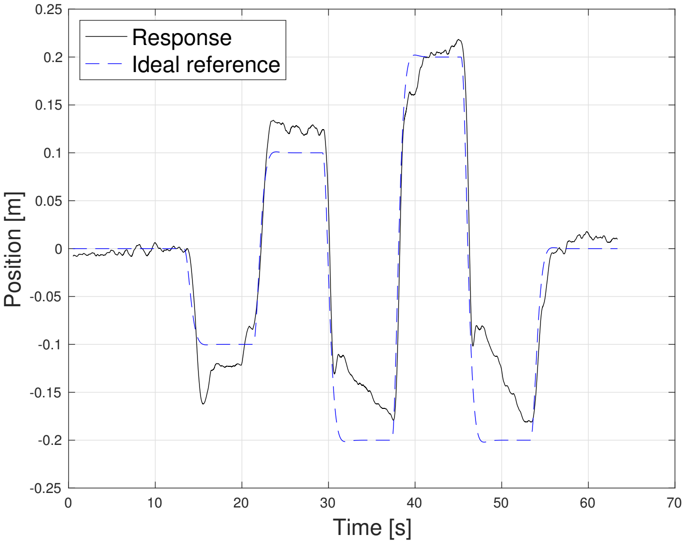


Figure 2.8: *Drone position with the standard MRAC architecture.*

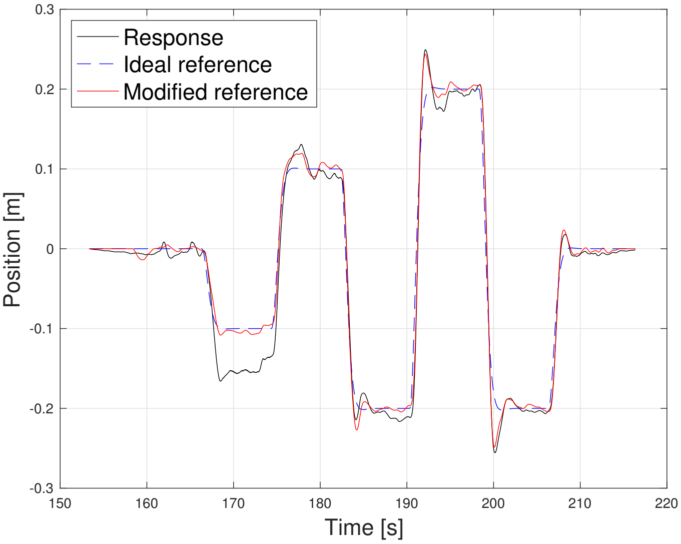


Figure 2.9: *Drone position with the proposed architecture.*

Similarly to the previous Section, we define the following metric to compare the performance achieved by the two controllers:

$$e^x(t) := |(p_x(t) - p_x^i(t))| \,, \tag{2.28}$$

where $p_x^i(t)$ is the x-position given by the *ideal* reference system. We report in Table 2.2 the peak ($e^x_{PK} = \max_t e^x(t)$) and the root mean square (e^x_{RMS}) of the signal $e^x(t)$. A remarkable performance improvement is obtained with the proposed approach. Furthermore, the LMI-based feasible limit $M_\theta^{\min}$ provides a (conservative) lower bound on the allowable closed-loop attitude bandwidth such that the overall closed-loop system remains bounded, avoiding trial-and-error tuning procedure. For the sake of completeness, we show the difference between commanded attitude and the current one in Figure 2.10, and the time evolution of the adaptive parameters $\hat{W}_x$ in Figure 2.11 with the corresponding projection bounds.

Table 2.2: *Experiment performance metrics.*

	Hedging-based	Standard MRAC
Peak e^x_{PK} [m]	0.0860	0.1211
RMS e^x_{RMS} [m]	0.0275	0.0428

2.6 Concluding remarks

In this Chapter, we defined a systematic approach to design a position controller based on MRAC theory for a multirotor UAV taking into account the closed-loop attitude dynamics. In particular, after having reformulated the problem considering attitude dynamics as an actuator for the translational dynamics, we exploited the LMI-based hedging MRAC controller (initially proposed in [51]) to achieve desirable tracking performance specifications despite uncertainties. Finally, results from simulation and experiments on the ANT-X 2DoF drone showed the effectiveness of the proposed strategy. Future research can extend the proposed approach by leveraging the *command governor* architecture presented in [50], where asymptotic convergence to the *ideal* trajectories of the reference model is guaranteed. This modification could potentially recover the goal of standard MRAC, considering the effects of actuator dynamics.

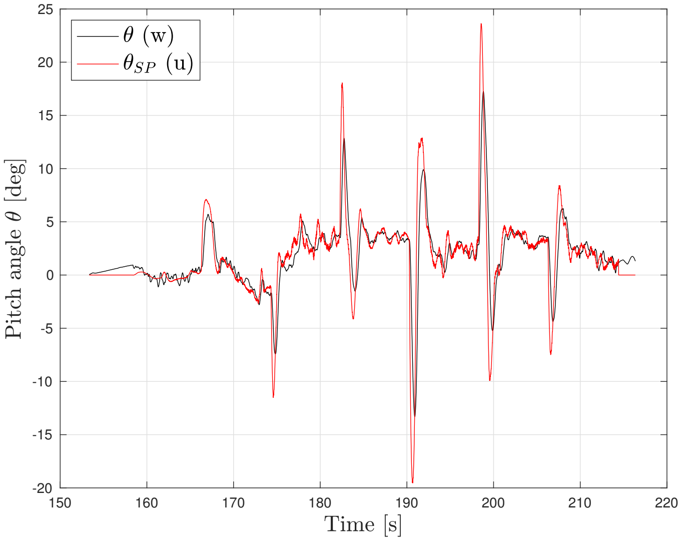


Figure 2.10: *Commanded attitude vs attitude response.*

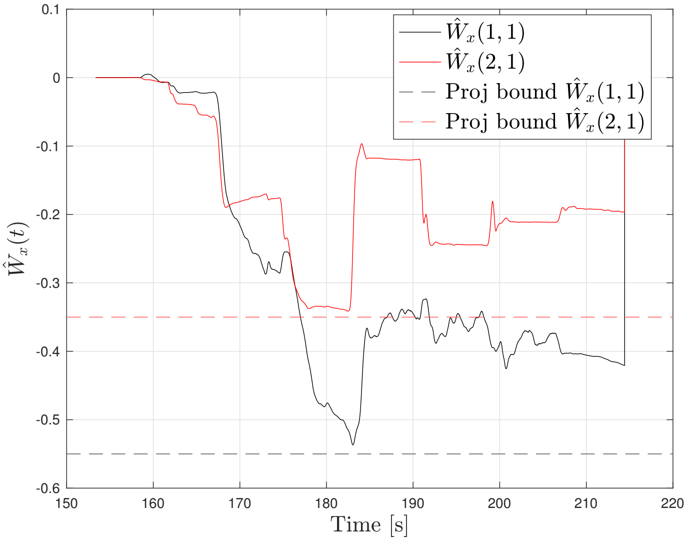


Figure 2.11: *Time evolution of the adaptation parameters $\hat{W}_x$.*

CHAPTER 3

Adaptive control for time-varying systems

In the previous Chapters, different approaches to designing an adaptive controller with constant unknown parameters have been presented. Unfortunately, these controllers cannot handle time-varying (potentially nonlinear) dynamics that is common for aerospace systems to exhibit. This Chapter addresses this issue by developing a continuous adaptive controller for nonlinear dynamical systems with linearly parametrizable uncertainty involving time-varying uncertain parameters. In the first part of the Chapter, we present a brief overview of the various adaptive control solutions proposed in the literature to handle uncertain time-varying systems, highlighting common points and differences between the different approaches developed over the years. Then, we propose a modification of *congelation of variable* method to achieve trajectory tracking for a scalar nonlinear system. In the last part of the Chapter, numerical results are provided to illustrate the performance of the proposed scheme.

3.1 Historical background

Since the *1950s*, adaptive control has undergone extensive research (see *e.g.*, [12, 64, 9]) being an effective way to cope with system uncertainties. In most cases, adaptive control is designed to handle uncertain systems with unknown constant parameters and unknown external disturbances [65, 66]. However, there are relatively few results for adaptive control of uncertain systems with unknown time-varying parameters. Some pioneering works on adaptive control for time-varying systems (see *e.g.*, [67]) required the PE condition to guarantee closed-loop stability ensuring that parameter estimates converge to the actual parameters. Then, Goodwin *et al.* in [68] avoided the PE condition, but assumed that the parameter variations and their derivatives are bounded in the average (integral) sense. According to [69], most recent works can be divided into two categories:

- *Robust adaptive law* or *switching σ-modification*: This type of algorithms uses a mechanism that adds leakage to the parameter update law if the parameter estimates drift out of a pre-specified reasonable region to guarantee boundedness of the parameter estimates (see, *e.g.*, [9] for more details). These techniques guarantee asymptotic tracking only when the parameters are constant. Otherwise, the tracking error converges to a bounded set whose dimension is related to the rates of the parameter variations [70].

- *Filtered transformation with projection operator*: This category of algorithms involves an adaptive observer described via a change of coordinates with a projection operator, which confines the parameter estimates within a pre-specified compact set to guarantee the boundedness of the parameter estimates (see, *e.g.*, [71]). These techniques guarantee asymptotic tracking only if the parameters are bounded in a compact set, their derivatives are $\mathcal{L}_1$ [72].

These methods based on using a single identification model are found to be inadequate to cope with large uncertainties or rapidly time-varying parameters. During the past 25 years, efforts have been made to extend the general methodology of adaptive control by the use of *multiple identifi-*

cation models [73, 74, 75]. Initially, the idea was limited to switching between multiple fixed controllers in a deterministic sequence. However, the result was not feasible for most applications, as it is challenging to decide on a meaningful controller sequence for different plants. Later, Narendra *et al.* proposed a novel switching architecture, known as *indirect switching* in [76]. In this architecture, the system selects the model by utilizing the properties of the system parameter space. Instead of directly screening the system states or outputs, the switching systems switch the models of the parameters in a predefined parameter space. The *indirect switching* control architecture is schematized in Figure 3.1.

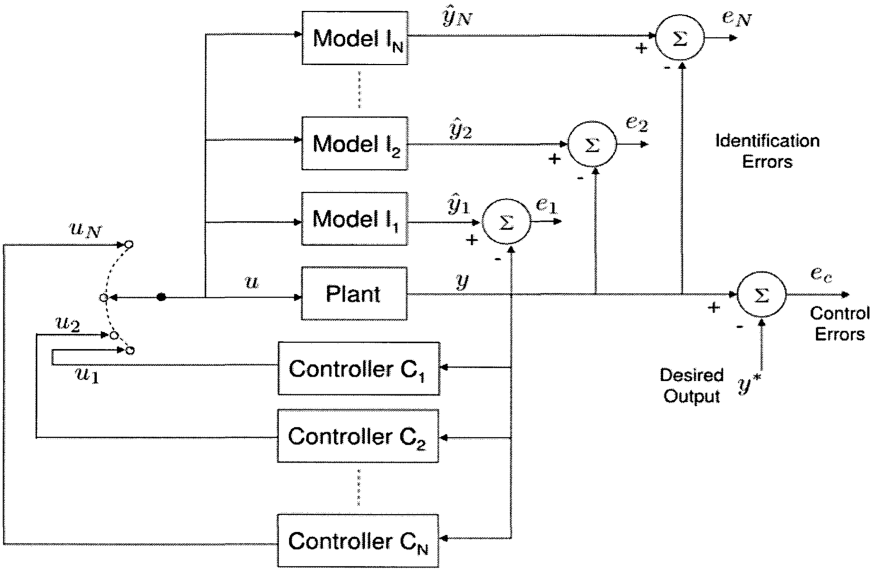


Figure 3.1: *Architecture for indirect switching adaptive control [76].*

Among numerous modifications of the aforementioned approach that have been proposed, the *second level adaptation* approach has emerged over the years as the most successful one [77]. Specifically, it uses the information generated by a finite number of conventional adaptive identifiers (referred to as *first level*) to re-parameterize and identify rapidly the unknown plant (at a *second level*). The accepted philosophy in the *multiple* *identification models* approaches has been that if an adaptive system is fast and accurate in a time-invariant environment, it will perform satisfactorily in a time-varying environment [78, 79].

Over the past decade, researchers have focused on improving the tracking performance of the MRAC approach by proposing novel architectures to handle time-varying parameters. In [80, 81], Gaudio *et al.* presented a novel parameter estimation algorithm for a class of time-varying plants. This algorithm involves a matrix of time-varying adaptation gains, which enables exponentially fast trajectories of parameter estimation errors towards a compact set whenever a finite excitation condition is satisfied. It is shown that even in the presence of time-varying parameters, this algorithm guarantees global boundedness of the state and parameter errors of the system. In [82], Arabi *et al.* proposed a set-theoretic control architecture to limit the tracking error within a prescribed performance bound, while rejecting the effects of parameter variation.

All the above methods may improve the transient response, but they do not achieve asymptotic convergence of the tracking error to zero. Indeed, the time-derivative of the parameter behaves as an unknown exogenous disturbance in the parameter estimation dynamics, which is difficult to cancel with a conventional adaptive update law in a Lyapunov-based stability analysis. Iterative learning control approaches (see, *e.g.*, [83]) produce asymptotic tracking for nonlinear systems involving periodic time-varying uncertain parameters. However, repetitive operational conditions and controlled environments are essential for these data-based control algorithms. In [84], Patil *et al.* addressed this technical challenge by omitting the terms related to the uncertain parameter estimation error in the Lyapunov function, and including a P-function as in [85]. The resulting architecture yields asymptotic tracking error convergence and boundedness of the closed-loop signals for Euler-Lagrange systems. Similarly, to avoid involving the time derivative of the parameter in the stability analysis, Lin *et al.* did not use a time-varying parameter controller that would directly cancel the time-varying perturbation in [86]. Instead, Lin *et al.* designed a constant parameter controller through dominance design, whose constant parameter was estimated online. Specifically, using a parameter separation technique and the tool of *adding a power integrator* (see [87] for more details), the authors developed a feedback domination design approach for the explicit construction of a smooth adaptive controller that solves the problem of global state regulation without imposing any extra conditions on the unknown parameters [86]. Another architecture that guarantees asymptotic regulation was proposed by Chen *et al.* in [88]. Namely, the proposed method, called *congelation of variables*, treats each unknown time-varying parameter as a nominal constant unknown parameter perturbed by a time-varying perturbation. The resulting control input involves an adaptive term to eliminate effects due to the nominal constant parameters and a robust term to dominate the time-varying perturbation.

To the best knowledge of the author, the *congelation of variables* approach has not been applied to achieve trajectory tracking. Indeed, no theoretical analysis of the effects of a regression matrix non-vanishing with the state has been performed so far for this approach. This Chapter contributes by evaluating these effects in such an application and proposing a modification to extend the *congelation of variables* method to the trajectory tracking problem.

3.2 Notation

For an n-dimensional time-varying signal $s : \mathbb{R} \to \mathbb{R}^n$, the image of which is contained in a compact set $\mathcal{S}, \Delta_s : \mathbb{R} \to \mathbb{R}^n$ denotes the deviation of s from a constant value ℓ_s ($\Delta_s(t) = s(t) - \ell_s$), and $\delta_s \in \mathbb{R}$ denotes the supremum of the 2-norm of s ($\delta_s = \sup_{t \geq 0} |s(t)| \geq 0$). In this Chapter, we consider that the unknown time-varying system parameters $\theta(t) : \mathbb{R} \to \mathbb{R}^q$ verify the following assumption.

Assumption 3. *The unknown parameter θ is piecewise continuous and $\theta(t) \in \Theta_0$, for all $t \geq 0$, where Θ_0 is a compact set. The "radius" δ_{Δ_θ} of Θ_0 is assumed to be known, while Θ_0 can be unknown (see Figure 3.2).*

3.3 *Congelation of variable* method: State regulation

The *congelation of variables* method has been initially proposed in [88] to achieve asymptotic state regulation for linear systems with time-varying

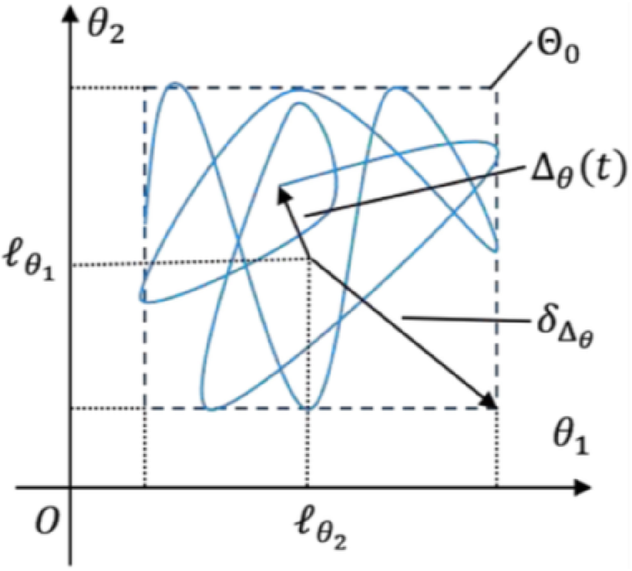


Figure 3.2: *Graphical illustration of the role of* Θ_0 , ℓ_θ , Δ_θ, *and* δ_{Δ_θ} *[69].*

parameters. Then, the theory is extended to deal with non-linear systems both with state and output feedback considering *matched* and a special case of *unmatched* uncertainties (non-linear system in the so-called *parametric strict-feedback* form, where the unknown parameters are separated from the input by integrators) [69, 89]. Since the method does not alter the classical parameter update law created for time-invariant systems, it is compatible with the majority of adaptive control schemes using parameter estimates. Indeed, the *congelation of variables* method has been developed, for instance, in combination with adaptive backstepping [90], the *immersion and invariance* approach [91, 92], and others techniques [93].

This method treats each unknown time-varying parameter as a combination of a nominal unknown constant parameter and a time-varying perturbation term due to the difference between the actual parameter and the nominal one. Taking advantage of this formulation, the controller design is divided into a classical adaptive control design with constant unknown parameters and a damping design via dominance to accommodate the time-varying perturbation terms.

3.3.1 Motivating example

This Section provides a motivating example to highlight the advantages of the *congelation of variables* method. Namely, we present the steps for the derivation of the control law following the approach in [69] and compare them to the classical approach (see, *e.g.*, [9]). We consider a scalar nonlinear system defined as:

$$\dot{x} = \theta x^2 + u \tag{3.1}$$

where $x \in \mathbb{R}$ is the state, $u \in \mathbb{R}$ is the input, and $\theta \in \mathbb{R}$ is an unknown time-varying parameter satisfying Assumption 3.

Classical approach

A standard procedure to deal with the unknown parameter θ is to consider a quadratic Lyapunov function candidate of the form

$$V(x, \hat{\theta}, \theta) = \frac{1}{2}x^2 + \frac{1}{2\gamma_\theta}(\theta - \hat{\theta})^2 \, , \tag{3.2}$$

where $\hat{\theta} \in \mathbb{R}$ is an estimate of the parameter θ. Assuming θ is differentiable with respect to time and taking the Lie derivative of V along the solutions of (3.1) yields

$$\dot{V} = \hat{\theta}x^3 + ux + (\theta - \hat{\theta})x^3 - (\theta - \hat{\theta})\frac{\dot{\hat{\theta}}}{\gamma_\theta} + (\theta - \hat{\theta})\frac{\dot{\theta}}{\gamma_\theta} \, . \tag{3.3}$$

The effect of the unknown $(\theta - \hat{\theta})x^3$ term can be cancelled selecting the parameter update law as

$$\dot{\hat{\theta}} = \gamma_\theta x^3 \, , \tag{3.4}$$

where the constant $\gamma_\theta \in \mathbb{R}_{>0}$ is the adaptation gain. The classical adaptive control approach, assuming θ is constant, involves the control law [9]

$$u = -kx - \hat{\theta}x^2 \tag{3.5}$$

with $k \in \mathbb{R}_{>0}$, which yields $\dot{V} = -kx^2 \leq 0$. Hence, the closed-loop trajectories are bounded, and x converges to 0 by invoking Barbalat's lemma[1].

[1] *Barbalat's lemma:* Let $f : R \to R$ be a uniformly continuous function on $[0, \infty)$. Suppose that $\lim_{t\to\infty} \int_0^t f(\tau)\mathrm{d}\tau$ exists and is finite. Then, $\lim_{t\to\infty} f(t) = 0$.

On the other hand, the sign-indefinite term "$(\theta - \hat{\theta})\dot{\theta}/\gamma_\theta$" must be considered when $\dot{\theta} \neq 0$. The *projection operator* [94], which confines the estimated parameter $\hat{\theta}$ inside a convex compact set and, as a result, ensures the boundedness of $(\theta - \hat{\theta})$, is one way to accommodate the sign-indefinite term by modifying the adaptation law (3.4). Consequently, the boundedness of $\dot{\theta}$ guarantees the boundedness of x (either exact boundedness, as in [95], or boundedness in an average sense, as in [68]). Furthermore, the asymptotic convergence of x to 0 is guaranteed if $\dot{\theta} \in \mathcal{L}_1$ [96]. Using the *switching σ-modification* (also referred to as *soft projection*) is another way to ensure the boundedness [97]. Namely, employing this modification, leakage is added to the adaptation law (3.4) when the parameter estimate drifts outside of a predefined region.

Remark 5. *The aforementioned adaptive schemes share the similarity of treating $\dot{\theta}$ as a disturbance. Hence, these approaches cannot guarantee zero-error regulation when the unknown parameter is persistently time-varying, i.e., $\dot{\theta}$ is non-vanishing [93].*

Novel approach

Chen and Astolfi presented in [88] an approach to achieve stabilization under parameter variations that relies on eliminating the effects of varying parameters from the Lyapunov function. This approach is based on the fact that, since $\dot{\theta}$ appears in $\dot{V}$ because of the parameter estimation error term "$1/(2\gamma_\theta)(\theta - \hat{\theta})^2$" in V, removing θ from the parameter estimation error term also results in removing $\dot{\theta}$ from $\dot{V}$. In the classical approach, this term is included not to guarantee the convergence of the estimation error, but only to ensure the boundedness of $\hat{\theta}$. Thus, congealing θ, *i.e.*, replacing θ with a constant ℓ_θ to be determined, can guarantee the same properties. Following this reason, we consider the modified Lyapunov function candidate

$$V_\ell\left(x, \hat{\theta}, \ell_\theta\right) = \frac{1}{2}x^2 + \frac{1}{2\gamma_\theta}\left(\ell_\theta - \hat{\theta}\right)^2. \tag{3.6}$$

Taking the Lie derivative of V_ℓ along the trajectories of (3.1) yields

$$\dot{V}_\ell = \hat{\theta}x^3 + ux + \left(\ell_\theta - \hat{\theta}\right)x^3 - \left(\ell_\theta - \hat{\theta}\right)\frac{\dot{\hat{\theta}}}{\gamma_\theta} + \Delta_\theta x^3 \tag{3.7}$$

where $\Delta_\theta = \theta - \ell_\theta$. Note that the *congelation* approach removes the $\dot{\theta}$ term from $\dot{V}_\ell$, but adds a perturbation term $\Delta_\theta x^3$ due to the difference between θ and ℓ_θ. Considering the same adaptation law (3.4) of the classical approach and the following control law

$$u = -\left(k + \frac{1}{2\varepsilon}\delta_{\Delta_\theta}\right)x - \frac{1}{2}\varepsilon\delta_{\Delta_\theta}x^3 - \hat{\theta}x^2 \tag{3.8}$$

where $k \in \mathbb{R}_{>0}$ and $\varepsilon \in \mathbb{R}_{>0}$ is a constant balancing the linear and nonlinear damping terms, we obtain

$$\begin{aligned}\dot{V}_\ell &= -\left(k + \frac{1}{2\varepsilon}\delta_{\Delta_\theta}\right)x^2 - \frac{1}{2}\varepsilon\delta_{\Delta_\theta}x^4 + \Delta_\theta x^3 \\ &\leq -kx^2 \leq 0\end{aligned} \tag{3.9}$$

Hence, using the same argument of the classical approach, the closed-loop trajectories are bounded, and invoking Barbalat's lemma leads to $\lim_{t\to\infty} x = 0$ without any assumptions about $\dot{\theta}$. The method of substituting the constant ℓ_θ for the time-varying θ is called *congelation of variables* [69]. The controllers based on the *congelation of variables* approach can be applied to systems with fast-varying parameters, because the design is independent of any $\dot{\theta}$ properties.

Remark 6. *As noted in [69], the control law* (3.8) *and the adaptation law* (3.4) *are independent of ℓ_θ, just as conventional adaptive controllers are independent of θ, thus showing the adaptiveness of the proposed mechanism. The resulting controller can be seen as a combination of an adaptive controller, to cope with the unknown (constant) parameter ℓ_θ, and a robust controller, to cope with the time-varying perturbation Δ_θ. Moreover, when θ is a constant, the control law* (3.8) *is reduced to the classical control law* (3.5) *selecting $\ell_\theta = \theta$ and $\delta_{\Delta_\theta} = 0$.*

Remark 7. *The congealed parameter ℓ_θ can be chosen according to the specific application [98]. Namely, it can be a nominal value for robust design or an extreme value to ensure sign definiteness, given that the resulting perturbation Δ_θ is considered consistently. Furthermore, ℓ_θ can be selected as a time-varying parameter satisfying typical assumptions used in the literature (e.g., $\dot{\ell}_\theta \in \mathcal{L}_\infty$ [68] or $\dot{\ell}_\theta \in \mathcal{L}_1$ [96]), and the congelation of variables method can be used to relax these assumptions [93].*

Remark 8. *The control law* (3.8) *depends on δ_{Δ_θ}, which is assumed to be known by Assumption 3. This assumption can be avoided by developing an online estimate for δ_{Δ_θ}. Indeed, since δ_{Δ_θ} is a constant and the control law is linearly parametrized, an estimate for δ_{Δ_θ} can be obtained using conventional adaptive control techniques. For instance, we can substitute the estimate $\hat{\delta}$ for δ_{Δ_θ} in equation* (3.8) *using the adaptation law*

$$\dot{\hat{\delta}} = \gamma_\delta \left(x^2 + x^4\right) \tag{3.10}$$

with $\gamma_\delta \in \mathbb{R}_{>0}$.

Proof. We consider the modified Lyapunov function candidate

$$V_\ell\left(x, \hat{\theta}, \ell_\theta, \hat{\delta}, \delta_{\Delta_\theta}\right) = \frac{1}{2}x^2 + \frac{1}{2\gamma_\theta}\left(\ell_\theta - \hat{\theta}\right)^2 + \frac{1}{2\gamma_\delta}\left(\delta_{\Delta_\theta} - \hat{\delta}\right)^2. \tag{3.11}$$

Taking the Lie derivative of V_ℓ along the trajectories of (3.1) yields

$$\begin{aligned}\dot{V}_\ell =& \hat{\theta}x^3 + ux + \left(\ell_\theta - \hat{\theta}\right)x^3 - \left(\ell_\theta - \hat{\theta}\right)\frac{\dot{\hat{\theta}}}{\gamma_\theta} - \left(\delta_{\Delta_\theta} - \hat{\delta}\right)\frac{\dot{\hat{\delta}}}{\gamma_\delta} + \Delta_\theta x^3 \\ \leq& \hat{\theta}x^3 + ux - \left(\delta_{\Delta_\theta} - \hat{\delta}\right)\frac{\dot{\hat{\delta}}}{\gamma_\delta} + \frac{1}{2\varepsilon}\delta_{\Delta_\theta}x^2 + \frac{1}{2}\varepsilon\delta_{\Delta_\theta}x^4\end{aligned} \tag{3.12}$$

Considering the following control law

$$u = -\left(k + \frac{1}{2\varepsilon}\hat{\delta}\right)x - \frac{1}{2}\varepsilon\hat{\delta}x^3 - \hat{\theta}x^2 \tag{3.13}$$

we obtain (assuming without loss of generality $\varepsilon = 1$)

$$\dot{V}_\ell \leq -kx^2 + \frac{1}{2}\left(\delta_{\Delta_\theta} - \hat{\delta}\right)(x^2 + x^4) - \left(\delta_{\Delta_\theta} - \hat{\delta}\right)\frac{\dot{\hat{\delta}}}{\gamma_\delta}\ . \tag{3.14}$$

Finally, substituting the adaptation law (3.10) we obtain:

$$\dot{V}_\ell \leq -kx^2\ . \tag{3.15}$$

Therefore, the closed-loop system trajectories are bounded, and invoking Barbalat's lemma leads to $\lim_{t\to\infty} x = 0$. □

3.4 *Congelation of variable* extension: Trajectory tracking

In this Section, we extend the *congelation of variable* method to the trajectory tracking problem leveraging the *Lyapunov redesign* approach [99][100, Section 14.2]. We consider the same scalar nonlinear system of the previous Section:

$$\dot{x} = \theta x^2 + u \ . \tag{3.16}$$

The control objective is now to track the desired trajectory $x_r \in C^1$. Defining the tracking error as

$$e = x - x_r \ , \tag{3.17}$$

we can write the error dynamics as

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_r \\ &= \theta x^2 + u - \dot{x}_r \\ &= \theta (e + x_r)^2 + u - \dot{x}_r \\ &= \theta e^2 + u - \dot{x}_r + \theta x_r^2 + 2\theta e x_r \ . \end{aligned} \tag{3.18}$$

After congealing θ, we consider the Lyapunov function candidate

$$V_\ell\left(e, \hat{\theta}, \ell_\theta\right) = \frac{1}{2}e^2 + \frac{1}{2\gamma_\theta}\left(\ell_\theta - \hat{\theta}\right)^2 . \tag{3.19}$$

Taking the Lie derivative of V_ℓ along the trajectories of (3.18) yields

$$\dot{V}_\ell = e\left(\theta e^2 + u - \dot{x}_r + \theta x_r^2 + 2\theta e x_r\right) - \left(\ell_\theta - \hat{\theta}\right)\frac{\dot{\hat{\theta}}}{\gamma_\theta} \tag{3.20}$$

Considering the parameter update law

$$\dot{\hat{\theta}} = \gamma_\theta e x^2 \ , \tag{3.21}$$

and the control law

$$u = -ke - \hat{\theta}x^2 + u_{rob} + \dot{x}_r \ , \tag{3.22}$$

with $u_{rob} \in \mathbb{R}$ being the robust control part, we obtain

$$\begin{aligned} \dot{V}_\ell &\leq -ke^2 + e\left(u_{rob} + \Delta_\theta x^2\right) \\ &\leq -ke^2 + eu_{rob} + |e|\delta_{\Delta_\theta} x^2 \ . \end{aligned} \tag{3.23}$$

Defining $u_{rob} = -\text{sign}(e)\delta_{\Delta_\theta}x^2$, we obtain:

$$\dot{V}_\ell \leq -ke^2 \; . \tag{3.24}$$

Remark 9. *The control part u_{rob} is designed to dominate the term $\Delta_\theta x^2$ which does not vanish asymptotically . Consequently, unlike the control design of the previous Section, adding a nonlinear damping term is not sufficient to guarantee asymptotic tracking.*

Remark 10. *The control part u_{rob} is a discontinuous function of the error e. This discontinuity generates both theoretical and practical concerns. Theoretically, we need to examine more carefully the question of existence and uniqueness of solutions, since the feedback function is not locally Lipschitz in e [101]. Practically, the implementation of such discontinuous controllers is characterized by the chattering phenomena that could excite the high-frequency dynamics of the system [102].*

Instead of trying to work out all the problems highlighted in Remark 10, we will choose the simpler and more practical way of approximating the discontinuous control law with a continuous law. The development of such approximation is reported in the next Section following an approach similar to that presented in [103].

3.4.1 Smooth modification

Considering the following smooth control law

$$u_{rob} = -\delta_{\Delta_\theta}\tanh(\lambda e)x^2 + u_{add} \tag{3.25}$$

with $\lambda \in \mathbb{R}_{>0}$ and u_{add} being an additional control term defined in the following, and substituting (3.25) into (3.23), we obtain

$$\begin{aligned}\dot{V}_\ell &\leq -ke^2 + e\,(\text{sign}(e) - \tanh(\lambda e))\,\delta_{\Delta_\theta}x^2 + eu_{add} \\ &\leq -ke^2 + \delta_{\Delta_\theta}|e|\text{sign}(e)\,(\text{sign}(e) - \tanh(\lambda e))\,x^2 + eu_{add} \\ &\leq -ke^2 + \delta_{\Delta_\theta}|e|\,(1 - \tanh(\lambda|e|))\,x^2 + eu_{add} \; .\end{aligned} \tag{3.26}$$

By isolating the second term of this inequality and using algebraic features of the hyperbolic tangent function, one can introduce an upper bound that depends only on the parameter λ (see [103] for more details). Specifically, for all real scalars $x \in \mathbb{R}$ and all non-zero real scalars $y \in \mathbb{R}$ we have:

$$0 \leq |x|(1 - \tanh|x/y|) \leq \alpha|y| \tag{3.27}$$

where $\alpha \in \mathbb{R}_{>0}$ is a constant with minimum value $\alpha^* = x^*(1 - \tanh x^*)$ for x^* satisfying $e^{-2x^*} + 1 - 2x^* = 0$. In [103], it was shown that $\alpha^* < \frac{1}{e_{log}}$, where e_{log} is the base of the natural logarithm. The property (3.27) yields:

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{1}{\lambda e_{log}}x^2\right) + eu_{add} \tag{3.28}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{1}{\lambda e_{log}}\right)(e^2 + 2ex_r + x_r^2) + eu_{add}\ . \tag{3.29}$$

Exploiting the Young's inequality[2], we obtain that

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{1}{\lambda e_{log}}\right)(e^2 + (e^2 + x_r^2) + x_r^2) + eu_{add} \tag{3.30}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)(e^2 + x_r^2) + eu_{add}\ . \tag{3.31}$$

Defining the additional control term as

$$u_{add} = -\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}e\ , \tag{3.32}$$

we obtain

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2\ . \tag{3.33}$$

From this bound it is clear that $\dot{V}_\ell < 0$ for all

$$|e| > \sqrt{\frac{2\delta_{\Delta_\theta}}{\lambda e_{log}k}x_r^2}\ . \tag{3.34}$$

[2] *Young's inequality*: If $a \geq 0$ and $b \geq 0$ are nonnegative real numbers and if $p > 1$ and $q > 1$ are real numbers such that $\frac{1}{p} + \frac{1}{q} = 1$, then $ab \leq \frac{a^p}{p} + \frac{b^q}{q}$. Equality holds if and only if $a^p = b^q$.

Whenever $|e|$ becomes large, the negative term "$-ke^2$ " in equation (3.31) dominates the remaining term, making $\dot{V}_\ell < 0$ and drawing the error toward the origin. When $|e|$ reaches sufficiently small values (determined by the parameter λ), the sign of V_ℓ is indefinite, and progress toward the origin cannot be guaranteed. This critical size represents the boundary of a residual error set to which all the closed-loop trajectories will eventually converge. Therefore, the error e is globally bounded.

Remark 11. *The asymptotic tracking is no more achieved using the smooth control law* (3.25). *However, it is important to underline that the boundary of the residual set* (3.34) *can be made smaller by increasing the λ parameter (for the limit case $\lambda \rightarrow \infty$ the asymptotic tracking is recovered). In practical terms, the λ parameter regulates the bandwidth in which we want to attenuate the perturbation introduced by the time-varying parameter. Furthermore, although the scheme has the capacity for infinite control input rates (in the limit as $\lambda \rightarrow \infty$), the control signal is only as sharp as it needs to be to counteract the time-varying perturbation $\Delta_\theta x^2$. Specifically, to offset perturbations with high-frequency components, the control input must be able to change at least as rapidly as the perturbation does, and the controller presented here (when properly tuned) does not introduce high-frequency signals unless they are already acting on the system through $\Delta_\theta x^2$. Finally, note that at no point in the stability proof is it assumed that the time-varying parameter θ has bounded derivative.*

3.4.2 Robustness modifications

The adaptation law (3.21) suffers from the drawback that the parameter $\hat{\theta}$ may depart arbitrarily away from its true value. To solve this issue, we describe in this Section two different *robustness modifications* of the parameter update law: a projection operator and a σ-modification scheme.

Projection modification:

If the true parameter is bounded by a known scalar constant, then the estimate can also be constrained to evolve within a bounded convex set with known bound [16]. To this end, we define two convex sets:

$$\Omega_{\ell_\theta} \triangleq \left\{ \ell_\theta \in \mathbb{R}^1 \mid \|\ell_\theta\|^2 < \varepsilon_1 \right\} , \tag{3.35}$$

$$\Omega_{\hat{\theta}} \triangleq \left\{ \hat{\theta} \in \mathbb{R}^1 \mid \|\hat{\theta}\|^2 < \varepsilon_1 + \delta_1 \right\} \tag{3.36}$$

with known $\varepsilon_1 \in \mathbb{R}_{>0}$ and $\delta_1 \in \mathbb{R}_{>0}$. Consider the following smooth projection scheme for $\hat{\theta}$:

$$\dot{\hat{\theta}} = \text{Proj}(\hat{\theta}, \Phi); \quad \text{with } \Phi \triangleq ex^2 \tag{3.37}$$

where

$$\text{Proj}(\hat{\theta},\Phi) \triangleq \begin{cases} \gamma_\theta \Phi & \text{if } C1)\ \|\hat{\theta}\|^2 < \varepsilon_1 \text{ or} \\ & \text{if } C2)\ \|\hat{\theta}\|^2 \geq \varepsilon_1 \wedge \Phi^T \hat{\theta} \leq 0 \\ \gamma_\theta \left(\Phi - \frac{(\|\hat{\theta}\|^2 - \varepsilon_1)\Phi^T \hat{\theta}}{\delta_1 \|\hat{\theta}\|^2} \hat{\theta} \right) & \text{if } C3)\ \|\hat{\theta}\|^2 \geq \varepsilon_1 \wedge \Phi^T \hat{\theta} > 0 \end{cases}$$

Note that this adaptation law is exactly equal to equation (3.21) in *cases C1* and *C2*. The projection operator $\text{Proj}(\hat{\theta},\Phi)$ is locally Lipschitz in $(\hat{\theta},\Phi)$ and switches smoothly between *cases C1-C3*. Furthermore, it can be shown that $\text{Proj}(\hat{\theta},\Phi)$ satisfies

$$\hat{\theta}(0) \in \Omega_{\hat{\theta}} \Rightarrow \hat{\theta}(t) \in \Omega_{\hat{\theta}} \tag{3.38}$$

for all $t \geq 0$ [16, Chapter 11]. Namely:

- In *case C1*, (3.38) readily holds because $\hat{\theta} \in \Omega_{\ell_\theta}$ and $\Omega_{\ell_\theta} \subset \Omega_{\hat{\theta}}$.
- In *case C2*, $\|\hat{\theta}\|^2$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{dt}} \|\hat{\theta}\|^2 = 2\hat{\theta}^T \dot{\hat{\theta}} = 2\gamma_\theta \hat{\theta}^T \Phi \tag{3.39}$$

  which is negative semidefinite by the conditions stated in *case C2*. Consequently, the estimate approaches the origin.

- Finally, for *case C3*,

$$\frac{\mathrm{d}}{\mathrm{dt}} \|\hat{\theta}\|^2 = 2\hat{\theta}^T \dot{\hat{\theta}} = 2\frac{\gamma_\theta}{\delta_1} \hat{\theta}^T \Phi \left(\delta_1 + \varepsilon_1 - \|\hat{\theta}\|^2 \right) \tag{3.40}$$

which decreases when $||\hat{\theta}||^2 > \varepsilon_1 + \delta_1$, increases if $||\hat{\theta}||^2 < \varepsilon_1 + \delta_1$, and is exactly zero when $||\hat{\theta}||^2 = \varepsilon_1 + \delta_1$. Thus, the projection scheme ensures that $\hat{\theta}$ remains in the set $\Omega_{\hat{\theta}}$.

Proof. Consider the Lyapunov function defined previously in (3.19). Taking the Lie derivative of V_ℓ along the closed-loop system trajectories yields:

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 - \frac{1}{\gamma}(\ell_\theta - \hat{\theta})\left(\dot{\hat{\theta}} - \gamma\Phi\right) \tag{3.41}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 + \dot{V}_2 \tag{3.42}$$

which has the same properties of (3.33) if

$$\dot{V}_2 = \frac{1}{\gamma_\theta}(\hat{\theta} - \ell_\theta)\left(\dot{\hat{\theta}} - \gamma_\theta\Phi\right) \leq 0 \tag{3.43}$$

If $\dot{\hat{\theta}}$ is prescribed according to the projection scheme (3.37), $\dot{V}_2 \leq 0$ is trivially satisfied for *cases C1* and *C2*. For *case C3*,

$$(\hat{\theta} - \ell_\theta)\left(\dot{\hat{\theta}} - \gamma_\theta\Phi\right) = (\hat{\theta} - \ell_\theta)\left(\gamma_\theta\left(\Phi - \frac{\left(||\hat{\theta}||^2 - \varepsilon_1\right)\Phi^T\hat{\theta}}{\delta_1||\hat{\theta}||^2}\hat{\theta}\right) - \gamma_\theta\Phi\right)$$

$$= -\gamma_\theta\left(\frac{\left(||\hat{\theta}||^2 - \varepsilon_1\right)\Phi^T\hat{\theta}}{\delta_1||\hat{\theta}||^2}\hat{\theta}\right) \leq 0 \tag{3.44}$$

which is true, because $(\hat{\theta} - \ell_\theta)\hat{\theta} = ||\hat{\theta}||^2 - \ell_\theta\hat{\theta} \geq 0$ when $||\hat{\theta}||^2 \geq \varepsilon_1$. Therefore,

$$\dot{V}_2 = \begin{cases} 0 & \text{in } \textit{case } 1 \text{ and } \textit{case } C2 \\ -\gamma_\theta\left(\frac{(||\hat{\theta}||^2 - \varepsilon_1)\Phi^T\hat{\theta}}{\delta_1||\hat{\theta}||^2}\tilde{\theta}^T\hat{\theta}\right) \leq 0 & \text{in } \textit{case } C3 \end{cases} \tag{3.45}$$

from which it follows that $\dot{V}_2 \leq 0$. □

σ-modification:

In the previous Section, a prior knowledge of the convex set Ω_{ℓ_θ} is assumed. In [104], Ioannou and Kokotovic developed the σ-modification scheme that does not require any prior information on the system uncertainty domain. The adaptive law with the σ-modification is defined as:

$$\dot{\hat{\theta}} = \gamma_\theta e x^2 - \sigma\hat{\theta} \tag{3.46}$$

where $\sigma \in \mathbb{R}_{>0}$ is a strictly positive constant. In essence, this modification adds damping to the ideal adaptive law (3.21). Consider again the Lyapunov function V_ℓ defined previously in (3.19). Evaluating the Lie derivative along the closed-loop system trajectories (3.18) yields:

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 - \frac{1}{\gamma_\theta}(\ell_\theta - \hat{\theta})\left(\dot{\hat{\theta}} - \gamma_\theta\Phi\right) \tag{3.47}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 - \frac{1}{\gamma_\theta}(\ell_\theta - \hat{\theta})\left(-\sigma\hat{\theta}\right) \tag{3.48}$$

Defining $\tilde{\theta} = \ell_\theta - \hat{\theta}$ and using $\hat{\theta} = \ell_\theta - \tilde{\theta}$, we obtain:

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 + \frac{\sigma}{\gamma_\theta}\left(\tilde{\theta}\ell_\theta - \tilde{\theta}^2\right) \tag{3.49}$$

Using $2ab \leq a^2 + b^2$ for any a and b, we obtain:

$$\dot{V}_\ell \leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 + \frac{\sigma}{\gamma_\theta}\left(\frac{1}{2}\left(\tilde{\theta}^2 + \ell_\theta^2\right) - \tilde{\theta}^2\right) \tag{3.50}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 + \frac{\sigma}{2\gamma_\theta}\left(\ell_\theta^2 - \tilde{\theta}^2\right) \tag{3.51}$$

$$\leq -ke^2 + \left(\delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right)x_r^2 + \frac{\sigma}{2\gamma_\theta}\ell_\theta^2 \,. \tag{3.52}$$

Hence, $\dot{V}_\ell < 0$ for all

$$e > \sqrt{\frac{2\delta_{\Delta_\theta}}{\lambda e_{log}}x_r^2 + \frac{\sigma}{2\gamma_\theta}\ell_\theta^2}\,.$$

As expected, the σ-modification enlarges the residual set that is now dependent also on the “size” of the uncertain parameter ℓ_θ.

Remark 12. *The σ-modification has drawbacks in terms of performance [9, 16]. The adaptation law* (3.46) *can be approximated as $\dot{\hat{\theta}} \approx -\sigma\hat{\theta}$ when the tracking error is near the origin ($e \approx 0$). Therefore, for small tracking errors, the adaptive parameter θ tends to the origin, i.e., the algorithm unlearns the gains. The e-modification was introduced by Narendra and Annaswamy to combat these negative effects [34] substituting a term proportional to the tracking error for the constant damping gain σ in* (3.46)*. The justification for using an error-dependent damping is that as the tracking error decreases, the damping term tends to zero [16].*

3.4.3 Adaptive modification

The control law (3.25) depends on δ_{Δ_θ}, which was assumed to be known. Even if δ_{Δ_θ} is unknown, we can substitute an online estimate $\hat{\delta}$ for δ_{Δ_θ}. Namely, being the bound δ_{Δ_θ} constant and the control law (3.25) linearly parametrized, we can use the following adaptation law:

$$\dot{\hat{\delta}} = \gamma_\delta e\left(\frac{2}{\lambda e_{log}} e + x^2 \tanh(\lambda e)\right) \quad \text{with } \gamma_\delta \in \mathbb{R}_{>0} \, . \tag{3.53}$$

Proof. We consider the modified Lyapunov function candidate

$$V_\ell\left(e, \hat{\theta}, \ell_\theta, \hat{\delta}, \delta_{\Delta_\theta}\right) = \frac{1}{2}e^2 + \frac{1}{2\gamma_\theta}\left(\ell_\theta - \hat{\theta}\right)^2 + \frac{1}{2\gamma_\delta}\left(\delta_{\Delta_\theta} - \hat{\delta}\right)^2 . \tag{3.54}$$

Taking the Lie derivative of V_ℓ along the trajectories of (3.54) and substituting the adaptation law (3.21) and the following control law

$$u = -ke - \hat{\theta}x^2 - \hat{\delta}\tanh(\lambda e)x^2 - \hat{\delta}\frac{2}{\lambda e_{log}} + \dot{x}_r \, , \tag{3.55}$$

we obtain

$$\begin{aligned} \dot{V}_\ell &\leq -ke^2 + eu_{rob} - \left(\delta_{\Delta_\theta} - \hat{\delta}\right)\frac{\dot{\hat{\delta}}}{\gamma_\delta} + |e|\delta_{\Delta_\theta}x^2 \\ &\leq -ke^2 + e\left(\delta_{\Delta_\theta}\operatorname{sign}(e) - \hat{\delta}\tanh(\lambda e)\right)x^2 - e\hat{\delta}\frac{2}{\lambda e_{log}} + \\ &\quad - \left(\delta_{\Delta_\theta} - \hat{\delta}\right)\frac{\dot{\hat{\delta}}}{\gamma_\delta} \pm e\left(\delta_{\Delta_\theta}\tanh(\lambda e)x^2 + \delta_{\Delta_\theta}\frac{2}{\lambda e_{log}}\right) . \end{aligned} \tag{3.56}$$

Using the adaptation law (3.53) we obtain again $\dot{V}_\ell$ as in (3.26). Therefore we can conclude boundedness of all trajectories of the closed-loop system using the same argument as Section 3.4.1. □

Remark 13. *The parameter update law in* (3.53) *suffers from the drawback that the parameter* $\hat{\delta}$ *can drift arbitrarily away from its respective true value. To solve this issue, we can use the robustness modifications explained in the previous Section with similar considerations.*

3.5 Simulation results

In this Section, we compare in simulation the proposed approach with the standard adaptive control scheme (without robustness term) with two different robustness modifications: the projection operator and the σ-modification. Specifically, we compare three state feedback controllers:

- *Controller A* is the proposed controller with the control law (3.55), and the update law (3.37) for $\hat{\theta}$ and (3.53) for $\hat{\delta}$ with the projection operator modification;
- *Controller B* uses the control law (3.22) with $u_{rob} = 0$ and the update law for $\hat{\theta}$ (3.37) with the projection operator modification;
- *Controller C* is identical to *Controller B* but also includes the σ-modification in the update law (3.21) .

We consider the same scalar nonlinear system of the previous Sections:

$$\dot{x} = \theta x^2 + u \ , \tag{3.57}$$

and the control objective is to track a given reference trajectory $x_r(t) = \sin(0.1t) + 0.1\sin(t)$. To consider an open-loop unstable plant we set $\ell_\theta = 2$, *i.e.*, the mean of θ equal to 2. Namely, the time-varying parameter θ evolves as shown in Figure 3.3. As in [88], to show that the proposed approach does not require any restriction on θ, the parameter variation intentionally includes a fast varying component (repeating sequence stair with sampling time equal to 0.1 s) to remove the restriction on $|\dot{\theta}|$, and a slowly but persistently varying components (two harmonics) to remove

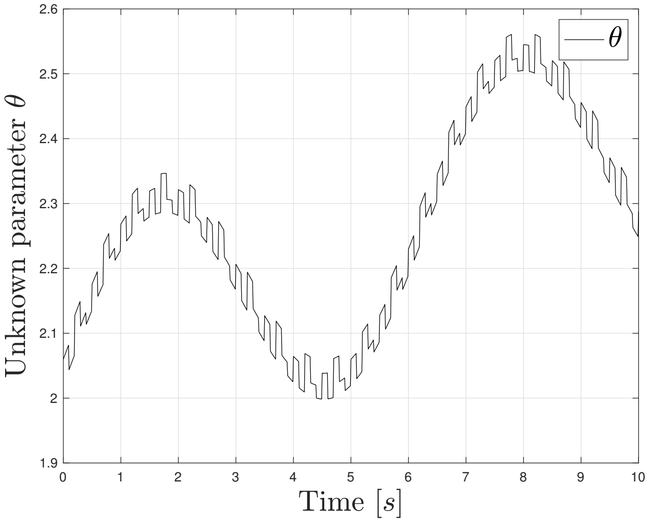


Figure 3.3: *Time-varying parameter evolution.*

the restrictions on the integral of $|\dot{\theta}|$. The parameters used in the simulation are $x(0) = 0$, $\hat{\theta}(0) = 0$, $k = 0.01$ and $\gamma_\theta = 100$ for all the adaptive controllers, $\hat{\delta}(0) = 0$, $\lambda = 10$ and $\gamma_\delta = 100$ for the *Controller A*, and $\sigma = 0.1$ for the *Controller C*. Simulating the three closed-loop systems yields the results listed in Table 3.1 in terms of error norms. The time evolutions of the error and input are shown in Figure 3.4 and Figure 3.5, respectively. We can state that the proposed controller shows better performance in terms of smaller overshoot at the beginning, smoother response, faster rate of convergence, and smaller steady-state error. These effects are due to the robustness term u_{rob} defined in equation (3.55) which dominates the perturbation introduced by the time-varying parameter Δ_θ.

Table 3.1: *Error norms with the different controllers.*

	Error 2-norm $\|e\|_2$	Error ∞-norm $\|e\|_\infty$
Controller A	1.58	0.17
Controller B	15.79	0.25
Controller C	7.30	0.23

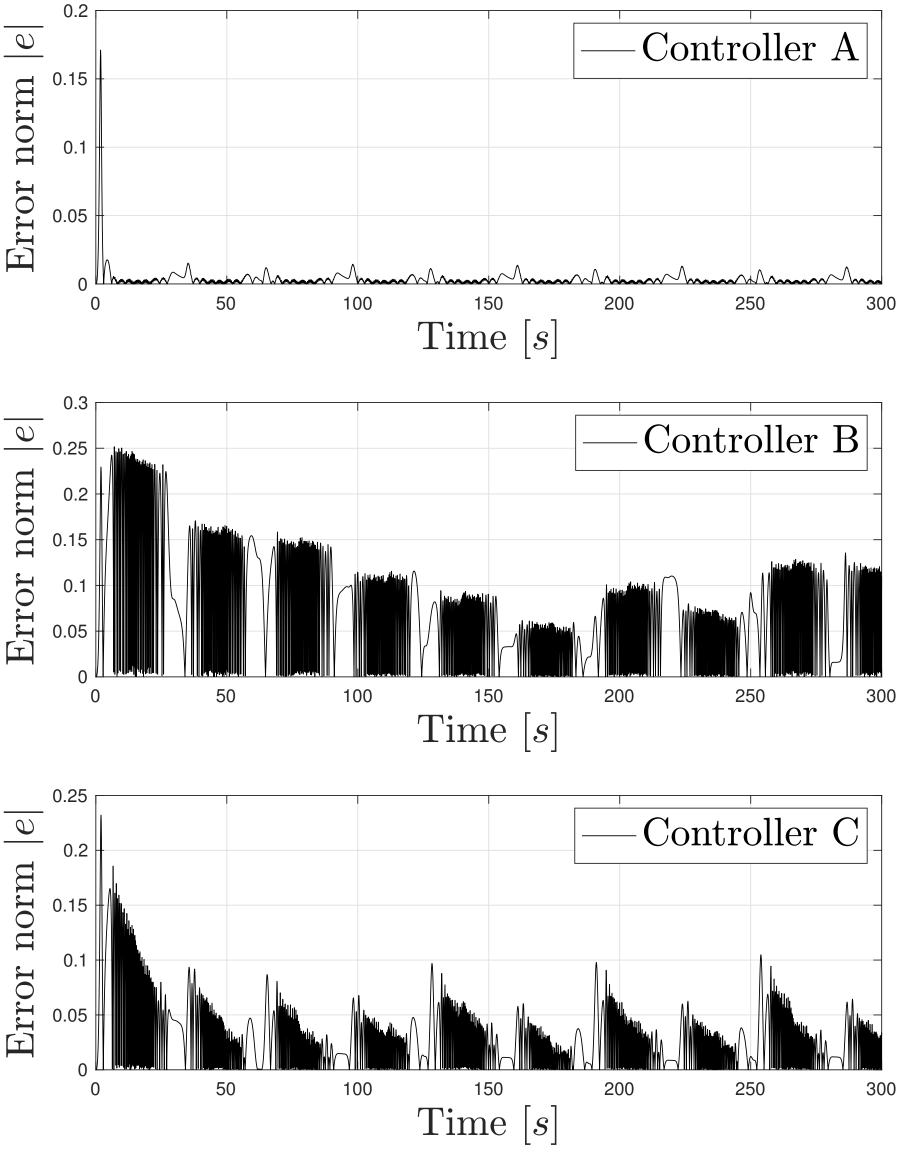


Figure 3.4: *Time evolution of the error.*

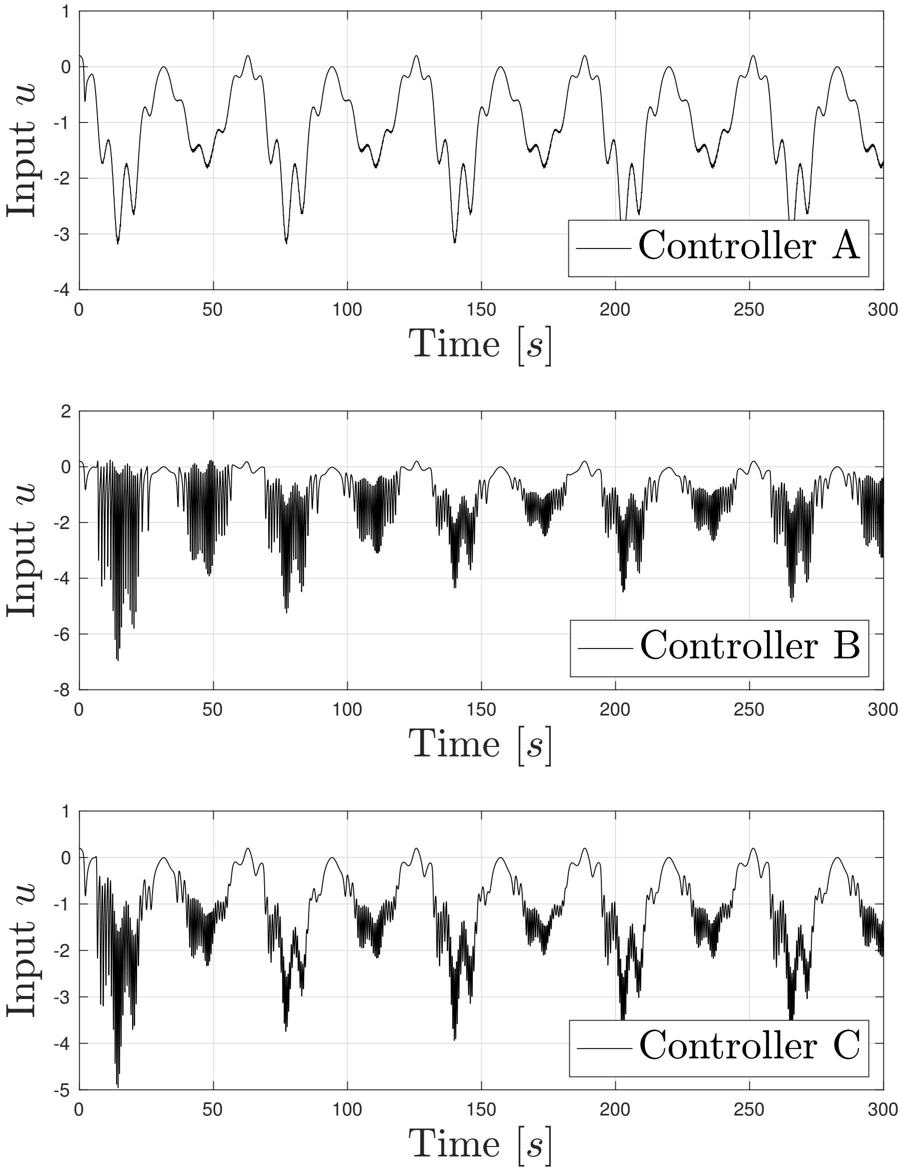


Figure 3.5: *Time evolution of the input.*

The smooth modification proposed in Section 3.4.1 limits the input bandwidth resulting in a smoother input for *Controller A*. This effect can be seen in Figure 3.6, where the amplitude of the input spectrum is shown. In fact, the proposed controller does not excite high-frequency dynamics.

Remark 14. *The standard adaptive schemes (Controller B and C) need a large adaptation rate to "follow" the evolution of θ, and introducing the σ-modification improves the performance by acting as a forgetting factor that accelerates the parameter learning. On the other hand, the proposed controller can run with a lower adaptation rate obtaining a similar steady-state response and reducing the oscillations introduced by the adaptive feedforward term. In this numerical example, we use $\gamma_\theta = 100$ in all the adaptive schemes to compare fairly the performance achieved.*

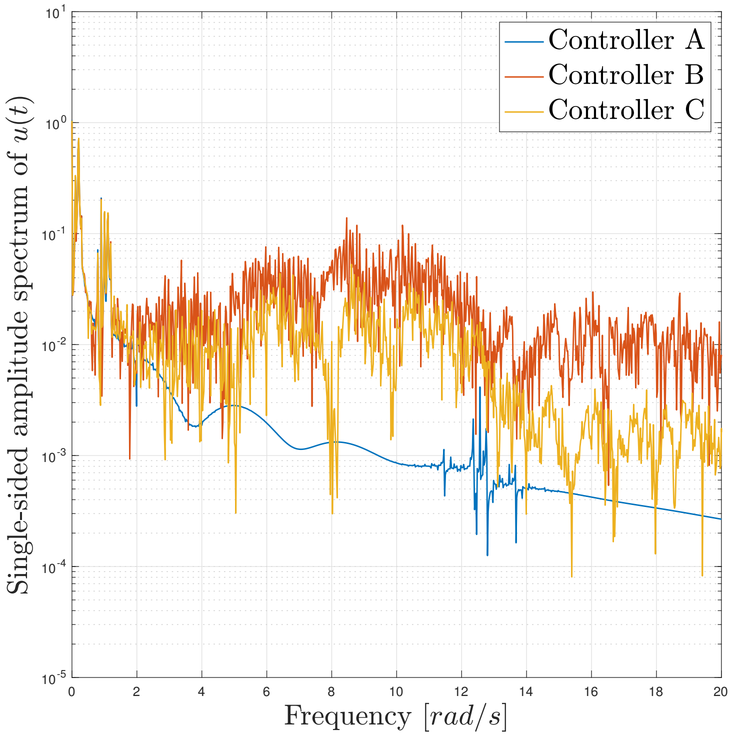


Figure 3.6: *Single-sided amplitude spectrum of the input.*

CHAPTER 4

Attitude control with uncertain time-varying inertia parameters

There are many interesting aerospace systems that exhibit time-varying (potentially nonlinear) characteristics [105]. Examples of such systems are a UAV that has delivered or received a payload [106], a spacecraft with variable mass distribution due to on-orbit refuelling [107], and an aircraft that has deployed munitions [108]. This Chapter focuses on the attitude control problem of a spacecraft affected by time-varying inertia variation. Although adaptive control schemes for spacecraft attitude control are numerous in the literature, few have been designed to ensure consistent performance for a spacecraft with both rigid and non-rigid (time-varying) inertia components. Because inertia variations are common due to phenomena like fuel depletion or mass displacement in a deployable spacecraft, an adaptive control algorithm that takes explicit account of such information is of significant interest. In view of this, we propose novel adaptive attitude control schemes based on the *congelation of variables* method presented in the previous Chapter. Detailed derivations of the control laws are provided, along with a thorough analysis of the associated stability and error convergence properties. In addition, numerical simulations are presented to highlight the performance benefits compared with classical control schemes that do not account for inertia variations.

4.1 Introduction

Spacecraft attitude control remains a nonlinear control problem of great practical and intellectual importance [109, 110]. Despite the wide range of techniques available, developing an attitude control system is a labor-intensive and time-consuming process [111]. For applications where spacecraft must be launched at short notice, it is attractive to employ control algorithms that are robust to uncertainty, such as inaccurate knowledge of spacecraft mass distribution, errors in sensor and actuator alignment, and measurement errors. This Chapter addresses an additional complication in spacecraft attitude control: the spacecraft mass distribution is not only uncertain but also time-varying. Indeed, the increasing requirements of satellite functions result in appreciable changes in inertia during on-orbit operation. For example, an expanding solar array or appendage movements [112] cause noticeable changes in inertia, especially for small satellites [113]. A spacecraft may also have rotating components, such as a reflector or antenna that rotates relative to the spacecraft bus (see Chapter 6 as an example). In addition, a capture or docking mission can significantly and abruptly increase the mass of the satellite [114]. Therefore, it is relevant to study the attitude control of satellites by taking into account significant and fast changes in inertia.

Different control strategies for satellite attitude control with inertia uncertainties have been proposed in the literature. For instance, in the last decade, several adaptive control [115, 116, 117, 118, 119] and sliding mode control [120] techniques have been developed. To improve performance in repetitive tasks, iterative learning control has been developed in [121]. Utilizing a minimal kinematics approach, robust optimal attitude control with uncertain inertia was proposed in [122]. Studies on fault-tolerant controllers that take into account also external disturbance have been conducted in [123]. These works assume the unknown parameter constant and rely on robustness modification techniques and high-gain learning rates, because they do not explicitly account for time variations arising in the unknown parameters. As a consequence, the aforementioned controllers can be appropriate if the inertia changes slowly. On the other hand, if the inertia varies rapidly over a wide range, the system performance will be deteriorated and instability can occur. Therefore, some researchers have considered time-varying inertia explicitly during the controller design. Jin *et al.* developed a gain-scheduled controller to guarantee the steady-state and transient performance of the system in [124]. Weiss *et al.* presented a solution approach in [125] assuming that changes in the inertia matrix occur as a result of a known purely time-dependent dynamic phenomenon. Similarly, Thakur *et al.* proposed a control method capable of handling not only time-dependent but also a combination of time- and state-dependent inertial parameters in [126]. The contributions of this Chapter are summarized in the following:

- An extension of the controller for *motion-to-rest* manoeuvres presented in [125] that does not require knowledge of the bound of inertia time derivative.

- A development of a novel adaptive controller for *motion-to-motion* manoeuvres based on the *congelation of variables* method that does not assume that the time-variation of the time-varying component of the inertia is known as in [125, 126]

Remark 15. *We design different controllers for motion-to-rest and motion-to-motion manoeuvres exploiting the considerations done in [115]. In particular, Sanyal et al. demonstrated that for motion-to-rest manoeuvrers, e.g., detumbling tasks, in the absence of disturbance torques, no knowledge of the inertia matrix is needed, hence no estimates of the inertia matrix have to be constructed. On the other hand, for motion-to-motion manoeuvrers, e.g., spinning tasks, the control law needs an estimate of the inertia matrix. However, this estimate need not converge to the actual inertia matrix to achieve the control objective.*

4.2 Model formulation

As a spacecraft model, we consider a fully actuated non-rigid body without onboard momentum storage. We examine only the rotational motion and not the translational motion of the spacecraft center of mass. Therefore, we take into account only the torque applied by the force or torque actuators. We assume that a body-fixed frame is defined for the spacecraft, whose origin is chosen to be the center of mass, and that an inertial frame is specified for determining the attitude of the spacecraft. The attitude dynamics of the non-rigid body are governed by the following rotational equations of motion [126]:

$$J(t)\dot{\omega}(t) = -\dot{J}(t)\omega(t) - S(\omega(t))J(t)\omega(t) + u(t) \tag{4.1}$$

where $\omega(t) \in \mathbb{R}^3$ is the angular velocity of the spacecraft frame with respect to the inertial frame resolved in the spacecraft frame, $J(t) \in \mathbb{R}^{3\times 3}_{>0}$ is the time-varying inertia matrix of the spacecraft, $u(t) \in \mathbb{R}^3$ is the external control torque, and $S(\cdot)$ is the matrix representation of the linear cross-product operation such that for any $a, b \in \mathbb{R}^3$ $S(a)b = a \times b$. Note that, for notational convenience, the time argument t is hereafter left out, unless specifically stated for clarification or emphasis.

Remark 16. *For the sake of conciseness, we neglect the disturbance effects, i.e., all internal and external torques acting on the spacecraft aside from control torques, which may be caused by, e.g., onboard components, gravity gradients, solar pressure, atmospheric drag, or magnetic field [111]. This approximation is made without loss of generality. Indeed, the proposed control scheme, for instance, can be extended to counteract disturbances that are linear combinations of constant and harmonic signals for which the frequencies are known, but their amplitudes and phases are unknown (see [115] for more details).*

Remark 17. *Compared to the rigid body case [111], the time-varying inertia complicates the dynamic equations due to the term "$-\dot{J}\omega$" added to equation* (4.1). *Note that this term affects only the attitude dynamics when the spacecraft has a non-zero angular velocity [125].*

4.3 Detumbling control (motion-to-rest)

In this Section, we consider the problem of designing a feedback law to detumble a fully actuated non-rigid body controlled by force or torque actuators, such as thrusters or magnetic torque devices, and without onboard momentum storage. The objective of the proposed control law is to reduce the angular rate of the spacecraft from the initial value $\omega(0) = \omega_0 \in \mathbb{R}^3$ to zero. It can be formalized as follows.

Problem 1. *Considering the dynamics in* (4.1)*, design a state-feedback controller delivering a control torque $u \in \mathbb{R}^3$ such that the angular velocity $\omega(t)$ converges asymptotically to the origin ($\omega = 0$).*

In the following, we describe the derivation of the controller presented in [125] (referred to as *fixed-gain controller*). Then, we propose an adaptive modification that overcomes the limitations of the previous controller.

4.3.1 Fixed-gain controller

We define the quadratic Lyapunov function candidate:

$$V(\omega) = \frac{1}{2}\omega^{\mathrm{T}} J \omega. \tag{4.2}$$

Taking the Lie derivative along the trajectories of (4.1) yields

$$\begin{align}
\dot{V}(\omega) &= \omega^{\mathrm{T}} J \dot{\omega} + \frac{1}{2}\omega^{\mathrm{T}} \dot{J} \omega \tag{4.3}\\
&= \omega^{\mathrm{T}} \left(J\omega \times \omega + u - \dot{J}\omega + \frac{1}{2}\dot{J}\omega \right) \tag{4.4}
\end{align}$$

Let the control input given by a proportional controller:

$$u = -K_{\mathrm{v}} \omega \ , \tag{4.5}$$

where $K_{\mathrm{v}} \in \mathbb{R}^{3\times 3}_{>0}$ is the proportional gain. We obtain

$$\begin{align}
\dot{V}(\omega) &= \omega^{\mathrm{T}} \left(-K_{\mathrm{v}}\omega - \frac{1}{2}\dot{J}\omega \right) \tag{4.6}\\
&= -\,\omega^{\mathrm{T}} \left(K_{\mathrm{v}} + \frac{1}{2}\dot{J} \right) \omega. \tag{4.7}
\end{align}$$

Selecting $K_\mathrm{v} > -\frac{1}{2}\dot{J} + \varepsilon I$ for some $\varepsilon \in \mathbb{R}_{>0}$ ensures that Lyapunov candidate function decays, *i.e.*, $\dot{V}(\omega) < 0$. Hence, the stabilization of the angular velocity is achieved. The control law (4.5) achieves zero steady-state error without integral action and knowledge of the inertia matrix J.

Remark 18. *As highlighted in [125], beside the constraint $K_\mathrm{v} > -\frac{1}{2}\dot{J} + \varepsilon I$, the controller presented in [115] with fixed parameters requires no modification for the case of time-varying inertia. This condition is automatically satisfied when the inertia matrix is increasing, i.e., $J(t_1) \leq J(t_2)$ for all $t_1 \leq t_2$, which implies that $\dot{J} \geq 0$. Thus, for every positive-definite choice of K_v, it follows that $K_\mathrm{v} > -\frac{1}{2}\dot{J} + \varepsilon I$ for some $\varepsilon \in \mathbb{R}_{>0}$. Conversely, when the inertia matrix is decreasing, e.g. during the retraction of a solar panel, a bound on $\dot{J}$ must be known to properly select the gain K_v [125].*

Advantages:

- Fixed-gain controller with simple structure (only proportional term).

Disadvantages:

- A bound on the time derivative $\dot{J}$ must be known to ensure stability during retraction manoeuvrer;
- The tuning of the proportional controller gain K_v depends on the worst-case scenario ($\max_{t>0} |\dot{J}|$). Hence the controller could exhibit the problems arising with high-gain controllers [127].

4.3.2 Proposed adaptive controller

The controller gain K_v of the control law (4.5) depends on $\max_{t>0} |\dot{J}|$, which is assumed to be known. Even if $\max_{t>0} |\dot{J}|$ is unknown, one can easily overcome this by building an estimate via classical adaptive control techniques, since $\max_{t>0} |\dot{J}|$ is a constant and the control law is linearly parameterized. Specifically, let $\hat{\sigma} \in \mathbb{R}$ denote an estimate of $\sigma = \max_{t>0} \sqrt{\lambda_{\max}\left(\dot{J}^* \dot{J}\right)}$ that is the largest singular value of $\dot{J}$, where $\dot{J}^*$ denotes the conjugate transpose of $\dot{J}$. We define the Lyapunov candidate

$$V(\omega, \sigma, \hat{\sigma}) = \frac{1}{2}\omega^\mathrm{T} J \omega + \frac{1}{2\gamma}\tilde{\sigma}^2 , \tag{4.8}$$

where $\gamma \in \mathbb{R}_{>0}$ is the adaptation rate, and $\tilde{\sigma} := (\sigma - \hat{\sigma})$ is the estimation error. Taking the Lie derivative of (4.8) along the trajectories (4.1) yields

$$\begin{aligned}
\dot{V}(\omega) &= \omega^{\mathrm{T}} J \dot{\omega} + \frac{1}{2}\omega^{\mathrm{T}} \dot{J} \omega - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \\
&= \omega^{\mathrm{T}} \left(J\omega \times \omega + u - \dot{J}\omega + \frac{1}{2}\dot{J}\omega \right) - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \\
&= \omega^{\mathrm{T}} \left(J\omega \times \omega + u - \frac{1}{2}\dot{J}\omega \right) - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \\
&\leq \omega^{\mathrm{T}} \left(J\omega \times \omega + u + \frac{1}{2}\sigma\omega \right) - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \qquad (4.9)
\end{aligned}$$

Considering the control input given by

$$u = -K_{\mathrm{v}}\omega - \frac{\hat{\sigma}}{2}\omega \, , \qquad (4.10)$$

where $K_{\mathrm{v}} \in \mathbb{R}^{3\times 3}_{>0}$ is the proportional gain, we obtain

$$\begin{aligned}
\dot{V}(\omega) \leq & -\omega^{\mathrm{T}} K_{\mathrm{v}} \omega + \frac{1}{2}(\sigma - \hat{\sigma})\omega^{T}\omega - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \\
\leq & -\omega^{\mathrm{T}} K_{\mathrm{v}} \omega + \frac{1}{2}\tilde{\sigma}\omega^{T}\omega - \frac{1}{\gamma}\dot{\hat{\sigma}}\tilde{\sigma} \, . \qquad (4.11)
\end{aligned}$$

Considering the following adaptation law[1]

$$\dot{\hat{\sigma}} = \gamma\omega^{T}\omega \, , \qquad (4.12)$$

we obtain

$$\dot{V}(\omega) \leq -\omega^{\mathrm{T}} K_{\mathrm{v}} \omega \, , \qquad (4.13)$$

hence $\dot{V}$ is negative semi-definite. Since $V \geq 0$ and $\dot{V} \leq 0, V$ is a monotonic function indicating that $V(t) \leq V(0)$. Consequently, all closed-loop signals are bounded. Furthermore, $\int_0^t \dot{V}(t)\mathrm{d}t$ exists and is finite, which implies that $\omega \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ and, consequently, it follows that $\dot{\omega} \in \mathcal{L}_\infty$ from equation (4.1). Invoking Barbalat's lemma leads to $\lim_{t\to\infty} \omega = 0$.

[1] The estimated parameter $\hat{\sigma}$ increases over time ($\dot{\hat{\sigma}} \geq 0 \; \forall \; t$). A robustness modification should be introduced in real-world applications (acting as forgetting factor).

Remark 19. *The proposed control scheme can be modified to counteract disturbances that are linear combinations of constant and harmonic signals for which the frequencies are known, but their amplitudes and phases are unknown (see [115] for more details). Furthermore, the control law can be modified (as in [125]) to achieve almost global stabilization of a constant desired attitude configuration.*

Advantages:

- No bound on the inertia time derivative $\dot{J}$ must be known;
- The tuning of the proportional gain K_v does not depend on $\dot{J}$. Indeed, the perturbation term $\dot{J}\omega$ is dominated by the adaptive part of the controller $(-\frac{\hat{\sigma}}{2}\omega)$ that acts as a nonlinear damping term.

Disadvantages:

- The proposed controller involves an additional state $\hat{\sigma}$.

4.4 Angular velocity tracking (motion-to-motion)

The objective of this Section is to introduce an adaptive controller guaranteeing angular velocity tracking for a fully actuated non-rigid body with an unknown time-varying inertia matrix. For such a problem, we assume the following standard assumption for the desired attitude trajectory that should be tracked.

Assumption 4. *Smoothness and boundedness of the desired trajectory: The desired trajectory $t \mapsto \omega_d(t) \in \mathbb{R}^3$ is continuously differentiable and uniformly bounded.*

The state feedback dynamic angular velocity tracking problem for a fully actuated non-rigid body can be then formalized as follows.

Problem 2. *Considering the dynamics in* (4.1) *and given a desired trajectory $t \mapsto \omega_d(t)$ satisfying Assumption 4, design a state-feedback dynamic controller delivering a control torque $u \in \mathbb{R}^3$ such that the closed-loop trajectories $(\omega - \omega_d)$ are uniformly ultimately bounded.*

4.4.1 Error dynamics

Defining the tracking error as

$$e(t) = \omega - \omega_d \ , \tag{4.14}$$

we can express the error dynamics as

$$\begin{aligned} \dot{e} &= \dot{\omega} - \dot{\omega}_d \\ &= J^{-1} \left(u + S(J\omega)\ \omega - J\ \dot{\omega}_d - \dot{J}\omega \right) \ . \end{aligned} \tag{4.15}$$

In this work, we consider that the uncertainty is due only to uncertain inertia parameters J and their derivatives $\dot{J}$. Exploiting the symmetry of inertia matrix, the uncertain inertial terms can be parametrized by $\theta = [J_{11}, J_{12}, J_{13}, J_{22}, J_{23}, J_{33}]^T \in \mathbb{R}^6$ that contains the six unique parameters of J. Specifically, a regressor matrix is defined through the following affine representation:

$$W(\omega, \dot{\omega}_d)\ \theta = S(J\omega)\ \omega - J\ \dot{\omega}_d \ . \tag{4.16}$$

Benefiting by this property, W can be obtained by taking Jacobian for the right-hand side of equation (4.16) with respect to θ. Exploiting the affine representation, we can rewrite equation (4.15) as[2]

$$\dot{e} = J^{-1} \left(u + W(\omega, \dot{\omega}_d)\ \theta - \dot{J}\omega \right) \ . \tag{4.17}$$

4.4.2 Adaptive control law design

After congealing θ, we consider the Lyapunov function candidate:

$$V_\ell \left(e, \hat{\theta}, \ell_\theta \right) = \frac{1}{2} e^\top J e + \frac{1}{2} \left(\ell_\theta - \hat{\theta} \right)^\top \Gamma_\theta^{-1} \left(\ell_\theta - \hat{\theta} \right) \ , \tag{4.18}$$

where the definite positive matrix $\Gamma_\theta \in \mathbb{R}^{6\times 6}_{>0}$ is the adaptation rate. Taking the Lie derivative of (4.18) along the trajectories given by (4.15) yields

$$\dot{V}_\ell = e^\top \left(u + W\ \theta + \frac{1}{2} \dot{J}\omega \right) - \left(\ell_\theta - \hat{\theta} \right)^\top \Gamma_\theta^{-1} \dot{\hat{\theta}} \ . \tag{4.19}$$

[2] Note that the $(\omega, \dot{\omega}_d)$ argument is hereafter left out for notational convenience.

We consider the control input $u \in \mathbb{R}^3$ given by

$$u = u_p + u_{ad} + u_{rob} \tag{4.20}$$

where $u_p = -ke \in \mathbb{R}^3$ regulates the desired error dynamics with $k \in \mathbb{R}_{>0}$ being the proportional gain, $u_{ad} \in \mathbb{R}^3$ is the adaptive part, and $u_{rob} \in \mathbb{R}^3$ is the robust part that dominates the perturbation generated by the time-varying inertia. Because the term involving ℓ_θ is unknown and cannot be directly cancelled by u_{ad}, the adaptive control part is designed using parameter estimate $\hat{\theta}$. Hence, we consider the adaptive control part as

$$u_{ad} = -W\ \hat{\theta}\ . \tag{4.21}$$

Substituting (4.20) into (4.19) yields

$$\begin{aligned} \dot{V}_\ell &\leq e^\top \left(-ke + u_{rob} + W\ (\theta - \hat{\theta}) + \frac{1}{2}\dot{J}\omega\right) - \left(\ell_\theta - \hat{\theta}\right)^\top \Gamma_\theta^{-1}\dot{\hat{\theta}} \\ &\leq e^\top \left(-ke + u_{rob} + W\ (\ell_\theta + \Delta_\theta - \hat{\theta}) + \frac{1}{2}\dot{J}\omega\right) + \\ &\qquad - \left(\ell_\theta - \hat{\theta}\right)^\top \Gamma_\theta^{-1}\dot{\hat{\theta}}\ . \end{aligned} \tag{4.22}$$

Considering the parameter update law

$$\dot{\hat{\theta}} = \ \Gamma_\theta W\ e\ , \tag{4.23}$$

we obtain

$$\dot{V}_\ell \leq e^\top \left(-ke + u_{rob} + W\ \Delta_\theta + \frac{1}{2}\dot{J}\omega\right)\ . \tag{4.24}$$

Remark 20. *In this work, we do not estimate the inertia derivative parameters $\dot{J}$ to reduce the dimensions of the adaptation law. Indeed, assuming that the mean of $\dot{J}$ with respect time is almost zero, adding a feedforward adaptive term (based on the congealed $\dot{J}$) that cancels out the term $\dot{J}\omega$ is pointless. This choice is done without loss of generality, indeed the proposed controller can estimate also the inertia derivative parameters $\dot{J}$ considering $\theta = [J_{11}, J_{12}, J_{13}, J_{22}, J_{23}, J_{33}, \dot{J}_{11}, \dot{J}_{12}, \dot{J}_{13}, \dot{J}_{22}, \dot{J}_{23}, \dot{J}_{33}]^T \in \mathbb{R}^{12}$.*

Remark 21. *The closed-loop dynamics is perturbed not only by the time-varying term due to the parameters change Δ_θ (as in Section 3.3.1), but also by the term $\dot{J}\omega$ that can prevent $\dot{V}_\ell$ to be semi-definite negative.*

To design the robustness term u_{rob}, we expand the term $W\Delta_\theta$ by exploiting the "physics" of the problem and recalling $J = \ell_J + \Delta_J$:

$$\begin{aligned} \dot{V}_\ell &\leq e^\top \left(-ke + u_{rob} + S(\Delta_J\ \omega)\ \omega - \Delta_J\ \dot{\omega}_d + \frac{1}{2}\dot{J}\omega\right) \\ &\leq -ke^\top e + e^\top \left(u_{rob} + S(\Delta_J\ (\omega_d + e))\ (\omega_d + e) - \Delta_J\ \dot{\omega}_d + \frac{1}{2}\dot{J}\omega\right) \\ &\leq -ke^\top e + e^\top \left(u_{rob} + S(\Delta_J\ (\omega_d + e))\ \omega_d - \Delta_J\ \dot{\omega}_d + \frac{1}{2}\dot{J}\omega\right) \qquad (4.25) \end{aligned}$$

Now we introduce the following definitions to find an upper bound of $\dot{V}_\ell$.

Definition. *Absolute value of vector and matrix. Given $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$, the absolute value is defined as*

$$|x| = \begin{bmatrix} |x_0| \\ |x_1| \\ \vdots \\ |x_{n-1}| \end{bmatrix} \quad \textit{and} \quad |A| = \begin{bmatrix} |A_{0,0}| & |A_{0,1}| & \cdots & |A_{0,n-1}| \\ |A_{1,0}| & |A_{1,1}| & \cdots & |A_{1,n-1}| \\ \vdots & \vdots & \ddots & \vdots \\ |A_{m-1,0}| & |A_{m-1,1}| & \cdots & |A_{m-1,n-1}| \end{bmatrix}.$$

Definition. *Let $\Delta \in \{<, \leq, =, \geq, >\}$ and $x, y \in \mathbb{R}^n$. Then*

$$|x|\Delta|y| \quad \textit{iff} \quad |\chi_i|\,\Delta\,|\psi_i|,$$

for all $i = 0, \ldots, n-1$. Similarly, given A and $B \in \mathbb{R}^{m\times n}$,

$$|A|\Delta|B| \quad \textit{iff} \quad |\alpha_{ij}|\,\Delta\,|\beta_{ij}|,$$

for all $i = 0, \ldots, m-1$ and $j = 0, \ldots, n-1$.

Lemma 1. *Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. Then $|AB| \leq |A||B|$.*

Proof. Let $C = AB$. Then the (i, j) entry in $|C|$ is given by

$$|\gamma_{i,j}| = \left|\sum_{p=0}^{k-1} \alpha_{i,p}\beta_{p,j}\right| \leq \sum_{p=0}^{k-1} |\alpha_{i,p}\beta_{p,j}| = \sum_{p=0}^{k-1} |\alpha_{i,p}|\,|\beta_{p,j}|$$

which equals the (i, j) entry of $|A||B|$. Thus $|AB| \leq |A||B|$.

□

These bounds can be converted into bounds involving norms exploiting the following theorem, where $||\cdot||_F$ indicates the Frobenius matrix norm.

Theorem 1. *Let $A, B \in \mathbb{R}^{m\times n}$. If $|A| \leq |B|$ then $\|A\|_F \leq \|B\|_F$, $\|A\|_1 \leq \|B\|_1$, and $\|A\|_\infty \leq \|B\|_\infty$.*

Proof.

- Show that if $|A| \leq |B|$ then $\|A\|_F \leq \|B\|_F$.

$$\|A\|_F^2 = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\alpha_{i,j}\right|^2 \leq \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\beta_{i,j}\right|^2 = \|B\|_F^2 \tag{4.26}$$

  Hence $\|A\|_F \leq \|B\|_F$.

- Show that if $|A| \leq |B|$ then $\|A\|_1 \leq \|B\|_1$. Let

$$A = \begin{bmatrix} a_0 & \cdots & a_{n-1} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} b_0 & \cdots & b_{n-1} \end{bmatrix} \tag{4.27}$$

  Then

$$\|A\|_1 = \max_{0\leq j<n} \left\|a_j\right\|_1 = \max_{0\leq j<n}\left(\sum_{i=0}^{m-1}\left|\alpha_{i,j}\right|\right) = \left(\sum_{i=0}^{m-1}\left|\alpha_{i,k}\right|\right) \tag{4.28}$$

  with k being the index that maximizes (4.28). Then

$$\|A\|_1 =\leq \left(\sum_{i=0}^{m-1}\left|\beta_{i,k}\right|\right) \leq \|b_k\|_1 \leq \max_{0\leq j<n}\left\|b_j\right\|_1 \leq \|B\|_1 \ . \tag{4.29}$$

- Show that if $|A| \leq |B|$ then $\|A\|_\infty \leq \|B\|_\infty$. Note that: $\|A\|_\infty = \left\|A^T\right\|_1$ and $\|B\|_\infty = \left\|B^T\right\|_1$. Furthermore, if $|A| \leq |B|$ then, clearly, $\left|A^T\right| \leq \left|B^T\right|$. Hence

$$\|A\|_\infty = \left\|A^T\right\|_1 \leq \left\|B^T\right\|_1 = \|B\|_\infty \ . \tag{4.30}$$

□

Remark 22. *The previous theorem shows that the bound $|AB| < |A||B|$ can be converted into bounds involving norms. As a consequence, we can use the absolute value function to upper bound $\dot{V}_\ell$ and design a robustness term component-wise, potentially avoiding useless conservativism.*

Exploiting these properties, we obtain

$$\begin{aligned}\dot{V}_\ell &\leq -ke^\top e + e^\top u_{rob} + e^\top \left(-S(\omega_d)\ \Delta_J\ \omega_d + \Delta_J\ \dot{\omega}_d + \frac{1}{2}\dot{J}\omega \right) \\ &\leq -ke^\top e + e^\top u_{rob} + e^\top \left(S(\omega_d)\ \Delta_J + \frac{1}{2}\dot{J} \right)\ e + \\ &\quad + |e|^\top \left(|S(\omega_d)|\ |\Delta_J|\ |\omega_d| + |\Delta_J|\ |\dot{\omega}_d| + \frac{1}{2}|\dot{J}||\omega_d| \right)\ . \qquad (4.31)\end{aligned}$$

Defining $|\Delta_J|^{max}$ and $|\dot{J}|^{max}$ with the (i,j) entry given respectively by $|\Delta_J|^{max}_{i,j} = \sup_{t>0} |\Delta_{J_{i,j}}| \geq 0$ and $|\dot{J}|^{max}_{i,j} = \sup_{t>0} |\dot{J}_{i,j}| \geq 0$ yields

$$\begin{aligned}\dot{V}_\ell \leq -ke^\top e + e^\top u_{rob} + e^\top \left(S(\omega_d)\ \Delta_J + \frac{1}{2}\dot{J} \right)\ e + \\ + |e|^\top \left(|S(\omega_d)|\ |\Delta_J|^{max}\ |\omega_d| + |\Delta_J|^{max}\ |\dot{\omega}_d| + \frac{1}{2}|\dot{J}|^{max}|\omega_d| \right) \qquad (4.32)\end{aligned}$$

To avoid u_{rob} being not differentiable with respect the reference trajectory $(\omega_d, \dot{\omega}_d)$, we exploit the following lemma.

Lemma 2. *Let $A \in \mathbb{R}^{m\times k}$, then $|A| \leq A^+$ with the (i,j) entry given by:*

$$A^+_{i,j} = \frac{1}{2}(A^2_{i,j} + 1)\ . \qquad (4.33)$$

Hence, we can upper bound (4.32) with

$$\begin{aligned}\dot{V}_\ell \leq -ke^\top e + e^\top u_{rob} + e^\top \left(S(\omega_d)\ \Delta_J + \frac{1}{2}\dot{J} \right)\ e + \qquad (4.34) \\ + |e|^\top \left(S(\omega_d)^+\ |\Delta_J|^{max}\ \omega_d^+ + |\Delta_J|^{max}\ \dot{\omega}_d^+ + \frac{1}{2}|\dot{J}|^{max}\omega_d^+ \right)\end{aligned}$$

Defining the element-by-element hyperbolic tangent function as

$$\mathrm{Tanh}(x) := [\tanh(x_1), \tanh(x_2), \tanh(x_3)]^T\ ,$$

the element-by-element sign function as

$$\mathrm{Sign}(x) := [\mathrm{sign}(x_1), \mathrm{sign}(x_2), \mathrm{sign}(x_3)]^T\ ,$$

and the operator

$$\mathrm{diag}(x) := \begin{bmatrix} x_1 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{bmatrix} ,$$

we consider the following smooth control law[3]:

$$\begin{aligned} u_{rob} = &-\left(S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}\right) e + \qquad (4.35) \\ &-\mathrm{diag}(\tanh(\lambda e))\left((S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}) \; \omega_d^+ + |\Delta_J|^{max} \; \dot{\omega}_d^+\right) \end{aligned}$$

with $\lambda \in \mathbb{R}_{>0}$. Substituting (4.35) into (4.34), we obtain

$$\begin{aligned} \dot{V}_\ell \leq\, & e^\top \mathrm{diag}\left(\mathrm{Sign}(e) - \mathrm{Tanh}(\lambda e)\right)\left((S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}) \; \omega_d^+ + \right. \\ & \left. |\Delta_J|^{max} \; \dot{\omega}_d^+ \right) - k e^\top e \qquad (4.36) \\ \leq\, & |e|^T \mathrm{diag}(\mathrm{Sign}(e)) \; \mathrm{diag}\left(\mathrm{Sign}(e) - \mathrm{Tanh}(\lambda e)\right)\left(|\Delta_J|^{max} \; \dot{\omega}_d^+ + \right. \\ & \left. (S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}) \; \omega_d^+ \right) - k e^\top e \qquad (4.37) \\ \leq\, & |e|^T \; \mathrm{diag}\left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} - \mathrm{Tanh}(\lambda |e|)\right)\left(|\Delta_J|^{max} \; \dot{\omega}_d^+ + \right. \\ & \left. (S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}) \; \omega_d^+ \right) - k e^\top e \qquad (4.38) \end{aligned}$$

Using algebraic features of the hyperbolic tangent function yields:

$$\begin{aligned} \dot{V}_\ell \leq \mathrm{diag}\left(\frac{1}{\lambda e_{log}}\right)\left((S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2}|\dot{J}|^{max}) \; \omega_d^+ + |\Delta_J|^{max} \; \dot{\omega}_d^+\right) + \\ - k e^\top e \, . \qquad (4.39) \end{aligned}$$

[3] If a bound on ω_d and $\dot{\omega}_d$ is known during the entire trajectory, the robustness term u_{rob} can be determined using these bounds and avoiding time-varying parameters, such as $S(\omega_d)^+$.

From this bound it is clear that $\dot{V}_\ell < 0$ for all

$$\|e\|_2^2 > \left\| \operatorname{diag}\left(\frac{1}{\lambda e_{log} k}\right) \left((S(\omega_d)^+ \, |\Delta_J|^{max} + \frac{1}{2} |\dot{J}|^{max}) \, \omega_d^+ + \right. \right. \\ \left. \left. + |\Delta_J|^{max} \, \dot{\omega}_d^+ \right) \right\|_2 . \quad (4.40)$$

Whenever $\|e\|_2$ becomes large, the negative term "$-ke^\top e$" in equation (4.39) dominates the remaining terms, making $\dot{V}_\ell < 0$ and drawing the error toward the origin. When $\|e\|_2$ reaches sufficiently small values (determined by λ), the sign of V_ℓ is indefinite, and progress toward the origin cannot be guaranteed. This critical size represents the boundary of a residual error set to which all trajectories will eventually converge. This set can be made smaller by increasing the λ parameter (for the limit case $\lambda \to \infty$ the asymptotic tracking is achieved). As a result, the closed-loop trajectories and the error e are globally bounded.

Remark 23. *As in Chapter 3, the adaptation law in* (4.23) *suffers from the drawback that the parameter* $\hat{\theta}$ *may depart arbitrarily away from its actual value. However, we can solve this issue by using the robustness modifications described in Section 3.4.2 with similar considerations. Furthermore, if the matrices* $|\Delta_J|^{max}$ *and* $|\dot{J}|^{max}$ *are unknown, an adaptive modification can be designed as in Section 3.4.3.*

4.5 Simulation results

To show the performance characteristics of the proposed adaptive control schemes, numerical simulations are conducted in this Section. Specifically, the results related to two different manoeuvres are presented: a *motion-to-rest* manoeuvre, *i.e.* angular rate stabilization, and a *motion-to-motion* manoeuvre, *i.e.* angular velocity tracking. As in [126], we assume that the inertia of the spacecraft takes the form

$$J(t) = J_0 + J_1(t) \, , \quad (4.41)$$

where J_0 is constant and represents the rigid part of the spacecraft, and $J_1(t)$ is time-varying and represents a moving part of the spacecraft. The true value of the inertia constant component J_0 is taken as:

$$J_0 = \begin{bmatrix} 20 & 1.2 & 0.9 \\ 1.2 & 17 & 1.45 \\ 0.9 & 1.4 & 15 \end{bmatrix} \text{Kgm}^2 \,. \tag{4.42}$$

Concerning the time-varying part $J_1(t)$, a sinusoidal mass displacement profile is used to represent a persistent mass movement of an articulated appendage (see [126, Section V.A] for more details on the modeling part).

4.5.1 Motion-to-rest manoeuvre

We use the controller proposed in Section 4.3.2 with the control law (4.10) and adaptation law (4.12) to bring the spacecraft from initial angular velocity $\omega(0) = [20\ -5\ 15]^{\mathrm{T}}$ rad/sec to rest $\omega = [0\ 0\ 0]^{\mathrm{T}}$ rad/sec. The parameters used in the simulation are $\gamma = 10$, $\hat{\sigma}(0) = 0$, and the proportional gain $K_{\mathrm{v}} = \mathrm{diag}([0.1\ 0.1\ 0.1])$. Namely, we select K_{v} that does not satisfy the constraint mentioned in Remark 18 ($K_{\mathrm{v}} > -\frac{1}{2}\dot{J}$) to show the effectiveness of the proposed adaptive control scheme. Simulating the closed-loop system yields the time evolutions of the angular rate and control input shown in Figure 4.1 and Figure 4.2, respectively.

We can state that the proposed controller achieves asymptotic stabilization of the angular rate despite the time-varying nature of the inertia without any *a priori* knowledge of the spacecraft characteristics.

4.5.2 Motion-to-motion manoeuvre

As in the previous Chapter, we compare in simulation the proposed approach with the standard adaptive control scheme (without robustness term) with two different robustness modifications. Specifically, we compare three state feedback adaptive controllers:

- *Controller A* is the proposed controller with the control law (4.35), and the update law (4.23) for $\hat{\theta}$ with the projection modification;

- *Controller B* uses the control law (4.35) with $u_{rob} = 0$ and the update law for $\hat{\theta}$ (4.23) with a projection modification;

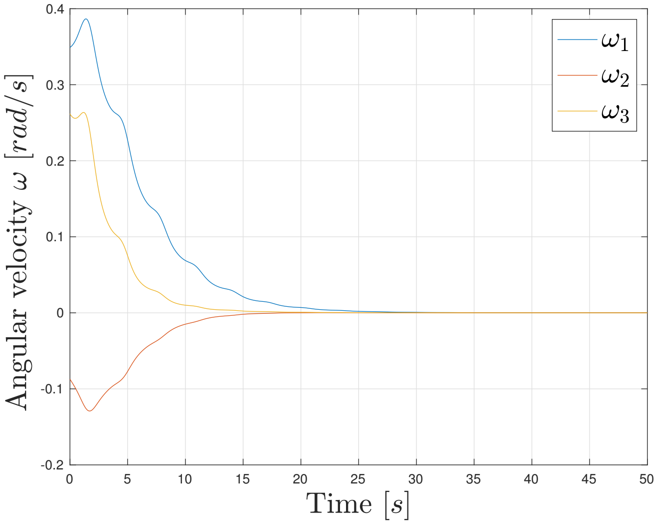


Figure 4.1: *Angular velocity components for a motion-to-rest manoeuvrer.*

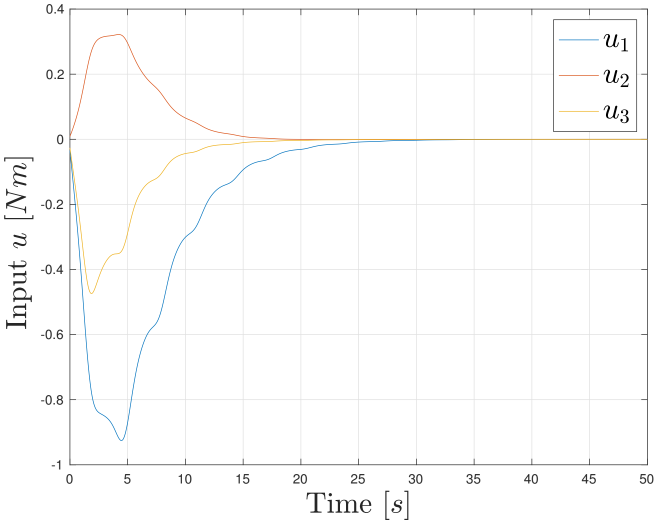


Figure 4.2: *Control torque components for a motion-to-rest manoeuvrer.*

- *Controller C* is identical to *Controller B* but also includes the σ-modification in the update law (4.23).

The control objective is to track a given angular velocity profile $\omega_d = s(t)[1\ \ 1\ \ 1]^{\mathrm{T}}$ rad/sec with $s(t) = 2\sin(0.1t) + 0.1\sin(t)$. The parameters used in the simulation are $\omega(0) = [0.1\ 0.1\ 0.1]^{\mathrm{T}}$ rad/sec, $\hat{\theta}(0) = [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$, $k = 100$ and $\Gamma_\theta = 1000\ \mathrm{I}_6$ for all the adaptive controllers, $\lambda = 10$ for the *Controller A*, and $\sigma = 10$ for the *Controller C*. Simulating the three closed-loop systems yields the results listed in Table 4.1 in terms of error norms. The time evolutions of the error norm and control input are shown in Figure 4.3 and Figure 4.4, respectively.

Table 4.1: *Error norms during a motion-to-motion manoeuvrer.*

	Error 2-norm $\|e\|_2$ rad/s	Error ∞-norm $\|e\|_\infty$ rad/s
Controller A	0.14	0.03
Controller B	10.47	0.32
Controller C	9.29	0.17

We can state that the proposed controller shows better performance with respect to the standard adaptive schemes in terms of smaller overshoot at the beginning, smoother response and control input, faster rate of convergence, and smaller steady-state error. These effects are due to the robustness term u_{rob} which dominates the perturbation introduced by the time-varying parameters, *i.e.*, “$W\ \Delta_\theta$” and “$-\dot{J}\omega$”. Finally, we can see the proposed controller exhibits less oscillations with respect to the standard adaptive schemes thanks to the introduction of the robustness term filtered by the hyperbolic tangent function. Figure 4.5, showing the amplitude of the input spectrum[4], makes this effect more evident.

[4] For the sake of conciseness, only the first component u_1 is reported.

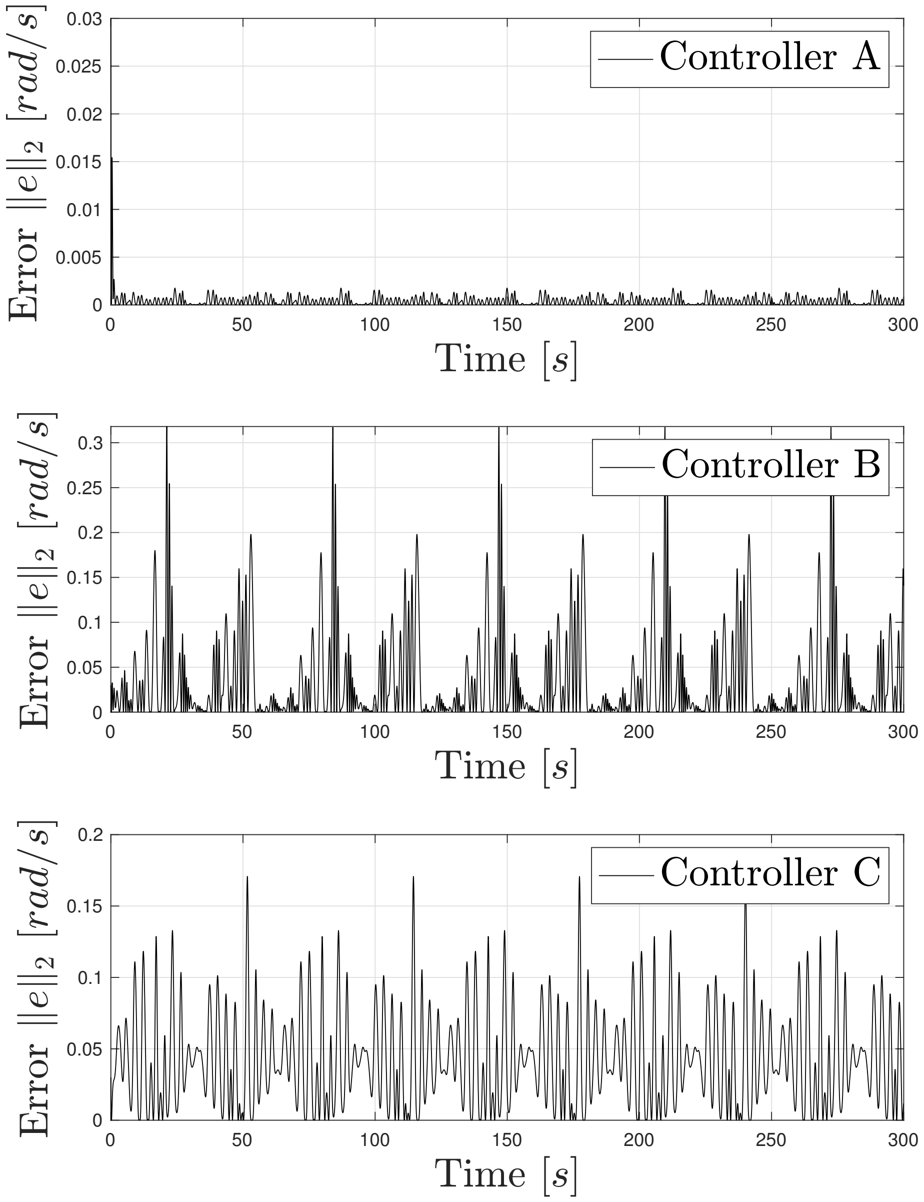


Figure 4.3: *Time evolution of the error 2-norm with different controllers.*

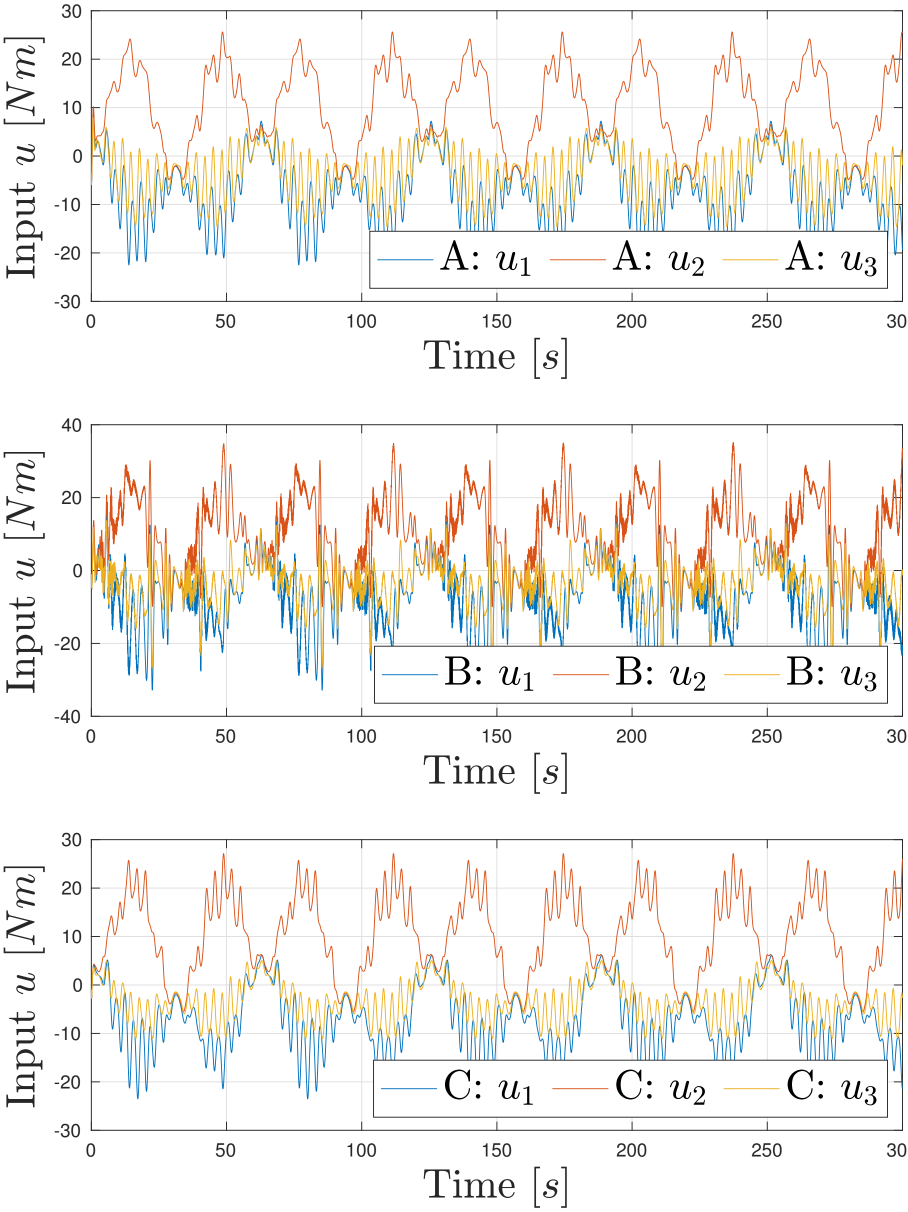


Figure 4.4: *Control torque components for a motion-to-motion manoeuvrer.*

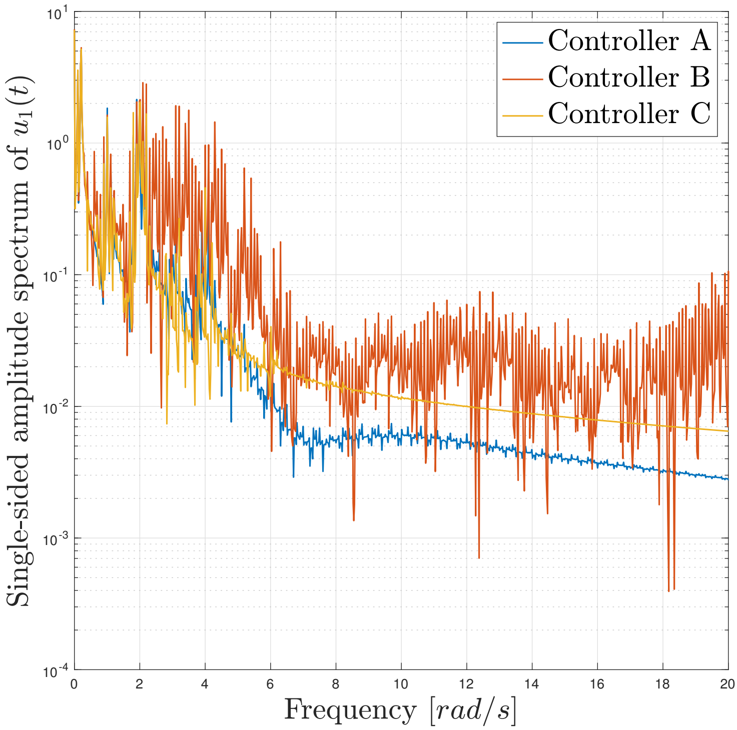


Figure 4.5: *Single-sided amplitude spectrum of u_1 in motion-to-motion manoeuvrer.*

Part II

Harmonic Control for disturbance rejection

CHAPTER 5

Harmonic Control theory

Harmonic Control (short, HC) has been the subject of extensive research over the last few decades, particularly in the rotorcraft community. HC enables the system to reach harmonic steady-state in the case of tonal or multi-tonal disturbances with a known spectrum. The HC framework is based on the representation of the system as a linear quasi-steady model constructed in the frequency domain, known as T-matrix representation. This Chapter is devoted to presenting a brief overview of the HC algorithms starting from the definition of the T-matrix for linear systems.

5.1 Background

Feedback control for disturbance rejection plays a central role in an increasing number of applications, and control techniques aimed at improving its overall effectiveness have been an established topic of ongoing research since the early *1980s* [128]. Several families of algorithms have emerged over the years, the classification of which has generally been based on two main points [129]: knowledge of the system dynamics and the properties of the disturbances. When the disturbance has a known spectrum and is tonal or multi-tonal, the controller can be modified to include a model of the exogenous signal, yielding high-gain feedback at the frequencies that make up the spectrum of the disturbance. As a result, the controller could apply infinite gain to the frequencies of the disturbance to achieve a complete reduction of the disturbance. An alternative approach allows the system to reach harmonic steady-state and uses measurements of the steady-state response amplitude and phase to determine the required control signal. Active rotor balancing control, also known as Convergent Control [130, 131, 132], and the active control of rotor-induced vibration in helicopters, where the technique is referred as Harmonic Control [133], are two renowned applications of this approach. For a complete overview of the HC techniques, the readers are referred to the survey papers produced by Friedmann and Millott [134] and Teves *et al.* [135], and references therein.

5.2 Linear quasi-steady model

Referring to Figure 5.1, we consider $u \in \mathbb{R}^m$ the vector of control inputs and $y \in \mathbb{R}^p$ the vector of measured outputs. Assuming that the dynamics

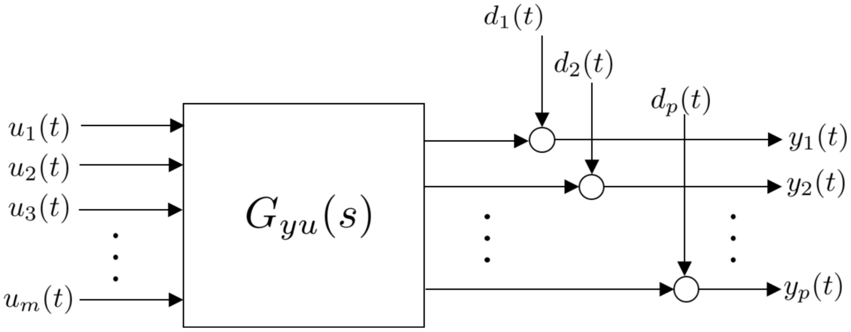


Figure 5.1: *Mixed time-frequency domain representation in terms of $G_{yu}(s)$ [136].*

relating u to y is Linear Time Invariant (LTI), it can be described by a nominal transfer function denoted as:

$$G_{yu}(s) = \{g_{y_i,u_j}(s)\} \quad i = 1,\ldots,\ p \ \text{ and } j = 1,\ldots,\ m. \tag{5.1}$$

Furthermore, we assume that $G_{yu}(s)$ corresponds to an asymptotically stable system. We consider the control input and the baseline disturbance $d \in \mathbb{R}^p$, acting on the system output y, be multi-harmonic with frequencies ω_h with $h = 1,\ldots,\ H$. As a result, the steady-state signals $d(t)$, $u(t)$ and $y(t)$ can be expressed as:

$$d(t) = \begin{bmatrix} d_1(t) & d_2(t) & \cdots & d_p(t) \end{bmatrix}^T , \tag{5.2}$$

$$y(t) = \begin{bmatrix} y_1(t) & y_2(t) & \cdots & y_p(t) \end{bmatrix}^T , \tag{5.3}$$

$$u(t) = \begin{bmatrix} u_1(t) & u_2(t) & \cdots & u_m(t) \end{bmatrix}^T \tag{5.4}$$

with

$$d_i(t) = \sum_{h=1}^{H} \left(d_{c_i} \cos(\omega_h t) + d_{s_i} \sin(\omega_h t)\right) \quad i = 1,\ldots,\ p\ , \tag{5.5}$$

$$y_i(t) = \sum_{h=1}^{H} \left(y_{c_i} \cos(\omega_h t) + y_{s_i} \sin(\omega_h t)\right) \quad i = 1,\ldots,\ p\ , \tag{5.6}$$

$$u_j(t) = \sum_{h=1}^{H} \left(u_{c_j} \cos(\omega_h t) + u_{s_j} \sin(\omega_h t)\right) \quad j = 1,\ldots,\ m\ . \tag{5.7}$$

where $y_{c_i}^{(h)}$ and $y_{s_i}^{(h)}$ (similarly $d_{c_i}^{(h)}, d_{s_i}^{(h)}$ and $u_{c_j}^{(h)}, u_{s_j}^{(h)}$) are given by

$$y_{c_i}^{(h)} = \frac{2}{T_p^{(h)}} \int_0^{T_p^{(h)}} y_i^{(h)} \cos(\omega_h t) dt\ , \tag{5.8}$$

$$y_{s_i}^{(h)} = \frac{2}{T_p^{(h)}} \int_0^{T_p^{(h)}} y_i^{(h)} \sin(\omega_h t) dt\ , \tag{5.9}$$

where $T_p^{(h)} = \frac{2\pi}{\omega_h}$. Then, we define the vector

$$Y_i^{(h)} = \begin{bmatrix} y_{c_i}^{(h)} \\ y_{s_i}^{(h)} \end{bmatrix} \quad i = 1,\ldots,\ p \tag{5.10}$$

and in the same way $D_i^{(h)}$ and $U_j^{(h)}$,

$$D_i^{(h)} = \begin{bmatrix} d_{c_i}^{(h)} \\ d_{s_i}^{(h)} \end{bmatrix} \quad i = 1, \ldots, \; p \; , \tag{5.11}$$

$$U_j^{(h)} = \begin{bmatrix} u_{c_j}^{(h)} \\ u_{s_j}^{(h)} \end{bmatrix} \quad j = 1, \ldots, \; m \; . \tag{5.12}$$

As a result, the following expression is obtained at the steady-state

$$Y_i^{(h)} = T_{i,j}^{(h)} U_j^{(h)} + D_i^{(h)} \tag{5.13}$$

with

$$T_{i,j}^{(h)} = \begin{bmatrix} \text{Real}[g_{y_i,u_j}(j\omega_h)] & \text{Imag}[g_{y_i,u_j}(j\omega_h)] \\ -\text{Imag}[g_{y_i,u_j}(j\omega_h)] & \text{Real}[g_{y_i,u_j}(j\omega_h)] \end{bmatrix} . \tag{5.14}$$

Defining the vector

$$Y^{(h)} = \begin{bmatrix} Y_1^{(h)T} & \cdots & Y_p^{(h)T} \end{bmatrix}^T \tag{5.15}$$

and similarly for $D^{(h)}$ and $U^{(h)}$, the steady-state response at frequency ω_h can be represented as

$$Y^{(h)} = T^{(h)} U^{(h)} + D^{(h)} \; , \tag{5.16}$$

where

$$T^{(h)} = \begin{bmatrix} T_{1,1}^{(h)} & \cdots & T_{1,m}^{(h)} \\ \vdots & & \vdots \\ T_{p,1}^{(h)} & \cdots & T_{p,m}^{(h)} \end{bmatrix} . \tag{5.17}$$

In the following, we consider, without any loss of generality, the problem of rejecting a disturbance consisting of a single harmonic, *i.e.*, $H = 1$. For the sake of simplicity, the index h will be dropped.

5.2.1 Extension to LTP systems: HTF

This Section provides some background on the frequency response of Linear Time Periodic (LTP) systems and uses such analytical tools to derive explicit expressions for the T-matrix. LTP systems can be lifted to formally LTI systems using various techniques [137]-[138]. This Section examines one such representation, the Harmonic Transfer Function (HTF). Considering a continuous-time LTP system

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t), \end{aligned} \tag{5.18}$$

each matrix can be expanded in a complex Fourier series[1]

$$A(t) = \sum_{m=-\infty}^{\infty} A_m e^{jm\omega t} \tag{5.19}$$

and similarly for $B(t)$, $C(t)$ and $D(t)$. The system can be analysed in the frequency domain as follows. Introduce the class of *Exponentially Modulated Periodic* (EMP) signals [140]. The complex signal $u(t)$ is said to be EMP of period T and modulation s if

$$u(t) = \sum_{k=-\infty}^{\infty} u_k e^{s_k t} = e^{st} \sum_{k=-\infty}^{\infty} u_k e^{jk\omega t} \tag{5.20}$$

where $t \geq 0$, $s_k = s + jk\omega$, and s is a complex scalar.

Remark 24. *The class of EMP signals is a generalization of the class of T-periodic signals, i.e. of signals with period T: in fact, an EMP signal with $s = 0$ is just an ordinary time-periodic signal.*

Remark 25. *An LTP system forced by an EMP input has an EMP steady-state response, similar to how an LTI system forced by a complex exponential input has an exponential steady-state response. Specifically, all signals of interest (x, $\dot{x}$, y) in an LTP system are EMP signals during a response subject to EMP input.*

[1] The expansion can be performed only under the assumptions depicted in [139].

By deriving the Fourier expansions for $A(t)$, $B(t)$, $C(t)$ and $D(t)$, it is possible to prove that the EMP steady-state response of the system can be expressed as the infinite dimensional matrix equation with *constant* elements [140]

$$\begin{array}{rcl} s\mathcal{X} & = & (\mathcal{A}-\mathcal{N})\mathcal{X}+\mathcal{B}\mathcal{U} \\ \mathcal{Y} & = & \mathcal{C}\mathcal{X}+\mathcal{D}\mathcal{U} \end{array} \tag{5.21}$$

where $\mathcal{X}, \mathcal{U}$ and $\mathcal{Y}$ are doubly infinite vectors formed with the harmonics of x, u and y respectively, organized in the following fashion:

$$\mathcal{X}^T = \begin{bmatrix} \cdots & x_{-2}^T & x_{-1}^T & x_0^T & x_1^T & x_2^T & \cdots \end{bmatrix} \tag{5.22}$$

and similarly for $\mathcal{U}$ and $\mathcal{Y}$. $\mathcal{N}$ is a block diagonal complex-valued matrix given by $\mathcal{N} = \text{blkdiag}\{jm\omega I\}$ where I is the identity matrix the size of which is equal to the number of states (see [141] for more details). $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are doubly infinite Toeplitz matrices [142]-[143]. They are formed with the harmonics of $A(\cdot)$, $B(\cdot)$, $C(\cdot)$ and $D(\cdot)$ respectively as follows

$$\mathcal{A} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & A_0 & A_{-1} & A_{-2} & A_{-3} & A_{-4} & \cdots \\ \cdots & A_1 & A_0 & A_{-1} & A_{-2} & A_{-3} & \cdots \\ \cdots & A_2 & A_1 & A_0 & A_{-1} & A_{-2} & \cdots \\ \cdots & A_3 & A_2 & A_1 & A_0 & A_{-1} & \cdots \\ \cdots & A_4 & A_3 & A_2 & A_1 & A_0 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{5.23}$$

(and similarly for $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$), where the submatrices A_n in equation 5.23 are the coefficients of the Fourier expansion of matrix $A(t)$, given in equation 5.19. To relate these coefficients to those of the Fourier series expansion in trigonometric form

$$A(t) = A_0 + \sum_{k=1}^{K} \left[A_{kc}\cos(k\omega t) + A_{ks}\sin(k\omega t)\right] \ , \tag{5.24}$$

it is recalled that the Fourier series expansion of a scalar function can be rewritten in complex exponential form, *i.e.*,

$$a(t) = a_0 + \sum_{k=1}^{\infty} (a_{kc}\cos k\omega t + a_{ks}\sin k\omega t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega t} \tag{5.25}$$

with $a_k = (a_{kc} - ja_{ks})/2$ and $a_{-k} = (a_{kc} + ja_{ks})/2$ for $k = 1, 2, \ldots, \infty$. Therefore, the coefficients of equations 5.19 and 5.24 are related by:

$$\begin{aligned} A_k &= \tfrac{1}{2}(A_{kc} - jA_{ks}) \\ A_{-k} &= \tfrac{1}{2}(A_{kc} + jA_{ks}) \end{aligned} \qquad k = 1, 2, \ldots \tag{5.26}$$

with A_0 identical in both equations 5.19 and 5.24. Similar relations hold for the harmonics of B, C, and D. From equation 5.21, one can define the HTF $\mathcal{G}(s)$ as the operator:

$$\mathcal{G}(s) := \mathcal{C}[s\mathcal{I} - (\mathcal{A} - \mathcal{N})]^{-1}\mathcal{B} + \mathcal{D} \tag{5.27}$$

which relates the input harmonics and the output harmonics (contained in the infinite vectors $\mathcal{U}$ and $\mathcal{Y}$ respectively). A more detailed explanation of this development with the related proofs is provided in [140, 144].

Harmonic Transfer Matrix

Analysis of an LTP system response to an EMP signal when $s = 0$ is of particular interest, *i.e.*, a periodic signal with the same period as the LTP system. The resulting output response, as noted in Remark 25, will be an EMP signal with $s = 0$. Namely, we have

$$u(t) = \sum_{k=-\infty}^{+\infty} u_k e^{jk\omega t} \xrightarrow{\mathcal{G}(s)} y(t) = \sum_{k=-\infty}^{+\infty} y_k e^{jk\omega t} \tag{5.28}$$

and

$$\mathcal{Y} = \mathcal{G}(0)\,\mathcal{U} \ \text{ with } \ \mathcal{G}(0) = \mathcal{C}[\mathcal{N} - \mathcal{A}]^{-1}\mathcal{B} + \mathcal{D}. \tag{5.29}$$

In this case, $\mathcal{U}$ and $\mathcal{Y}$ are constant vectors of the Fourier coefficients of $u(t)$ and $y(t)$, respectively. Indeed, $\mathcal{G}(0)$ is a steady-state matrix gain that gives the Fourier coefficients of the output when the input is a T-periodic signal of known Fourier coefficients. The operator $\mathcal{G}(0)$, usually called Harmonic Transfer Matrix (HTM), is for LTP systems what the frequency response is for LTI. HTM yields the T-matrix relation by converting the harmonics of the output from exponential to trigonometric form [144].

5.3 HC architecture

HC was initially proposed by Shaw in [10]. Exploiting the linear relationship (5.16), Shaw analysed the frequency response function between the Fourier coefficients of the measured vibration outputs and the harmonic inputs. As a consequence, a general complex non-linear problem is transformed into a straightforward linear association in the *frequency-domain*. A schematic overview of a typical HC system is depicted in Figure 5.2.

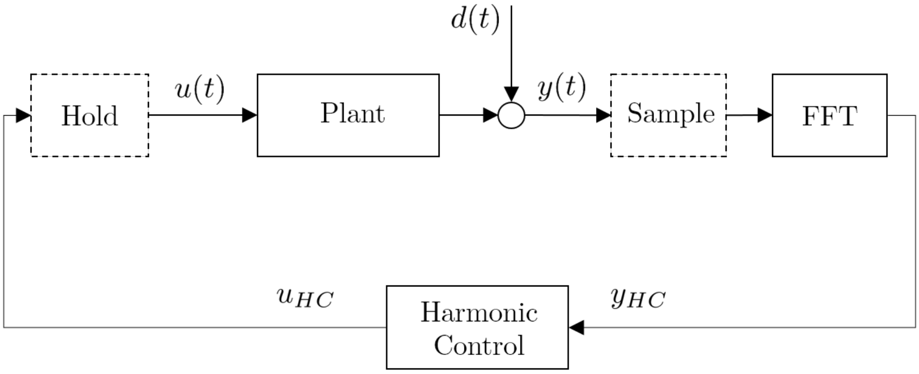


Figure 5.2: *Schematic overview of the HC architecture.*

Although HC is not the only method that could be used, it is significant for practical and historical reasons, particularly in the rotorcraft community (see, *e.g.*, in [145], [146], [133], [147]). The HC controller is considered to operate in the *frequency-domain*, because it handles only the harmonics of output and control components. Indeed, it involves a harmonic analysis, *e.g.*, a Fast Fourier Transform (FFT), to extract the harmonics of the output response. Then, the HC controller computes the optimal values for harmonic controls, which are injected as part of the control signals. In this way, rather than controlling the output itself, the control system controls the factors that determine the amplitude of the harmonic output signal.

The steps required to transform between the *time-* and *frequency-domain* are not instantaneous, but are carried out over finite time intervals. Indeed, the closed-loop HC algorithm is typically defined not as a continuous system, but rather at specific instants t_k, where $t_k = k\Delta t$ with $k \in \mathbb{Z}_{\geq 0}$

is the discrete-time index. Specifically, in order to give the system enough time to sample the output signal, perform the harmonic analysis, compute updated control amplitude and phase quantities, and allow for dissipation of transient dynamics, the controls are updated every t_k and kept constant for the entire time interval Δt. The underlying assumption is that HC inputs are updated slowly enough for the transient dynamics to have a negligible impact on the closed-loop dynamics [148].

5.4 Baseline T-*matrix* algorithm

As mentioned above, HC assumes that the measurements of the plant output and the updates of the control input are not performed continuously, but rather at specific times $t_k = k\Delta t$. We define the vector $U(k)$ of the harmonics of the control signals computed by the HC system at time t_k. Similarly, we define the vector $Y(k)$ which contains the cosine and sine harmonics of the output signal. The T-matrix of relation (5.16) can be either estimated from measured data, using on-line or off-line identification algorithms, or computed on the basis of a mathematical model of the plant [136]. At each discrete-time step k, the HC controller selects the value of the input harmonics $U(k)$ to attenuate the effects of the disturbance D (assumed constant over the time interval Δt) on the output $Y(k)$.

Considering the linear relation

$$Y(k) = TU(k) + D \,, \tag{5.30}$$

the optimal open-loop solution can be written as[2]

$$U(k) = -T^{-1}D. \tag{5.31}$$

Since the disturbance D cannot, in general, be measured directly, the same result can be achieved by employing a discrete-time integral control law in closed-loop, *i.e.*, based on the measurements $Y(k)$,

$$U(k+1) = U(k) - T^{-1}Y(k) \,. \tag{5.32}$$

Note that the integral action (5.32) ensures that $Y(k) \to 0$ when $k \to \infty$.

[2] Equation (5.30) assumes the T-matrix square ($m = p$). If this assumption does not hold, the control algorithm must be written as $U(k+1) = U(k) - T^{\dagger}Y(k)$, where $T^{\dagger} = (T^T T)^{-1} T^T$ is the pseudo-inverse of T.

By replacing k with $k+1$ in (5.30) and subtracting the resulting equation from (5.32), the model without the (unknown) disturbance is obtained

$$Y(k+1) = Y(k) + T(U(k+1) - U(k)) \tag{5.33}$$

This model representation is equivalent to a local response model, which relates the discrete change in the output to a step change in the control input between two consecutive time samples. Figure 5.3 shows the general block diagram of the T-matrix algorithm, where the delay operator z^{-1} is used as in the following equation:

$$U(k) = U(k-1) - T^{-1}Y(k-1) = z^{-1}(U(k) - T^{-1}Y(k)). \tag{5.34}$$

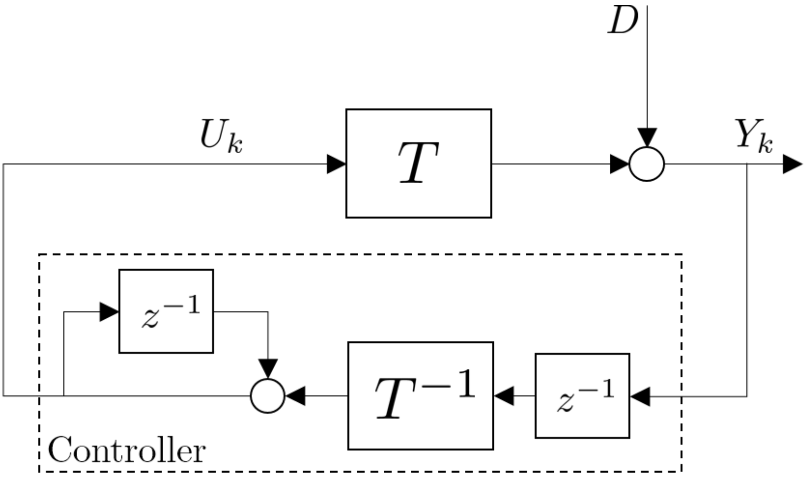


Figure 5.3: *General block diagram of discrete baseline T-matrix algorithm.*

5.5 Optimal *LQ*-based algorithm derivation

The discrete-time T-matrix algorithm has undergone modifications and improvements over the years, either with the goal of minimizing a defined cost-function (as the method described in this Section) or by resorting to adaptation mechanisms designed to recover quickly from improper initialization or enhanced robustness properties [149].

Specifically, a Linear Quadratic (*LQ*)-based HC version has been derived by minimizing at each discrete-time step k the following cost function

$$J(k) = Y(k)^T QY(k) + 2Y(k)^T SU(k) + U(k)^T RU(k) \tag{5.35}$$

where weighting matrices $Q = Q^T \geq 0$, $Q \in \mathbb{R}^{2p\times 2p}$, $S \in \mathbb{R}^{2p\times 2m}$ and $R = R^T \geq 0$, $R \in \mathbb{R}^{2m\times 2m}$ are defined such that

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \tag{5.36}$$

is positive semi-definite. Most implementations ignore the cross-weighting term S in equation (5.35), *i.e.*, the cost function is simplified to:

$$J(k) = Y(k)^T QY(k) + U(k)^T RU(k) \ . \tag{5.37}$$

Substituting $Y(k)$ from (5.30) into (5.37), we obtain

$$J(k) = U(k)^T VU(k) + 2U(k)^T T^T QD + D^T QD \ , \tag{5.38}$$

where V is defined as

$$V = T^T QT + R \ . \tag{5.39}$$

The optimal control law is found by differentiating the cost function (5.38) with respect to the control input harmonics U_k

$$\frac{\partial J(k)}{\partial U(k)} = 2VU(k) + 2T^T QD = 0 \ , \tag{5.40}$$

leading to the open-loop control algorithm

$$U(k+1) = -(T^T QT + R)^{-1}\big(T^T Q\big)D \ , \tag{5.41}$$

which can be equivalently written as

$$U(k+1) = -TKU(k) + KY(k) \ , \tag{5.42}$$

where K is defined as

$$K = -V^{-1}(T^T Q). \tag{5.43}$$

Remark 26. *Note that the control law* (5.42) *still introduces a discrete-time integral action. Considering $Q = I$ and $R = 0$, the output goes to zero after one discrete-time step (deadbeat control) if the T-matrix is known without uncertainty. Moreover, the control law* (5.42) *reduces to equation* (5.32) *that minimizes the cost function $J(k) = Y(k)^T Y(k)$ avoiding to bound the control effort.*

As for the implementation of this discrete algorithm, Algorithm 1 summarizes the operations need to be carried out at each time step t_k.

Algorithm 1 Implementation of the LQ-based HC law

1) Determination of the ω-harmonic component of y

$$y_{c_i} = \frac{2}{T_p} \int_{t_k}^{t_k+T_p} y_i \cos(\omega t) dt \ \text{ and } \ y_{s_i} = \frac{2}{T_p} \int_{t_k}^{t_k+T_p} y_i \sin(\omega t) dt \ \text{ for } i = 1, \dots, \ p$$

2) Update U_k using the optimal control law

$$U(k+1) = -(T^T QT + R)^{-1} \big(T^T Q\big)(Y(k) - TU(k))$$

3) Determination of the time domain value of the control variable u

$$u_j = u_{c_j} \cos(\omega t) + u_{s_j} \sin(\omega t) \text{ for } j = 1, \dots, \ m$$

5.5.1 Convergence analysis

At each time t_k, the effect of the disturbance can be evaluated as

$$D = Y(k) - TU(k). \tag{5.44}$$

Therefore, by substituting (5.44) into (5.30) evaluated at $k+1$ one obtains

$$\begin{aligned} Y(k+1) &= TU(k+1) + Y(k) - TU(k) \\ &= TKTU(k) - TKY(k) + Y(k) - TU(k) \\ &= (I - TK)Y(k) + T(KT - I)U(k). \end{aligned} \tag{5.45}$$

As discussed in [150], with initial conditions $Y(0)$ and $U(0)$, the system dynamics (5.45) is equivalent to

$$\begin{bmatrix} Y(k+1) \\ U(k+1) \end{bmatrix} = A_T \begin{bmatrix} Y(0) \\ U(0) \end{bmatrix} \tag{5.46}$$

$\forall\, k \geq 0$, where

$$A_T := \begin{bmatrix} I - TK & T(KT - I) \\ -K & KT \end{bmatrix} \tag{5.47}$$

is an idempotent matrix. As a result, the optimal values of the control input $U(k)$ and resulting output $Y(k)$ are attained after the first update.

Remark 27. *This result holds true under the assumption that T-matrix is known. If the T-matrix is uncertain, HC could result in degraded performance and possible instability (see the next Section for the robustness analysis with respect to an uncertain T). Moreover, the one-step convergence property is lost.*

Based on equation (5.46), the steady-state output can be written as

$$Y_{opt} := Y(k) = K_{LQ}\left(Y(0) - TU(0)\right) = K_{LQ} D \,, \tag{5.48}$$

where

$$K_{LQ} = Q^{-1}\left(Q^{-1} + TR^{-1}T^T\right)^{-1} \tag{5.49}$$

is a gain matrix which represents the ”distance” from the complete reduction of the disturbance D. Furthermore, following the steps described in [150], an upper bound of the output norm $\|Y(k)\|$ is obtained as

$$\|Y(k)\| \leq \frac{\sigma_{max}(R)}{\sigma_{min}(Q)\sigma_i(TT^T)} \|D\| \,, \tag{5.50}$$

where $i = \min(2p, 2m)$.

5.5.2 Robustness analysis

The analysis in the previous Section shows that the LQ-based HC controller ensures a bounded output for any bounded disturbance with a tunable upper bound, thus ensuring robustness against exogenous perturbations, as confirmed by previous studies in [151, 141, 152, 153]. However, the implementation of HC control law requires the exact knowledge of the T-matrix: as noted in Remark 27, an erroneous model of the system results in a wrong T-matrix which could degrade performance and possibly lead to instability.

Given the approximations introduced in deriving the model for control, it is worth understanding the robustness of the proposed algorithm with respect to model variations. To this aim, if an estimate $\hat{T}$ of T is given as

$$\hat{T} = T + \Delta T, \tag{5.51}$$

then the control law defined in equation (7.1) becomes

$$U(k+1) = -\hat{T}\hat{K}U(k) + \hat{K}Y(k), \tag{5.52}$$

where

$$\hat{K} = -(\hat{T}^T Q \hat{T} + R)^{-1}(\hat{T}^T Q). \tag{5.53}$$

As discussed in [151] the stability of the HC algorithm requires that

$$\rho_s(\hat{K}\Delta T) < 1, \tag{5.54}$$

where $\rho_s(\cdot)$ is the spectral radius. An upper bound on the spectral radius of $\hat{K}\Delta T$ can be derived by using (5.53) to obtain:

$$\begin{aligned}\rho_s(\hat{K}\Delta T) &= \rho_s((\hat{T}^T Q \hat{T} + R)^{-1}\hat{T}^T Q \Delta T) \\ &\leq \frac{(\sigma_{max}(T) + \sigma_{max}(\Delta T))\,\sigma_{max}(Q)\sigma_{max}(\Delta T)}{\sigma_{min}(R)}.\end{aligned} \tag{5.55}$$

Following the steps in [151], we obtain that if

$$\sigma_{max}(\Delta T) < -\frac{\sigma_{max}(T)}{2} + \frac{1}{2}\sqrt{\sigma_{max}(T)^2 + 4\frac{\sigma_{min}(R)}{\sigma_{max}(Q)}}\,, \tag{5.56}$$

then $\rho_s(\hat{K}\Delta T) < 1$.

Remark 28. *As noted in [150], there is evidence from equations* (5.50) *and* (5.56) *of the presence of a trade-off between performance and robustness properties of the closed-loop system (see Figure 5.4). Specifically, if the ratio* $\rho := \frac{\sigma_{min}(R)}{\sigma_{max}(Q)}$ *is large, i.e., tuning aims at minimum control effort* *(see equation* (5.35)*), then, according to inequality* (5.56)*, the control law achieves a high degree of robustness against parametric uncertainties. However, inequality* (5.50) *suggests that poor disturbance attenuation performance must be expected for large values of* ρ.

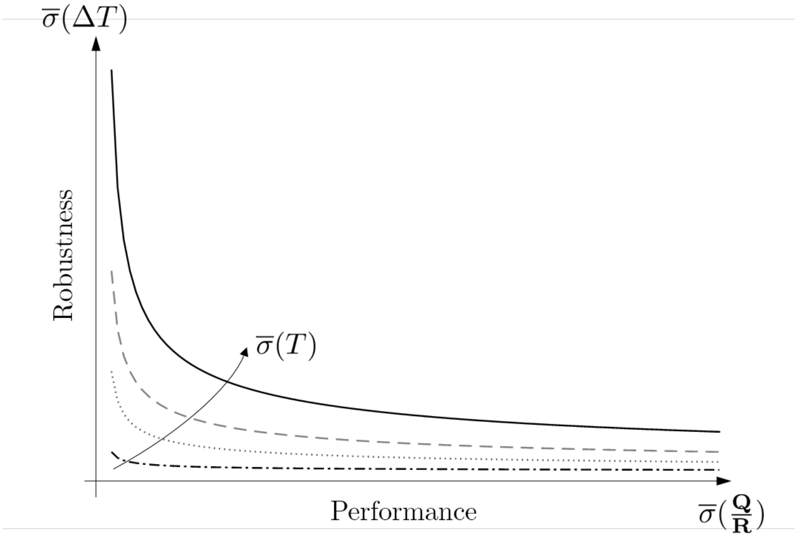


Figure 5.4: *Trade-off Robustness/Performance with the LQ-based HC Algorithm [136].*

Remark 29. *The validity of the previous method relies on the control input being updated slowly enough that transient dynamics do not affect the steady-state response measurements. Note that, if the transient dynamics is not allowed to dissipate, the overall closed-loop response at the sample times* t_k *will contain a non-zero spurious component corresponding to the zero-input response of the states, resulting in erroneous control amplitude and phase estimates for the next iteration. This is effectively equivalent to introducing model uncertainty in the estimate of the* T*-matrix. Continuous, high frequency, periodic excitation of the system transient behavior may result in unstable or unbounded amplitude responses [150].*

CHAPTER 6

Active balancing system for rotating orbital devices

This Chapter presents the design of an Active Balancing System (ABS) for rotating orbital devices, motivated by recent space applications for spacecraft endowed with rotating payloads. The main motivation behind this work is the Copernicus Imaging Microwave Radiometry (CIMR) mission which will feature a large rotating microwave radiometer to provide observations of sea-surface temperature, sea-ice concentration and sea-surface salinity. Due to the presence of highly uncertain inertial asymmetries in the rotating device, potentially large internal forces and torques can appear at interface between the spacecraft and the rotor which can cause a significant degradation of the system performance and can even affect its stability. To counteract such unbalance effects, in this work we develop an active balancing system made of a suitable set of actuated movable masses and sensors. The problem can be formulated as the one of compensating a periodic disturbance of a known frequency, being the unbalance effects essentially periodic due to the constant payload angular rate. In view of this, a HC algorithm has been designed to command the positions of the actuated masses in such a way that the effects of rotor unbalance are significantly reduced. After extensive numerical simulations, accounting for both parametric uncertainties and exogenous disturbances in the model, a dedicated breadboard has been developed and experimental validation of the control law has been carried out.

6.1 Introduction

Future space missions will increasingly rely on the use of large rotating payloads, in particular in the context of missions requiring high Earth observation capabilities [154, 155]. Accommodation restrictions, whether due to launcher fairing envelopes or limitations and constraints by the spacecraft, make it necessary to stow for launch and deploy in-orbit the payloads that are exceeding these restrictions due to their operational dimensions [156]. These systems require a careful design, because the unbalanced force and moment arising from inertial asymmetries can lead to the reduction of accuracy and stability of the spacecraft attitude and the induced vibrations can directly affect the quality of the collected data. For instance, one of the most challenging aspects of the Soil Moisture Active Passive (SMAP) observatory mission by NASA [154] was the design of the spinning reflector and beam assembly to minimize the impact of the inertial unbalances. Ground testing and characterization of the assembly was evaluated, but was considered impractical. It was determined that a program of detailed analysis and modeling (detailed to the screw, nut, washer and glue-line level) in conjunction with a rigorous hardware mass properties measurement process at the piece parts and sub-assembly level could effectively characterize the inertial properties of the system within the requirements dictated by the spacecraft dynamics [157]. However, the problems associated with unbalancing are magnified when the payload is large and, as a consequence, the operational life could become shorter and the operational reliability could degrade. In worst-case scenarios, the attitude control system may fail to stabilize the spacecraft or the unbalanced loads may damage the motor sustaining the spin motion, thereby undermining the outcome of the mission.

The future Copernicus Imaging Microwave Radiometry (CIMR) mission [158] aims to provide high spatial resolution microwave imaging radiometry measurements and derived products with global coverage and sub-daily revisit in the Polar regions and adjacent Seas. To guarantee a high spatio-temporal resolution, CIMR employs a conical scanning geometry with a spinning antenna reflector rotating about the nadir-pointing axis of the spacecraft to guarantee a high spatio-temporal resolution. The outward appearance of the CIMR spacecraft (see Figure 6.1) is dominated by a 9-meters large deployable mesh antenna that is mounted on the spun portion of the spacecraft and on the end of a 12-meters long boom. The CIMR payload involves components much larger than the SMAP ones (6-meters large reflector 6-meters long boom [159]) and, as a consequence, even an accurate manufacturing might not guarantee sufficient suppression of the inertial unbalances. For this reason, the effects of the unbalances might have to be corrected at commissioning by means of a dedicated balancing system [160].

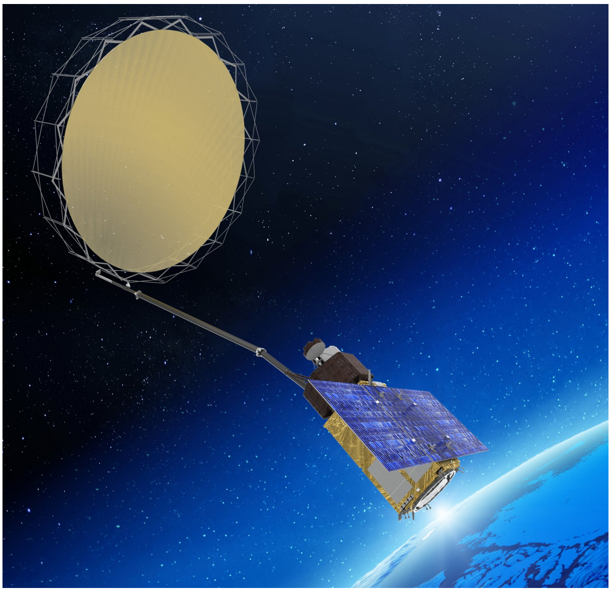


Figure 6.1: *CIMR spacecraft [161].*

6.2 Proposed architecture

In this Chapter, the concept introduced in the preliminary work [162] of an ABS to actively counteract unbalance effects in a rotor, is extended to explore a viable solution to the balancing of the CIMR rotating payload. The original ABS concept has been further analyzed, generalized and, then, experimentally validated on a simplified breadboard. Inspired by previous works [153, 163], the proposed ABS is based on:

- a set of movable masses, mounted on the rotor, and actuated through linear actuators;
- sensors, mounted on the spacecraft base, capable of measuring the components of the in-plane joint force and torque to be canceled;
- an electronic unit with the control system to process data from the sensors and to separately command the actuators to move the balancing masses to the target locations on the strokes.

After a detailed definition of a generic ABS configuration and the derivation of a suitable dynamical model, an analytical framework useful for the analysis of the system balancing capabilities is proposed. The proposed approach assumes a decoupled control architecture, wherein the attitude of the spacecraft is controlled through the actuators of the spacecraft base while the ABS controller tries to cancel out the effects of inertial unbalances in the rotor by moving the masses at suitable locations. In this regard, it has been recently shown in [164] that a properly tuned PD-like controller is capable of keeping a stable attitude in the presence of inertial unbalances in the rotating device and ensures convergence to the desired attitude for vanishing unbalances, thereby allowing for balancing operations to be carried out safely. In this work, assuming that the spacecraft controller keeps the attitude close to the desired one, we show that through suitable assumptions the rotating device can be considered as fixed on the ground when designing the control algorithm of the ABS. Due to the use of sensors on the (non-rotating) spacecraft base, we further show that the approximate dynamic model of the rotating device with the ABS is described by a perturbed LTP system. Based on such a model, we develop a HC law for the ABS to compensate for the effects of the unbalances.

6.3 ABS modeling and control problem formulation

This Section is devoted to presenting the problem addressed in this work. As mentioned in the Introduction, the system under investigation consists of a rotating device, henceforth called “rotor”, mounted on a spacecraft, which is referred to as “base”. To tackle the undesired effects associated with the inertial unbalance of the rotor, in this work we consider the use of an ABS made of a set of $N_m \in \mathbb{Z}_{>0}$ movable masses, mounted on the rotor, and by sensors, mounted on the spacecraft, capable of measuring the components of the joint force and torque perpendicular to the axis of rotation. The system comprising the rotor and the ABS will be referred to as the “payload” in the following.

6.3.1 Multibody system configuration and kinematics

To characterize the system configuration, several Cartesian reference frames must be introduced (see Figure 6.2):

- an inertial frame $\mathcal{F}_i := (O_i, \{i_1, i_2, i_3\})$ fixed at center of the Earth;
- a base-fixed frame $\mathcal{F}_b := (O_b, \{b_1, b_2, b_3\})$, with O_b being the CoM of the base. The difference vector between O_b and O_i resolved in $\mathcal{F}_i$ is denoted by $x_b \in \mathbb{R}^3$;
- a base-fixed frame $\mathcal{F}_a := (O_a, \{a_1, a_2, a_3\})$, with O_a being the attachment point between the base and the payload and $a_3 \in \mathbb{S}^2$ identifying the axis of rotation. The difference vector between O_a and O_b resolved in $\mathcal{F}_b$ is denoted by $h_a \in \mathbb{R}^3$;
- a payload-fixed frame $\mathcal{F}_r := (O, \{r_1, r_2, r_3\})$, with $O \equiv O_a$. The difference vector between O and O_i resolved in $\mathcal{F}_i$ is denoted by $x^r \in \mathbb{R}^3$. The difference vector between the payload CoM and O resolved in $\mathcal{F}_p$ is denoted by $x_G^r \in \mathbb{R}^3$.

Based on the above description, we are considering (without loss of generality) $a_i \equiv b_i \ \forall i \in \{1, 2, 3\}$, namely, that the frame $\mathcal{F}_a$ is aligned with $\mathcal{F}_b$. We now proceed by introducing the relevant quantities to describe the kinematics of the system. The attitude of the rotor is represented by

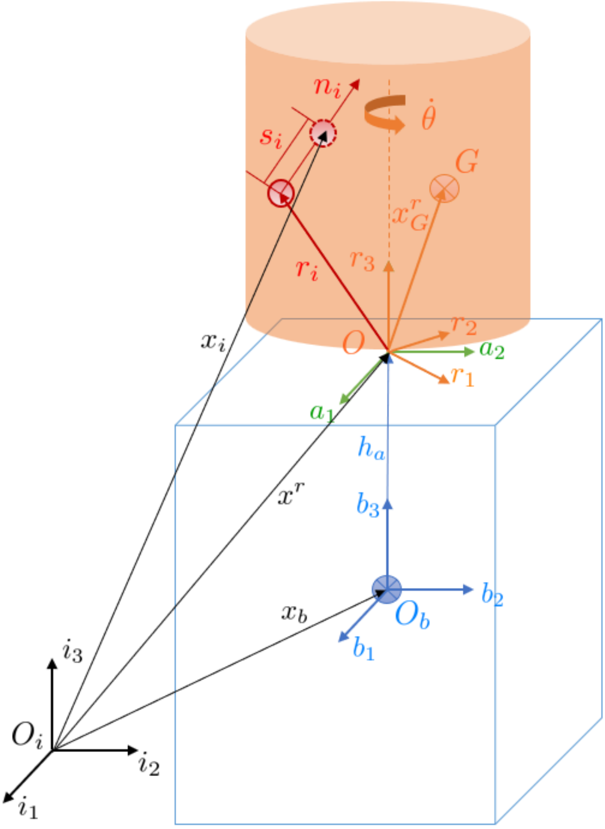


Figure 6.2: *Multibody spacecraft configuration and definition of the reference frames.*

the rotation matrix $R_r := \left[r_1 \; r_2 \; r_3 \right] \in \mathrm{SO}(3)$ which is described as a composition of two rotations:

$$R^r = R^b Q \tag{6.1}$$

where $R^b = \left[b_1 \; b_2 \; b_3 \right]$ is the rotation matrix describing the attitude of the spacecraft-fixed frame $\mathcal{F}_b$ with respect to the inertial frame $\mathcal{F}_i$ while

$$Q := \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

is the rotation matrix describing the attitude of the rotor-fixed frame $\mathcal{F}_r$ with respect to the spacecraft-fixed frame $\mathcal{F}_a$, parametrized in terms of the rotation angle $\theta \in [0, 2\pi)$ and the spin axis $e_3 = \left[0 \; 0 \; 1 \right]^\top$. The position of the *i-th* balancing mass expressed in $\mathcal{F}_i$, denoted by $x_i \in \mathbb{R}^3$, is a function of the corresponding relative displacement $s_i \in \mathbb{R}$ as follows:

$$x_i = x^r + r_i(s_i) \tag{6.3}$$

where $r_i(s_i) := R^r(\bar{r}_i + s_i n_i)$, with $\bar{r}_i := \left[\bar{x}_i \; \bar{y}_i \; \bar{z}_i \right]^\top \in \mathbb{R}^3$ is the zero location ($i.e. s_i = 0$) of the *i-th* balancing mass relative to O and expressed in $\mathcal{F}_r$, while $n_i \in \mathbb{S}^2$ is the unit vector assigning the corresponding displacement direction. As the strokes on which the actuated masses can move have finite length, we define $\underline{s}_i$ and $\bar{s}_i$ the lower and upper limit of the *i-th* stroke, respectively, such that $s_i \in [\underline{s}_i, \bar{s}_i]$.

The attitude kinematics of the rotor can be derived by differentiation of (6.1):

$$\begin{aligned} \dot{R}^r &= \dot{R}^b Q + R^b \dot{Q} = R^b S(\omega) Q + R^b Q S(e_3)\Omega \\ &= R^b Q(Q^\top S(\omega) Q + S(e_3)\Omega) = R^b Q S(Q^\top \omega + \Omega e_3) \end{aligned} \tag{6.4}$$

where $\omega \in \mathbb{R}^3$ is the (body) angular velocity of the spacecraft base, $\Omega := \dot{\theta}(t) \in \mathbb{R}$ is the angular rate of the rotor relative to the frame $\mathcal{F}_a$, which is assumed to be constant, *i.e.*, $\dot{\Omega}(t) = \ddot{\Omega}(t) \equiv 0 \; \forall t$. In deriving (6.4) we exploited the linearity property of the $S^{-1}(\cdot)$ map and the property $S^{-1}(R^\top S(\omega) R) = R^\top \omega \; \forall (R, \omega) \in \mathrm{SO}(3) \times \mathbb{R}^3$. From equation (6.4), one can define the payload angular velocity resolved in $\mathcal{F}_p$ as

$$\omega^r = Q^\top \omega + \Omega e_3, \tag{6.5}$$

which gives the compact expression $\dot{R}^r = R^r S(\omega^r)$. The inertial velocity of the center of mass of the rotor (point C in Figure 6.2) is obtained by differentiating the expression $x_G^r := x^r + R^r r_G^r$, which yields:

$$v_G^r = v_O + R^r S(\omega^r) x_G^r \tag{6.6}$$

where $v_O := \dot{x}_r$ denotes the inertial velocity of the attachment point O. The velocity of the *i-th* mass, resolved again in $\mathcal{F}_i$, is given by

$$v_i = v_O + R^r S(\omega^r)(\bar{r}_i + s_i n_i) + R^r \dot{s}_i n_i, \tag{6.7}$$

which comprises contributions from the motion of the attachment point (v_O), the overall angular velocity of the payload (ω^r) and the relative motion of the mass $(\dot{s}_i)$. At this point, all the information needed to characterize the proposed balancing system has been introduced and we can formally define the ABS system.

Definition 6.1. *Given a rotor-fixed frame $\mathcal{F}_r$, the ABS with N_m movable masses is defined by the tuple* $(m_1, \bar{r}_1, n_1, \underline{s}_1, \bar{s}_1, \ldots, m_{N_n}, \bar{r}_{N_n}, n_{N_n}, \underline{s}_{N_n}, \bar{s}_{N_n}) \in (\mathbb{R}_{>0} \times \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R} \times \mathbb{R})^{N_m}$.

6.3.2 Dynamics

The objective of the ABS is to move the balancing masses in such a way that the in-plane components of the reaction force and torque at the interface point, which are measured by sensors, are ideally canceled. To understand how the inertial unbalances of the rotor affect those quantities, we apply Euler-Newton's law to the rotor and the ABS system (*i.e.*, the overall payload):

$$\frac{dq}{dt} = f^e + f_O \tag{6.8}$$

$$\frac{dh_O}{dt} = -v_O \times q + \tau_O^e + \tau_O \tag{6.9}$$

where

$$\begin{aligned}
q &:= m^r v_G^r + \sum_{i=1}^{N_m} m_i v_i \\
&= m^p v_O + m^r R^r S(\omega^r) x_G^p + R^r \sum_{i=1}^{N_m} m_i \dot{s}_i n_i \qquad (6.10)\\
h_O &:= m^r S(R^r x_G^r) v_O + R^r J^r \omega + \sum_{i=1}^{N_m} r_i \times m_i v_i \\
&= m^p S(R^r x_G^p) v_O + R^r J^r \omega^r + \\
&+ R^r \sum_{i=1}^{N_m} m_i (S(\bar{r}_i + s_i n_i) S(\bar{r}_i + s_i n_i)^\top \omega^r + S(\bar{r}_i) n_i \dot{s}_i)) \qquad (6.11)
\end{aligned}$$

are the translational and the angular momentum, respectively. The exogenous force and torque have been split in contributions associated with the interface reaction at O (f_O and τ_O) and in contributions associated with environmental effects (f^e and τ_O^e). Herein, $J^r \in \mathbb{R}^{3\times 3}$ and $m^r \in \mathbb{R}_{>0}$ are the inertia matrix, with respect to O and expressed in $\mathcal{F}_r$, and the mass of the rotor, respectively, while $m^p := m^r + \sum_{i=1}^{N_m} m_i$ is the total mass of the payload and $x_G^p := \frac{1}{m^p}(m^r x_G^r + \sum_{i=1}^{N_m} m_i(\bar{r}_i + s_i n_i))$ is the location of the center of mass of the payload. Before proceeding, the following assumptions are considered to simplify the dynamical model in (6.8)-(6.9) to carry out the control design of the ABS controller.

Assumption 1. *Given any constant desired attitude $R_d \in \mathrm{SO}(3)$, the attitude control system of the spacecraft keeps the spacecraft attitude R^b close to the desired one R_d, so that one can consider $R^b(t) \equiv R_d$ $\forall t \geq 0$ when designing the ABS controller.*

Remark 30. *The above assumption is reasonable as the magnitude of the unbalance force and torque are dependent on the rotor speed, as shown next, and in a plausible scenario the rotor will be spun up slowly to avoid the risks associated with large unbalances. Moreover, based on the results of [164], a properly designed and tuned attitude control system for the spacecraft base can keep the attitude stable in the presence of a rotating unbalanced device, with guaranteed convergence to the desired at-*

titude for vanishing unbalances. Small mismatches with respect to the perfect stabilization condition embedded in Assumption 1 will be treated as disturbance terms acting on the nominal model.

Assumption 2. *The exogenous force and torque acting on the payload are given by*

$$f^e := f^g + \Delta f^e \tag{6.12}$$
$$\tau_O^e := R^r x_G^p \times f^g + \Delta \tau_O^e, \tag{6.13}$$

where $f^g := -\mu^{\oplus} m^p \frac{x_G^p}{\|x_G^p\|^3}$*, with* $\mu^{\oplus}$ *being the product of gravitational constant and the mass of the earth, while* $\Delta f^e \in \mathbb{R}^3$ *and* $\Delta \tau_O^e \in \mathbb{R}^3$ *represent unmodelled disturbances.*

Remark 31. *Assumption 2 implies that the main source of the exogenous force and torque is due to the gravity field, derived using a point mass model of the payload. Unmodelled (second order) effects associated with the actual mass distribution of the system and other environmental perturbations are included in the terms* Δf^e *and* $\Delta \tau^e$*, which will be dealt with a robust control design. It is worth underlying that the approximation introduced through Assumption 2 is standard when studying the dynamics of robotic space systems [165], and it essentially splits the overall motion of the spacecraft into the orbital motion of a point having the total mass of the system and a superimposed relative motion, as if the spacecraft is floating without gravity.*

As shown next, Assumption 1 and Assumption 2 allow for the development of a decoupled control architecture in which the ABS control design can be carried out independently of the spacecraft attitude control, the latter being considered in [164]. Let us focus on the translational dynamics first. By substituting (6.10) into the left-hand side of (6.8), we have:

$$\begin{aligned}\frac{dq}{dt} &= m^p \dot{v}_0 + R^r \sum_{i=1}^{N_m} m_i \left(S(\omega^r)\dot{s}_i n_i + \ddot{s}_i n_i\right) + \\ &\quad + \dot{R}^r \left(m^p S(\omega^r) r_C + \sum_{i=1}^{N_m} m_i (S(\omega^r)(\bar{r}_i + s_i n_i) + \dot{s}_i n_i) \right) + \\ &\quad + R^r \left(m^p S(\dot{\omega}^r) r_C + \sum_{i=1}^{N_m} m_i (S(\dot{\omega}^r)(\bar{r}_i + s_i n_i) + \dot{s}_i n_i) \right). \qquad (6.14)\end{aligned}$$

Based on Assumption 1, one also has $\omega(t) = \dot{\omega}(t) \equiv 0\ \forall t \geq 0$. Since the payload is rotating at constant rate, $\dot{\Omega}(t) \equiv 0\ \forall t \geq 0$, it follows that the angular velocity of the payload is identically zero as well, *i.e.*, $\dot{\omega}^r = \dot{\omega} + \dot{\Omega} e_3 \equiv 0$. Further assuming that the orbital acceleration times the mass of the payload is mostly balanced by the gravity force, *i.e.*, $m^p \dot{v}_O \approx f^g$ (we embed the mismatch into the disturbance term Δf^e), using (6.10), (6.14) and (6.12) the interface force can be approximated by the following expression:

$$\begin{aligned}f_O &= R^r S(\Omega e_3) \left(m^r S(\Omega e_3) r_C + \sum_{i=1}^{N_m} m_i \dot{s}_i n_i + S(\Omega e_3)(\bar{r}_i + s_i n_i) \right) + \\ &\quad + R^r \left(\sum_{i=1}^{N_m} m_i \ddot{s}_i n_i + S(\Omega e_3) \dot{s}_i n_i \right) + \Delta f^e. \qquad (6.15)\end{aligned}$$

We can now proceed similarly for the angular dynamics (6.9). To this end, consider the following equalities:

$$\begin{aligned} v_O \times q &= S(v_O)m^p v_O + S(v_O)(m^r R^r S(\omega^r)x_G^p + R^r \sum_{i=1}^{N_m} m_i \dot{s}_i n_i) \\ &= -S(m^p R^r S(\omega^r)x_G^p + R^r \sum_{i=1}^{N_m} m_i \dot{s}_i n_i)v_O, \qquad (6.16) \\ \frac{d}{dt}m^p S(R^r x_G^p)v_O &= m^p S(R^r S(\omega^r)x_G^p + R^r \dot{x}_G^p)v_O + m^p S(R^r x_G^p)\dot{v}_O \\ &= S(m^p R^r S(\omega^r)x_G^p + R^r \sum_{i=1}^{N_m} m_i \dot{s}_i n_i)v_O + S(R^r x_G^p)f^g, \qquad (6.17) \end{aligned}$$

where we exploited that $S(v_O)m^p v_O = m^p v_O \times v_O = 0$ and then used the approximation $m^p \dot{v}_O \approx f^g$, which was already introduced when deriving the approximated translational dynamics. Using (6.16) and (6.17), one obtains $v_O \times q + \frac{d}{dt}m^p S(R^r x_G^p) = S(R^r x_G^p)f^g$. Then, substituting (6.13) into (6.9), and performing some computations, we obtain the following expression for the approximated interface torque:

$$\begin{aligned} \tau_O = R^r S(\Omega e_3) \Bigg(& J^r \omega + \sum_{i=1}^{N_m} m_i S(\bar{r}_i + s_i n_i) S^\top (\bar{r}_i + s_i n_i)\Omega e_3 + \\ & + \sum_{i=1}^{N_m} m_i S(\bar{r}_i) n_i \dot{s}_i \Bigg) + R^r \Bigg(\sum_{i=1}^{N_m} m_i \Big(S(n_i) S^\top (\bar{r}_i + s_i n_i) + \\ & + S(\bar{r}_i + s_i n_i) S^\top (n_i) \Big) \Omega e_3 \dot{s}_i + \sum_{i=1}^{N_m} m_i S(\bar{r}_i) n_i \ddot{s}_i \Bigg) + \Delta\tau_O^e \qquad (6.18) \end{aligned}$$

where we considered the same approximations adopted for the translational dynamics stemming from Assumption 1. As the ABS setup described in the Introduction makes use of sensors that measure the interface force and torque in frame $\mathcal{F}_a$, it is necessary to express (6.15) and (6.18) in such a frame. To this end, multiplying both sides of (6.15)-(6.18) by $R^{b\top}$, the in-plane components of f_O and τ_O in $\mathcal{F}_a$ can be compactly written as follows:

$$\begin{bmatrix} f^a_{O_1} \\ f^a_{O_2} \\ \tau^a_{O_1} \\ \tau^a_{O_2} \end{bmatrix} = \underbrace{\begin{bmatrix} c(\theta(t)) & -s(\theta(t)) & 0 & 0 \\ s(\theta(t)) & c(\theta(t)) & 0 & 0 \\ 0 & 0 & c(\theta(t)) & -s(\theta(t)) \\ & 0 & s(\theta(t)) & c(\theta(t)) \end{bmatrix}}_{Z(t)} \begin{bmatrix} f^r_{O_1} \\ f^r_{O_2} \\ \tau^r_{O_1} \\ \tau^r_{O_2} \end{bmatrix}, \qquad (6.19)$$

where $c(\cdot)$, $s(\cdot)$ is a shorthand notation for $\cos(\cdot)$, $\sin(\cdot)$, respectively, and the expressions of f^r_O, τ^r_O are reported in the following equations:

$$\begin{aligned} f^r_{O_1} = {} & -\Omega^2 \left(m^r x^r_G + \sum_{i=1}^{N_m} m_i (\bar{x}_i + s_i e_1^\top n_i) \right) + 2\Omega \sum_{i=1}^{N_m} m_i e_2^\top n_i \dot{s}_i + \\ & + \sum_{i=1}^{N_m} m_i e_1^\top n_i \ddot{s}_i + \Delta f_1^{er} \qquad (6.20) \end{aligned}$$

$$\begin{aligned} f^r_{O_2} = {} & -\Omega^2 \left(m^r y^r_G + \sum_{i=1}^{N_m} m_i (\bar{y}_i + s_i e_2^\top n_i) \right) - 2\Omega \sum_{i=1}^{N_m} m_i e_1^\top n_i \dot{s}_i + \\ & + \sum_{i=1}^{N_m} m_i e_2^\top n_i \ddot{s}_i + \Delta f_2^{er} \qquad (6.21) \end{aligned}$$

$$\begin{aligned} \tau^r_{O_1} = {} & -\Omega^2 \left(J^r_{23} - \sum_{i=1}^{N_m} m_i (\bar{y}_i + s_i e_2^\top n_i)(\bar{z}_i + s_i e_3^\top n_i) \right) + \\ & + \sum_{i=1}^{N_m} m_i \begin{bmatrix} 0 & -\bar{z}_i & \bar{y}_i \end{bmatrix} n_i \ddot{s}_i + \Omega \sum_{i=1}^{N_m} m_i \Big(\begin{bmatrix} -\bar{z}_i & 0 & \bar{x}_i \end{bmatrix} n_i + \\ & + e_1^\top S(n_i) \begin{bmatrix} -\bar{y}_i - s_i e_2^\top n_i & \bar{x}_i + s_i e_1^\top n_i & 0 \end{bmatrix}^\top + \\ & + \begin{bmatrix} 0 & -\bar{z}_i - s_i e_3^\top n_i & \bar{y}_i + s_i e_2^\top n_i \end{bmatrix} S^\top(n_i) e_3 \Big) \dot{s}_i + \Delta \tau^{er}_{O_1} \qquad (6.22) \end{aligned}$$

$$\begin{aligned}\tau^r_{O_2} = \Omega^2 &\left(J^r_{13} - \sum_{i=1}^{N_m} m_i (\bar{x}_i + s_i e_1^\top n_i)(\bar{z}_i + s_i e_3^\top n_i) \right) + \\ &+ \sum_{i=1}^{N_m} m_i \begin{bmatrix} \bar{z}_i & 0 & -\bar{x}_i \end{bmatrix} n_i \ddot{s}_i + \Omega \sum_{i=1}^{N_m} m_i \Big(\begin{bmatrix} 0 & -\bar{z}_i & \bar{y}_i \end{bmatrix} n_i + \\ &+ e_2^\top S(n_i) \begin{bmatrix} -\bar{y}_i - s_i e_2^\top n_i & \bar{x}_i + s_i e_1^\top n_i & 0 \end{bmatrix}^\top + \\ &+ \begin{bmatrix} \bar{z}_i + s_i e_3^\top n_i & 0 & -\bar{x}_i - s_i e_1^\top n_i \end{bmatrix} S^\top(n_i) e_3 \Big) \dot{s}_i + \Delta\tau^{er}_{O_1}. \qquad (6.23)\end{aligned}$$

Remark 32. *Given the approximations introduced to derive the mathematical model* (6.19)*, the dynamics of the rotating device under an attitude controlled base has been reduced to the one of a ground fixed system, thereby effectively enabling a decoupled design for the attitude controller of the base and the control system of the ABS and giving the possibility of testing the ABS on ground by building a suitable breadboard, as discussed in Section 6.5.1.*

6.3.3 Control problem formulation

To describe the balancing control problem based on the dynamic model derived in the previous section, we first characterize the necessary control authority of the ABS to achieve a balanced configuration. Without loss of generality, assume that the rotor and the ABS unbalances for $s_i = 0$ can be split in nominal and perturbation terms as follows:

$$m^r x^r_G + \sum_{i=1}^{N_m} m_i \bar{x}_i = \bar{S}_1 + \Delta S_1 = \Delta S_1 \qquad (6.24)$$

$$J^r_{13} - \sum_{i=1}^{N_m} m_i \bar{x}_i \bar{z}_i = \bar{J}_{13} + \Delta J_{13} = \Delta J_{13} \qquad (6.25)$$

$$m^r y^r_G + \sum_{i=1}^{N_m} m_i \bar{y}_i = \bar{S}_2 + \Delta S_2 = \Delta S_2 \qquad (6.26)$$

$$J^r_{23} - \sum_{i=1}^{N_m} m_i \bar{y}_i \bar{z}_i = \bar{J}_{23} + \Delta J_{23} = \Delta J_{23} \qquad (6.27)$$

where, given $i = 2,3$, $\bar{S}_i$ and ΔS_i denote the static moment and the corresponding perturbation and similarly $\bar{J}_{i3}$ and ΔJ_{i3} denote the nominal inertia moment and the corresponding perturbation. The idea behind the decomposition in (6.24)-(6.27) is that the rotor and the ABS are designed to be self-balanced ($\bar{S}_1 = \bar{S}_2 = 0, \bar{J}_{13} = \bar{J}_{23} = 0$) but that there will be unavoidably residual unbalances in practice. Of note, for constant unbalances, balanced equilibrium conditions ($\dot{s}_i = \ddot{s}_i = f^r_{O_1} = f^r_{O_2} = \tau^r_{O_1} = \tau^r_{O_2} = 0$) can be obtained provided that the ABS satisfies the following assumption.

Assumption 3. *Given positive scalars* $\bar{\Delta}_i \in \mathbb{R}_{>0}$, $i = 1,\dots,4$, *for any* $\begin{bmatrix}\Delta S_1 & \Delta S_2 & \Delta J_{13} & \Delta J_{23}\end{bmatrix}^\top \in \Omega_\Delta := \left\{\Delta \in \mathbb{R}^4 : |\Delta_i| \le \bar{\Delta}_i, i = 1,\dots,4\right\}$, *the ABS* $(m_i, \bar{r}_i, n_i, \underline{s}_i, \bar{s}_i)$, $i = 1,\dots,N_m$, *is such that the system of algebraic equation*

$$\sum_{i=1}^{N_m} m_i e_1^\top n_i s_i = -\Delta S_1 \tag{6.28}$$

$$\sum_{i=1}^{N_m} m_i e_2^\top n_i s_i = -\Delta S_2 \tag{6.29}$$

$$\sum_{i=1}^{N_m} m_i \left((e_2^\top n_i)(e_3^\top n_i)s_i^2 \bar{y}_i(e_3^\top n_i) + \bar{z}_i(e_2^\top n_i)s_i\right) = \Delta J_{23} \tag{6.30}$$

$$\sum_{i=1}^{N_m} m_i \left((e_1^\top n_i)(e_3^\top n_i)s_i^2 \bar{x}_i(e_3^\top n_i) + \bar{z}_i(e_1^\top n_i)s_i\right) = \Delta J_{13} \tag{6.31}$$

admits at least one feasible solution, i.e.*, there exist positions of the balancing masses solving* (6.28)-(6.31) *and such that* $s_i \in [\underline{s}_i, \bar{s}_i]$ $\forall i = 1,\dots,N_m$.

Note that system (6.28)-(6.31) is obtained by substituting equations (6.24)-(6.27) in (6.20)-(6.23) and then by setting to zero the terms that multiply Ω^2. The above assumption is necessary to have a feasible control problem, as more formally defined at the end of this Section. This system can be rewritten in matrix form as

$$\begin{bmatrix} \Delta^1 J_{13}^2 & \cdots & \Delta^M J_{13}^2 \\ 0 & \cdots & 0 \\ \Delta^1 J_{23}^2 & \cdots & \Delta^M J_{23}^2 \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} s_1^2 \\ \vdots \\ s_M^2 \end{bmatrix} + \begin{bmatrix} \Delta^1 J_{13}^1 & \cdots & \Delta^M J_{13}^1 \\ \Delta^1 S_2 & \cdots & \Delta^M S_2 \\ \Delta^1 J_{23}^1 & \cdots & \Delta^M J_{23}^1 \\ \Delta^1 S_1 & \cdots & \Delta^M S_1 \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} = \begin{bmatrix} \Delta J_{13} \\ -\Delta S_2 \\ \Delta J_{23} \\ -\Delta S_1 \end{bmatrix} \tag{6.32}$$

where for the *i-th* mass the following definitions are valid

$$\begin{aligned} \Delta^i J_{13}^2 &= m_i(n_{x_i} n_{z_i}) & \Delta^i J_{23}^2 &= m_i(n_{y_i} n_{z_i}) \\ \Delta^i J_{13}^1 &= m_i(n_{x_i} z_{n_i} + n_{z_i} x_{n_i}) & \Delta^i J_{23}^1 &= m_i(n_{y_i} z_{n_i} + n_{z_i} y_{n_i}) \\ \Delta^i S_1 &= m_i n_{y_i} & \Delta^i S_2 &= m_i n_{x_i}. \end{aligned} \tag{6.33}$$

Important considerations can now be made:

- Second order terms vanish when n_i is on the xy plane or when $n_i = [0\ 0\ 1]^T$.

$$\Delta^i J_{13}^2 = m_i(n_{x_i} n_{z_i}) = 0 \quad \Delta^i J_{23}^2 = m_i(n_{y_i} n_{z_i}) = 0.$$

- First order terms vanish when

$$\begin{aligned} \Delta^i J_{13}^1 = m_i(n_{x_i} z_{n_i} + n_{z_i} x_{n_i}) &\longrightarrow n_{x_i} = -\frac{x_{n_i}}{z_{n_i}} n_{z_i} \text{ with } z_{n_i} \neq 0 \\ \Delta^i J_{23}^1 = m_i(n_{y_i} z_{n_i} + n_{z_i} y_{n_i}) &\longrightarrow n_{y_i} = -\frac{y_{n_i}}{z_{n_i}} n_{z_i} \text{ with } z_{n_i} \neq 0 \\ \Delta^i S_1 = m_i n_{y_i} &\longrightarrow n_{y_i} = 0 \\ \Delta^i S_2 = m_i n_{x_i} &\longrightarrow n_{x_i} = 0. \end{aligned}$$

- When n_i is aligned with a body axis in the xy plane, *e.g.*, $n_i = [0\ 1\ 0]^T$

$$\begin{aligned} \Delta^i J_{13}^1 &= m_i(n_{x_i} z_{n_i} + n_{z_i} x_{n_i}) = 0 \\ \Delta^i J_{23}^1 &= m_i(n_{y_i} z_{n_i} + n_{z_i} y_{n_i}) = m_i z_{n_i} \\ \Delta^i S_1 &= m_i n_{y_i} = m_i \\ \Delta^i S_2 &= m_i n_{x_i} = 0. \end{aligned}$$

In this case to have a well-defined problem we need at least two masses at different coordinates along the z-axis

$$\begin{aligned} m_1 z_1 s_1 + m_2 z_2 s_2 &= \Delta J_{23} \\ m_1 s_1 + m_2 s_2 &= \Delta S_1. \end{aligned}$$

- When n_i is aligned with the z-axis, *e.g.*, $n_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$

$$\begin{aligned} \Delta^i J_{13}^1 &= m_i(n_{x_i} z_{n_i} + n_{z_i} x_{n_i}) = m_i x_{n_i} \\ \Delta^i J_{23}^1 &= m_i(n_{y_i} z_{n_i} + n_{z_i} y_{n_i}) = m_i y_{n_i} \\ \Delta^i S_1 &= m_i n_{y_i} = 0 \\ \Delta^i S_2 &= m_i n_{x_i} = 0. \end{aligned}$$

It can be seen that masses moving in the xy plane are needed to balance the static moments.

Exploiting these considerations and by defining $w := \begin{bmatrix} f_{O_1}^r & f_{O_2}^r & \tau_{O_1}^r & \tau_{O_2}^r \end{bmatrix}^\top$ and by means of (6.24)-(6.27), equations (6.20)-(6.23) can be compactly written as[1]:

$$w = \sum_{i=1}^{N_m} C_{abs}^i y_a^i + D_{abs} d \tag{6.34}$$

where $y_a^i = \begin{bmatrix} s_i & \dot{s}_i & \ddot{s}_i \end{bmatrix}^\top$ and $d := \begin{bmatrix} \Delta S_1 & \Delta S_2 & \Delta J_{13} & \Delta J_{23} \end{bmatrix}^\top$ and the exogenous disturbances have been neglected for the purpose of deriving the nominal model for control. The exact expressions of C_{abs}^i and D_{abs} can be derived from (6.20)-(6.23) but they are omitted here for space reasons. The ABS system includes N_m position-controlled linear actuators to assign the motion of the balancing masses. Assuming a linear behavior, we can compactly write the actuators dynamics as

$$\dot{x}_a = A_a x_a + B_a u, \qquad y_a = C_a x_a + D_a u, \tag{6.35}$$

where $x_a = \begin{bmatrix} x_a^1 & \cdots & x_a^{N_m} \end{bmatrix}^\top \in \mathbb{R}^{N_m N_a}$ is a vector including all the states of the N_m actuators, $y_a = \begin{bmatrix} y_a^i & \cdots & y_a^{N_m} \end{bmatrix}^\top \in \mathbb{R}^{3N_m}$ is a vector collecting the outputs defined in (6.34) and $u = \begin{bmatrix} u_1 & \cdots & u_{N_m} \end{bmatrix} \in \mathbb{R}^{N_m}$ is the vector of control

[1] To handle a linear problem the n_i are considered always on the xy plane or $n_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ in order to eliminate the second order terms. This assumption holds in the entire work and in the following it is implied.

variables, *i.e.*, the desired positions of the ABS masses. Finally, $A_a = \text{blkdiag}(A_a^i)$, $B_a = \text{blkdiag}(B_a^i)$, $C_a = \text{blkdiag}(C_a^i)$ and $D_a = \text{blkdiag}(D_a^i)$ are block diagonal matrices formed from the quadruples $(A_a^i, B_a^i, C_a^i, D_a^i)$ characterizing the *i-th* actuator dynamics, which has order N_a. The in-plane torque and force in the spacecraft-base frame (equation (6.19)) are measured by load sensors, for which we again assume a linear behavior, given by:

$$\begin{aligned} \dot{x}_s &= A_s x_s + B_s Z(t) w \\ &= A_s x_s + B_s Z(t) \left(C_{abs} C_a x_a + C_{abs} D_a u + D_{abs} d \right) \qquad (6.36) \\ y_s &= C_s x_s \qquad (6.37) \end{aligned}$$

where $C_{abs} = \left[C_{abs}^1 \cdots C_{abs}^{Nm} \right]$, $Z(t)$ is defined in (6.19) and $A_s = \text{blkdiag}(A_s^i)$, $B_s = \text{blkdiag}(B_s^i)$, $C_s = \text{blkdiag}(C_s^i)$ are block diagonal matrices formed from the quadruples (A_s^i, B_s^i, C_s^i) characterizing the *i-th* load sensor dynamics, which has order N_s. By defining $x = \left[x_a \; x_s \right]^\top$ and $y = y_s \in \mathbb{R}^4$, the overall system can be written in state-space form as follows:

$$\dot{x} = A(t)x + B_u(t)u + B_d(t)d \qquad y = \begin{bmatrix} 0 & C_s \end{bmatrix} x \qquad (6.38)$$

where

$$A(t) = \begin{bmatrix} A_a & 0 \\ B_s Z(t) C_{abs} C_a & A_s \end{bmatrix}, \quad B_u(t) = \begin{bmatrix} B_a \\ B_s Z(t) C_{abs} D_a \end{bmatrix},$$
$$B_d(t) = \begin{bmatrix} 0 \\ B_s Z(t) D_{abs} \end{bmatrix}. \qquad (6.39)$$

By referring to system (6.38), the problem that we address in this work can be formulated as the design of an output feedback controller for u (desired positions of the ABS masses) such that the measured interface loads y are minimized for all inertial unbalances satisfying $|d_i| \leq \bar{\Delta}_i$, $i = 1, \ldots, 4$, where $\bar{\Delta}_i \in \mathbb{R}_{>0}$ are the minimum unbalances that the ABS should be able to compensate for according to Assumption 3.

6.4 Control law design

In this Section, we design a HC algorithm to command the positions of the actuated masses in such a way that the effects of rotor unbalance are significantly reduced. Specifically, we define the T-matrix representation of the LTP model of the rotor and the ABS in equation (6.38), following the approaches in [150, 141]. Then, we apply the optimal LQ-based HC algorithm described in Section 5.5.

6.4.1 T-matrix definition

It can be noticed that the rotor will be subject to a proper, steady-state harmonic control input at frequency Ω whenever the control vector u is *constant* (the same consideration holds for the disturbance d). This implies that we only have to study the response of the periodic model to a EMP input with $s = 0$, *i.e.*, we only have to compute the input/output operators $\mathcal{G}_u(0)$ and $\mathcal{G}_d(0)$. Given a constant input $u(t) = U_0$, the vector $\mathcal{U}$ corresponding to $u(t) = U_0$ is given by

$$\mathcal{U}^T = \begin{bmatrix} \cdots & 0 & 0 & U_0^T & 0 & 0 & \cdots \end{bmatrix} \tag{6.40}$$

and the steady-state response $\mathcal{Y}$ of the periodic system without the disturbance is given by

$$\mathcal{Y} = \mathcal{G}_u(0)\,\mathcal{U} \tag{6.41}$$

which can be equivalently written as

$$\begin{bmatrix} \vdots \\ Y_{-2} \\ Y_{-1} \\ Y_0 \\ Y_1 \\ Y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & G^u_{-2,-2} & G^u_{-2,-1} & G^u_{-2,0} & G^u_{-2,1} & G^u_{-2,2} & \cdots \\ \cdots & G^u_{-1,-2} & G^u_{-1,-1} & G^u_{-1,0} & G^u_{-1,1} & G^u_{-1,2} & \cdots \\ \cdots & G^u_{0,-2} & G^u_{0,-1} & G^u_{0,0} & G^u_{0,1} & G^u_{0,2} & \cdots \\ \cdots & G^u_{1,-2} & G^u_{1,-1} & G^u_{1,0} & G^u_{1,1} & G^u_{1,2} & \cdots \\ \cdots & G^u_{2,-2} & G^u_{2,-1} & G^u_{2,0} & G^u_{2,1} & G^u_{2,2} & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ 0 \\ 0 \\ U_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{6.42}$$

Focusing on the first harmonics, from equation (6.42) we have that

$$\begin{bmatrix} Y_{-1} \\ Y_1 \end{bmatrix} = \begin{bmatrix} G^u_{-1,0} \\ G^u_{1,0} \end{bmatrix} U_0. \tag{6.43}$$

Converting the harmonics of the output from exponential to trigonometric form, the following expression is obtained:

$$\begin{bmatrix} Y_{1c} \\ Y_{1s} \end{bmatrix} = TU_0, \tag{6.44}$$

where

$$T := 2 \begin{bmatrix} \text{Real}[G^u_{1,0}] \\ \text{Imag}[G^u_{1,0}] \end{bmatrix} \in \mathbb{R}^{2n_y \times n_u} \tag{6.45}$$

is the T-matrix.

6.4.2 *LQ*-based HC algorithm

Let us assume that the update of the input $u(t)$ (*i.e.*, the commanded position of the actuated masses) is at specific times $t_k = k\Delta t$, where Δt is the time interval between consecutive updates, during which the plant output is assumed to reach a steady level, and where $k \in \mathbb{Z}_{\geq 0}$ is the discrete-time index. Exploiting equation (6.45), the steady-state response of the system (6.38) to both inputs and disturbances can be expressed by the following discrete time mathematical model:

$$Y_1(k) = Tu(k) + Wd, \tag{6.46}$$

where $Y_1 \in \mathbb{R}^{2n_y}$ is the vector of the first harmonics of measured outputs, u is the vector of the desired positions of the ABS masses in the rotating frame and d is the vector of the inertial unbalances defined below equation (6.34). The matrix $W \in 2n_y \times n_d$ is a constant matrix defined in similar fashion as

$$W := 2 \begin{bmatrix} \text{Real}[G^d_{1,0}] \\ \text{Imag}[G^d_{1,0}] \end{bmatrix}. \tag{6.47}$$

For system (6.46), HC is a viable approach for the design of a feedback controller aiming at the minimization of the measured interface loads. In this work we apply the control law described in Section 5.5, namely:

$$u(k+1) = KTu(k) - KY_1(k) \tag{6.48}$$

where $K = (T^\top QT + R)^{-1}T^\top Q$ is a gain matrix.

6.5 Simulation and experimental results

By carefully inspecting equations (6.20)-(6.23), one sees that the balancing problem can be decoupled in two sub-problems, one for the xz plane and one for the yz plane, provided that the ABS is made of a suitable set of strokes directed along the coordinate axes. Based on this idea, a breadboard representative of a single-plane ABS has been designed and built to test the proposed control design.

6.5.1 ABS breadboard design and modeling

The breadboard (see Figure 6.3 and Figure 6.10) has been designed with an ABS made of three movable counter-masses lying in the xz plane of the rotor frame: a central mass, which can be moved along the x-axis $(n_1 = e_1)$, and two side masses, which can be moved along the z-axis $(n_{2,3} = e_3)$. While two masses would have been enough to balance the system, three masses give more flexibility and allow to easily obtain a self-balanced ABS. The main components of the breadboard are:

- the rotor structure, with the possibility of applying a known unbalance (static and dynamic) through the placement of four masses at the corners;
- three guides, each endowed with a linear actuator, to move the counter-mass;
- three load cell sensors in an equilateral triangle configuration forming a dynamometer;
- the stator structure;
- the rotary actuator, composed of motor, gearbox, and a differential digital encoder to rotate the rotor shaft;
- the slip ring, to guarantee the electrical connection between rotor and stator;
- the power supply, acquisition system, conditioners, and controllers.

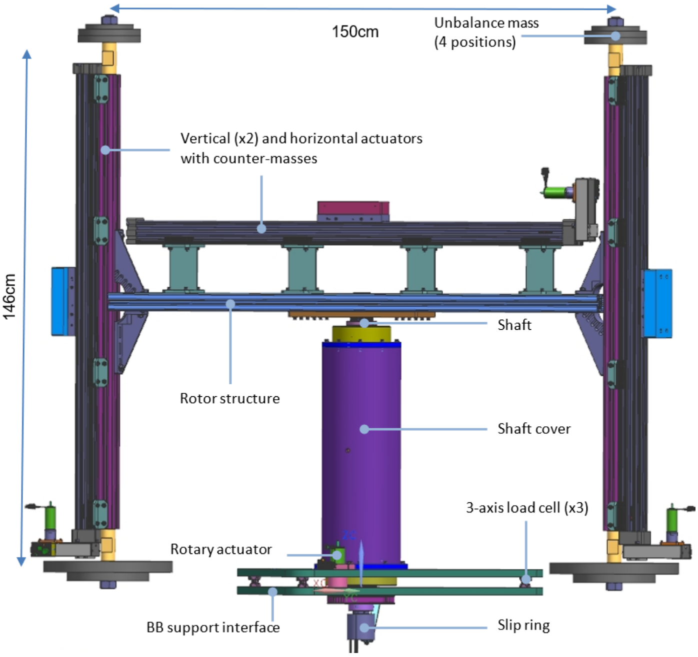


Figure 6.3: *Multibody model of the breadboard.*

The ABS configuration has been designed to be self-balanced, *i.e.*, when the counter masses are in their zero positions ($s_i = 0$) no unbalance is added to the system (see again equations (6.24)-(6.27)). The dynamometer allows the measurement of the interface force along the $x-$axis and the torque about the y-axis. The nominal angular rate of the rotor is $\Omega = 7.8\,\text{rpm}$. For the considered setup, the reaction force $f^r_{O_1}$ (6.20) and torque $\tau^r_{O_2}$ (6.23) reduce to the following expressions:

$$f^r_{O_1} = -\,\Omega^2 \left(\Delta S_1 + m_1 s_1\right) + m_1 \ddot{s}_1 - f^e_1 \tag{6.49}$$

$$\begin{aligned} \tau^r_{O_2} &= \Omega^2 \left(\Delta J^r_{13} - m_1 \bar{z}_1 s_1 - m_2 \bar{x}_2 s_2 - m_3 \bar{x}_3 s_3\right) \\ &\quad + \sum_{i=1}^{3} m_i \left[\begin{smallmatrix} \bar{z}_i & 0 & -\bar{x}_i \end{smallmatrix}\right] n_i \ddot{s}_i - \tau^e_{O_1} \end{aligned} \tag{6.50}$$

where $\Delta S_1 = \sum_{i=1}^{4} m^s_i \bar{x}^s_i$ and $\Delta J^r_{13} = \sum_{i=1}^{4} -m^s_i \bar{x}^s_i \bar{z}^s_i$. Herein $m^s_i \in \mathbb{R}_{>0}$ is the $i-th$ unbalance mass and $x^s_i \in \mathbb{R}$ and $z^s_i \in \mathbb{R}$ represent the *i-th* unbalance mass and its coordinates in the xz plane, respectively. The exogenous force component f_1^e is associated with disturbances acting on the platform, such as, *e.g.*, friction between the rotor and the stator, small inclination of the structure with respect to the gravity direction, residual static unbalance in the structure, *etc.*. Instead, the exogenous torque mainly comprises the torque associated with gravity, *i.e.*,

$$\tau_{O_1}^e \approx \sum_{i=1}^{4} m_i^s g x_i + m^r g x_G^r + m_1 g s_1, \tag{6.51}$$

and the torque associated with an unavoidable residual unbalance in the structure.

Remark 33. *While the gravity-induced torque* (6.51) *could be considered as a disturbance to be balanced by the harmonic controller (by including the term $m_1 g$ in matrix C_{abs}^1 in equation* (6.34)*), it was decided to remove it from the measured signal before applying the HC algorithm to better replicate on-orbit operations.*

At this point, the dynamic model of the breadboard can be written in the same form as (6.38). Specifically, the matrices in equation (6.34), entering the dynamics through (6.39), are given by:

$$C_{abs}^1 = m_1 \begin{bmatrix} -\Omega^2 & 0 & 1 \\ -\Omega^2 \bar{z}_1 & 0 & \bar{z}_1 \end{bmatrix}, \quad C_{abs}^2 = -m_2 \begin{bmatrix} 0 & 0 & 1 \\ \Omega^2 \bar{x}_2 & 0 & \bar{x}_2 \end{bmatrix}, \tag{6.52}$$

$$C_{abs}^2 = -m_3 \begin{bmatrix} 0 & 0 & 0 \\ \Omega^2 \bar{x}_3 & 0 & \bar{x}_3 \end{bmatrix}, \quad D_{abs} := \begin{bmatrix} -\Omega^2 & 0 \\ 0 & \Omega^2 \end{bmatrix}. \tag{6.53}$$

For simplicity, we consider critically-damped second order systems for both the actuators and the sensors dynamics with unit DC-gain (which fully define matrices $A_a, B_a, C_a, D_a, A_s, B_s, C_s$). Matrix $A(t)$ defined as in (6.39), can be expanded in a complex Fourier series $A(t) = \sum_{m=-\infty}^{\infty} A_m e^{jm\Omega t}$: since $Z(t) = \begin{bmatrix} \cos(\theta(t)) & 0 \\ 0 & \cos(\theta(t)) \end{bmatrix}$ for the considered case, only the terms A_0, A_1 and A_{-1} are different from the null matrix. Expanding in the same fashion $B_u(t)$ and $C(t)$, we consider the following finite-dimensional

Toeplitz matrices

$$\begin{aligned}
\mathcal{A} &= \begin{bmatrix} A_0 & A_{-1} & 0 \\ A_1 & A_0 & A_{-1} \\ 0 & A_1 & A_0 \end{bmatrix} & \mathcal{B}_u &= \begin{bmatrix} B_{u_0} & 0 & 0 \\ 0 & B_{u_0} & 0 \\ 0 & 0 & B_{u_0} \end{bmatrix} \\
\mathcal{B}_d &= \begin{bmatrix} B_{d_0} & B_{d_{-1}} & 0 \\ B_{d_1} & B_{d_0} & B_{d_{-1}} \\ 0 & B_{d_1} & B_{d_0} \end{bmatrix} & \mathcal{C} &= \begin{bmatrix} C_0 & 0 & 0 \\ 0 & C_0 & 0 \\ 0 & 0 & C_0 \end{bmatrix}
\end{aligned} \tag{6.54}$$

to compute the T-matrix as given by equation (6.45). The choice of the number of block rows used to approximate the infinite dimensional matrices in (6.41) will affect the numerical accuracy of the results (see [166] for an analysis of the effect of such a truncation in the study of frequency response operators). We found that using 3 × 3 block-matrices as in (6.54) was sufficient for our purposes.

6.5.2 ABS sizing and balancing capabilities

The proposed ABS is well-defined in the sense of Assumption 3 since the system

$$\begin{bmatrix} m_1 & 0 & 0 \\ m_1\bar{z}_1 & m_2\bar{x}_2 & m_3\bar{x}_3 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} -\Delta S_1 \\ \Delta J_{13} \end{bmatrix}, \tag{6.55}$$

derived from equations (6.49)-(6.50), admits ∞^1 solutions provided that $\bar{x}_2$ and $\bar{x}_3$ are different from zero, *i.e.*, the system is overactuated. Of course, due to the finite length of the strokes ($\bar{s}_i = -\underline{s}_i = 0.5\,\mathrm{m}$), the maximum static and dynamic unbalances which can be compensated are bounded. The balancing masses and their locations have been selected to counteract all the unbalances in the set $\Omega_\Delta := \left\{\Delta S_1, \Delta J_{13} \in \mathbb{R}^2 : |\Delta S_1| \le \overline{\Delta S}_1,\ |\Delta J_{13}| \le \overline{\Delta J}_{13}\right\}$, where $\overline{\Delta S}_1 = 2.2\,\mathrm{kgm}$ and $\overline{\Delta J}_{13} = 4.5\,\mathrm{kgm}^2$. The resulting ABS is characterized by the following parameters: $\bar{z}_1 = 0.98\,\mathrm{m}$, $\bar{z}_2 = \bar{z}_3 = 0.72\,\mathrm{m}\ \bar{x}_2 = -\bar{x}_3 = 0.87\,\mathrm{m}$, $m_1 = 4.6\,\mathrm{kg}$ and $m_2 = m_3 = 8.1\,\mathrm{kg}$. The balancing capabilities of the proposed sizing can be evaluated from (6.55) by computing the range of the map

$$(s_1, s_2, s_3) \mapsto \begin{bmatrix} m_1 & 0 & 0 \\ m_1\bar{z}_1 & m_2\bar{x}_2 & -m_3\bar{x}_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \tag{6.56}$$

in the feasible set, *i.e.*, for $s_i \in [-0.5,\ 0.5]$m, which corresponds to the area within the black parallelogram shown in Figure 6.4. As can be seen in the Figure, the red rectangle representing the set Ω_Δ is strictly inside the parallelogram with at least a 5% margin, thereby satisfying the balancing requirements.

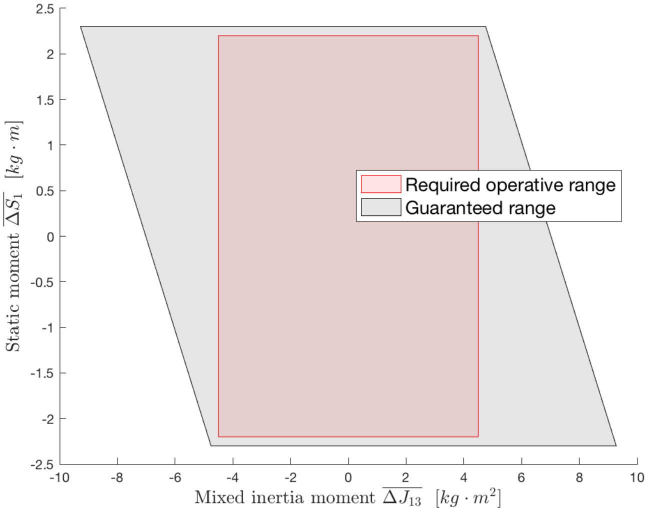


Figure 6.4: *Operative range of the breadboard.*

6.5.3 Control law tuning

In this Section, we present the procedure employed to tune the weighting matrices Q and R in the gain matrix (7.2). The procedure has been performed by requiring that the measured interface loads at the nominal rotor speed ($\Omega = 7.8\,$rpm) are below given bounds, specifically $|f^r_{O_1}| < \Delta S_1^{res}\Omega^2 = 0.07\,$N and $|\tau^r_{O_2}| < \Delta J_{13}^{res}\Omega^2 = 0.2\,$Nm, which correspond to maximum allowed residual unbalances $\Delta S_1^{res} \approx 0.1\,$kgm and $\Delta J_{13}^{res} \approx 0.3\,$kgm^2. As explained in Remark 28, the selection of the weighting matrices Q, R in the T-matrix algorithm (7.1) is based on the trade-off between performance and robustness. For simplicity, we consider $R = I_3$ and $Q = \alpha\ I_4$ with $\alpha := 1/\rho$ a scalar which is the only parameter to tune. Considering the worst-case scenario, namely when $(\Delta S_1, \Delta J_{13}) =$

$\pm(\overline{\Delta S}_1, \overline{\Delta J}_{13})$, different values of α are used in equation (5.48) and the related suppression levels are plotted in Figure 6.5, where

$$|\tau^r_{0_2}| = \sqrt{(\tau^b_{0_2})^2_c + (\tau^b_{0_2})^2_s} \text{ and } |f^r_{0_1}| = \sqrt{(f^b_{0_1})^2_c + (f^b_{0_1})^2_s} \, .$$

From the numerical results, $\alpha = 10$ has been chosen, which guarantees a worst-case suppression below the maximum allowed residual unbalance.

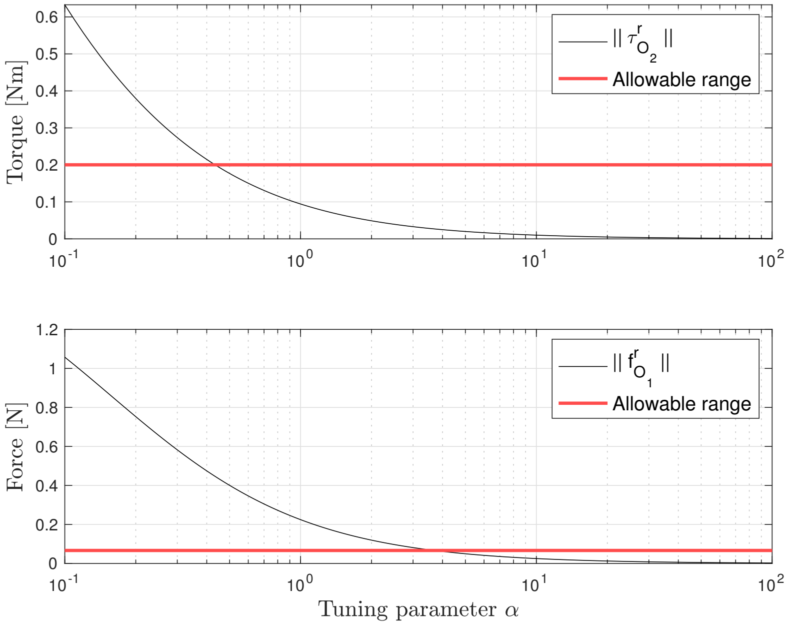


Figure 6.5: *Unbalances suppression with different α.*

Then, a Monte Carlo study has been carried out to assess the robustness of the tuned HC algorithm with respect to uncertainty on the balancing masses (±0.1 kg) and on their locations (±0.05 m). More precisely, 10000 samples have been generated and, for each of them, the upper bound on the spectral radius has been evaluated using (5.55): the maximum bound on $\rho_s(\hat{K}\Delta T)$ found in the tests is 0.0579 (which is much smaller than 1), showing that a high level of robustness is guaranteed with the proposed tuning.

6.5.4 Numerical results

In this Section, a simulation example is reported with the aim of showing the performance of the proposed ABS combined with HC. For this purpose, a multibody model of the breadboard which includes two unbalancing masses ($m_1^s = 6$kg at $(\bar{x}_1^s, \bar{z}_1^s) = (0.75, 1.46)$m and $m_2^s = 4$kg at $(\bar{x}_2^s, \bar{z}_2^s) = (-0.75, 0.02)$m), has been developed in Simulink. The control law has been implemented in discrete-time, with an update time of 200 s, which is enough to reach steady-state conditions. Extraction of the harmonics to be used in the T-matrix algorithm is performed through a real-time Fast Fourier Transform (FFT) algorithm that processes the measured signals.

The performance of the control law has been analyzed in two different conditions. In the first one the T-matrix is assumed to be exactly known. The results obtained for this (ideal) case in the discrete-time domain are collected in Table 6.1. In this table and in the following ones, ”Iteration” corresponds to the discrete-time index; the ”Force and Torque amplitude” are computed through the real time FFT algorithm with the balancing masses in the positions reported in the row ”Actual”; the rows corresponding to ”Computed” report the positions of the masses obtained with the HC at the end of the update-time and to be implemented at the next iteration. On the other hand, the results in continuous-time domain are plotted in Figure 6.6 and Figure 6.7.

Table 6.1: *Ideal case results.*

Iteration		**0**	**1-2-3-4**
Force amplitude	$[N]$	-1.008	-0.003
Torque amplitude	$[Nm]$	-4.340	-0.007
Actual s_1	$[m]$	0	-0.3251
Actual s_2	$[m]$	0	-0.3583
Actual s_3	$[m]$	0	+0.3583
Computed s_1	$[m]$	-0.3251	-0.3251
Computed s_2	$[m]$	-0.3583	-0.3583
Computed s_3	$[m]$	+0.3583	+0.3583

We can see that a satisfactory vibration suppression is achieved after one update of the control law: the interface force and torque are well below

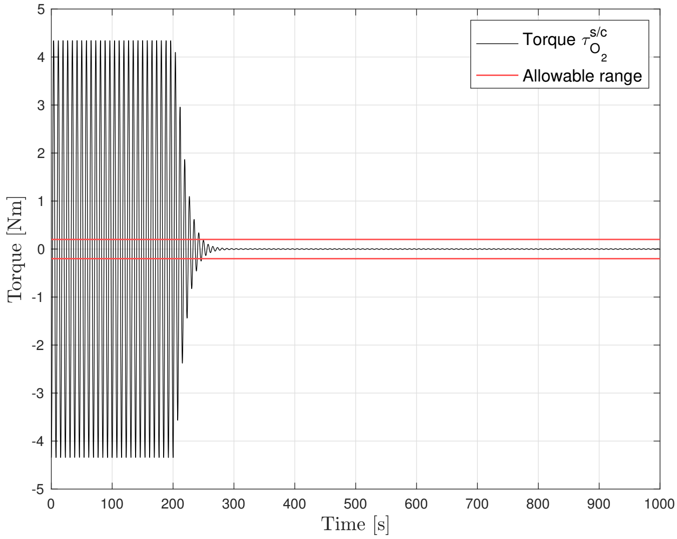


Figure 6.6: *Torque suppression - Nominal case.*

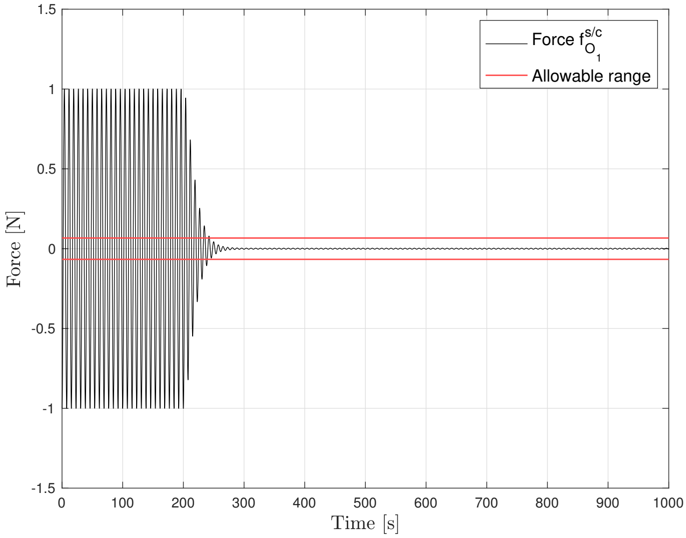


Figure 6.7: *Force suppression - Nominal case.*

the given thresholds, with the same values for all iterations after the first update (as expected from equation (5.46)).

In the second case, the ABS performance is evaluated in a more realistic scenario in which the system is affected by parametric uncertainties and imperfect knowledge of exogenous disturbances, which must be removed from the measured signal before applying the HC algorithm (Remark 33). Besides the already mentioned torque associated with gravity (6.51), a sinusoidal force of the form $f^b_{e_1} = A_e \sin(\omega_e t + \phi_e)$, with $A_e = 0.21\,\mathrm{N}$, $\phi_e = 1.93\,\mathrm{rad}$, has been removed from the measured interface force to replicate the disturbance identified on the real platform when rotating at very low speed, a condition in which the force and torque associated with inertial unbalances are very small.[2] A Monte Carlo study (500 simulations) has been carried out with respect to: uncertainty on the unbalancing masses ($\pm 0.2\,\mathrm{kg}$) and on their locations ($\pm 0.1\,\mathrm{m}$);[3] uncertainty on the amplitude ($\pm 0.05\,\mathrm{N}$) and phase ($\pm 0.1\,\mathrm{rad}$) of the compensated disturbance force; uncertainty on the ABS components (balancing masses and locations) using the same values considered in the previous Section. The most relevant statistics (mean, standard deviation, maximum and minimum values) of the absolute value of the interface loads at the last iteration of the HC algorithm are collected in Table 6.2, while Figure 6.8 and Figure 6.9 depict the corresponding time-domain representations. We can see that a satisfactory suppression is achieved in all the tests, despite the uncertainties.

Table 6.2: *Statistical properties Monte Carlo simulations.*

Statistical parameter	**Force** $[N]$	**Torque** $[Nm]$
Mean	0.003	0.007
Standard Deviation	0.007	0.006
Minimum	0.001	0.001
Maximum	0.029	0.022

[2] While the root-cause of such a disturbance is probably a combination of several factors, *e.g.*, friction, a small inclination of the structure with respect to the gravity direction, it was found to be almost invariant with respect to the rotating speed but dependent on the specific unbalancing. Hence, in the experimental tests, a model of the disturbance has been identified at low speed at the beginning of each test and then used to compensate the disturbance in the signal measured at the nominal operating speed.

[3] The values of the unbalancing masses are used only for the computation of the gravity torque (6.51) and are not needed for the on-orbit case (see Remark 33).

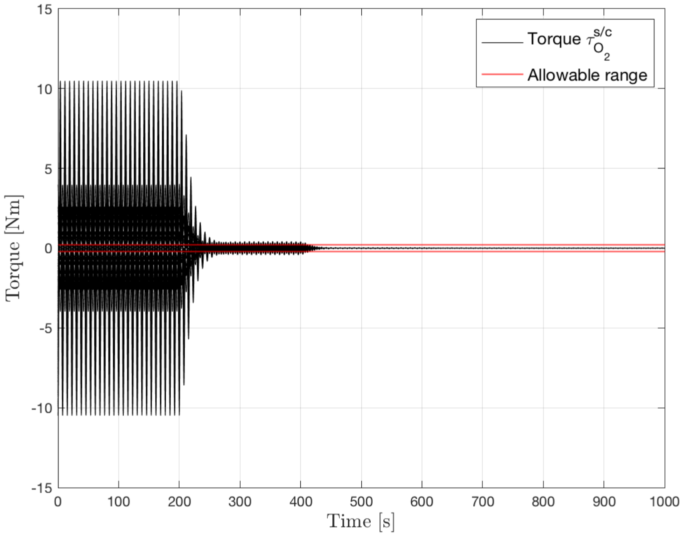


Figure 6.8: *Torque suppression - Monte Carlo analysis.*

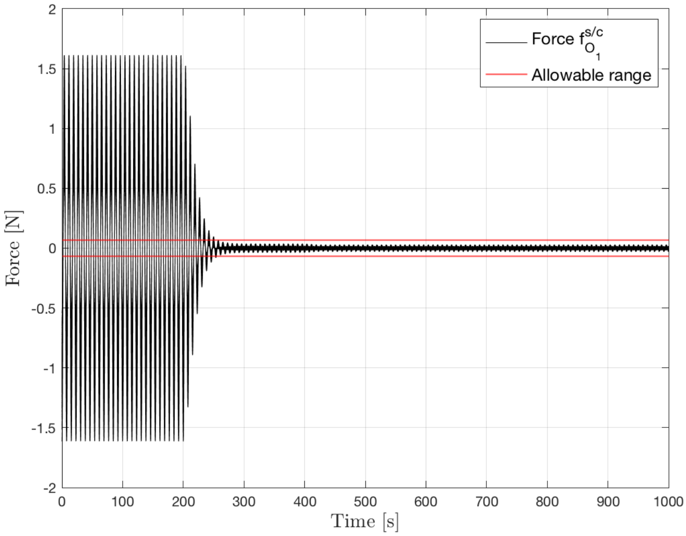


Figure 6.9: *Force suppression - Monte Carlo analysis.*

6.5.5 Experimental results

In this Section, we present the results obtained by applying the HC algorithm on the breadboard shown in Figure 6.10.

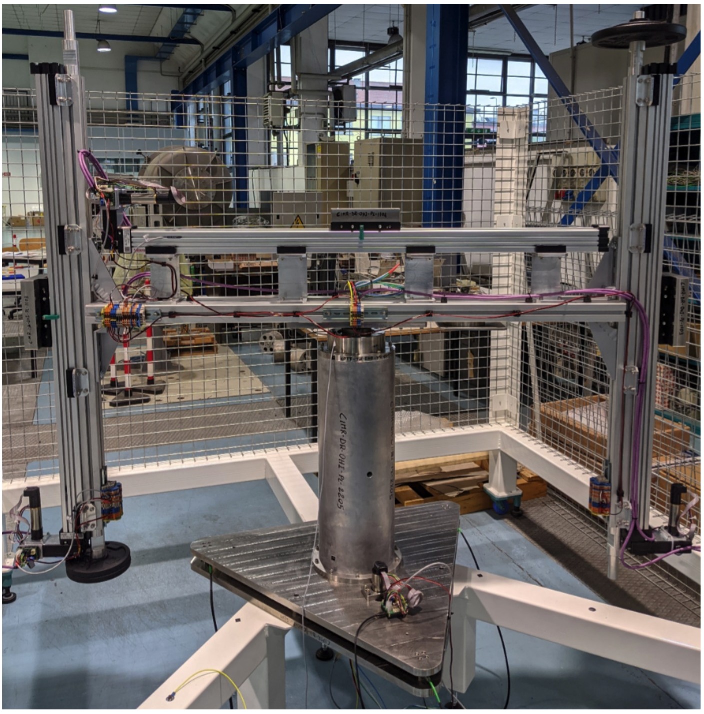


Figure 6.10: *Breadboard used in the experiments.*

In addition to the breadboard, the experimental setup includes:

- a charge Amplifier (*Kistler Type 5080A*) which receives the signals from the load cells and provides as output suitably amplified analog measurements of the interface force and torque;
- a data acquisition board (*National Instruments USB-6003*) which takes as input the amplified signals;
- a laptop computer which receives the output of the acquisition board, runs the HC algorithm and sends the commands to the linear actuators; it is also used to set the speed of the rotor.

A schematic view of the overall setup is shown in Figure 6.11.

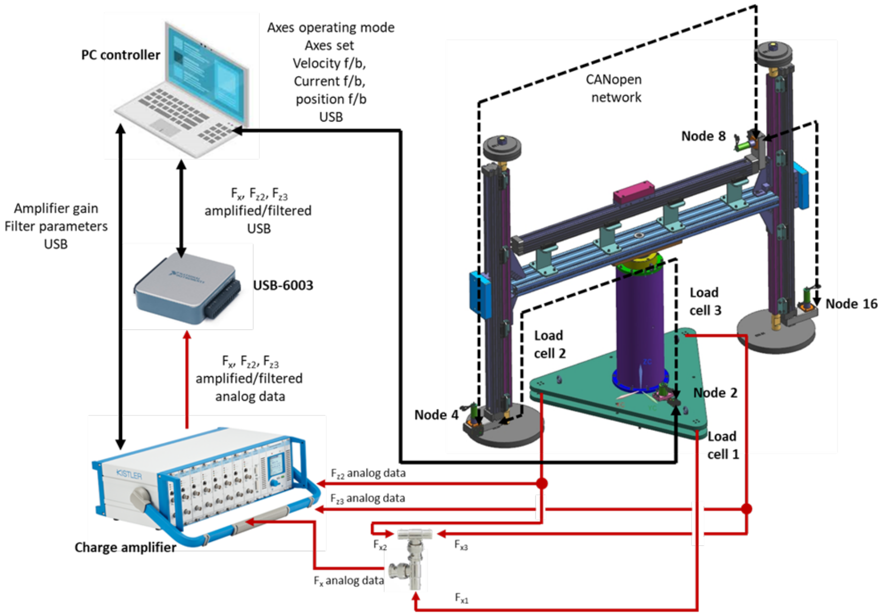


Figure 6.11: *Test setup scheme.*

The balancing tests reported in the following have been carried out at the nominal rotor speed ($\Omega = 7.8\,\text{rpm}$) through the following steps:[4]

1. a set of unbalance masses are mounted on the rotor;
2. data are logged for 300 s using the USB-6003 acquisition board;
3. the discrete FFT is applied to the logged signal to evaluate the interface loads amplitude at the rotating frequency of the rotor;
4. the target positions of the balancing masses are computed through the HC algorithm;
5. an operator checks the algorithm outputs. If the positions are feasible, the operator sends them to the actuators;

[4] As mentioned before, at the beginning of each test a preliminary test at low speed ($0.5RPM$) is performed with the aim of identifying possible disturbances, not related to inertial unbalances, to be removed from the measured signals before applying the HC algorithm.

6. the actuators receive the commands and we wait until the target positions are reached;
7. the steps 2-3 are repeated to check that the system is balanced. If not, the steps 4-6 are repeated until the system is balanced.

Remark 34. *The choice of implementing the HC algorithm with a human-in-the-loop approach is motivated by the expected on-orbit procedures in which a ground operator will have to assess, for safety reasons, the correctness of the computed positions of the masses, due to the criticality of balancing operations. It is worth mentioning that a limited amount of iterations is expected.*

In the following, three experiments with different unbalancing masses and with increasing complexity are presented:

- a *static unbalance compensation* test, in which only the central balancing mass is used;
- a *dynamic unbalance compensation* test, in which only the vertical balancing masses are used;
- *a combined static and dynamic unbalances compensation* test in which all the balancing masses are used.

Static unbalance compensation

The first balancing test involves $m_1^s = 1.051$kg at $(\bar{x}_1^s, \bar{z}_1^s) = (0.75, 1.46)$m and $m_2^s = 1.046$kg at $(\bar{x}_2^s, \bar{z}_2^s) = (0.75, 0.02)$m. To compensate for the resulting static unbalance ($\Delta S_1 = 1.57\,$kgm), the ideal position of the balancing mass m_1 should be

$$s_1 = -\frac{\Delta S_1}{m_1} = -0.34\text{m}. \tag{6.57}$$

The data related to this test are reported in Table 6.3. By inspecting the table, the slight difference in terms of the computed position (last row of Table 6.3, second iteration) with respect to the ideal position of the balancing mass (equation (6.57)) is likely due to the intrinsic unbalance of the structure and to a slightly erroneous compensation of the disturbance

$f_{e_1}^b$. Thanks to the second iteration of the HC algorithm, the measured interface load $|f_{O_1}^r|$ is reduced to 0.037 N, which is below the required threshold (0.07N).

Table 6.3: *Static unbalance compensation test.*

Iteration	**0**	**1**	**2**
Force amplitude $[N]$	-1.035	-0.071	-0.037
Static unbalance $[kgm]$	1.551	0.106	0.056
Actual s_1 $[m]$	0	-0.3101	-0.3130
Computed s_1 $[m]$	-0.3101	-0.3130	-0.3169

Dynamic unbalance compensation

The second balancing test involves $m_1^s = 5.047$kg at $(\bar{x}_1^s, \bar{z}_1^s) = (0.75, 1.46)$m and $m_2^s = 5.170$kg at $(\bar{x}_2^s, \bar{z}_2^s) = (-0.75, 0.02)$m. To compensate for the resulting dynamic unbalance ($\Delta J_{13} = -5.45\,\text{kgm}^2$), the ideal position of the balancing masses m_2 and m_3 should be

$$s_2 = -s_3 = \frac{\Delta J_{13}}{m_2 \bar{x}_2} = -0.38\text{m}. \tag{6.58}$$

As done for the first experiment, the data related to this test are reported in Table 6.4. In this case only one iteration of the HC algorithm is required. Indeed, the ideal and commanded positions of the balancing masses (equation (6.58)) are almost coincident after only one iteration and the measured interface load $|\tau_{O_2}^r|$ is reduced to 0.152 Nm, which is below the required threshold $(0.2Nm)$.

Table 6.4: *Dynamic unbalance compensation test.*

Iteration	**0**	**1**
Torque amplitude $[Nm]$	-3.657	-0.152
Dynamic unbalance $[kgm^2]$	-5.482	-0.228
Actual s_2 $[m]$	0	-0.3810
Actual s_3 $[m]$	0	+0.3810
Computed s_2 $[m]$	-0.3810	-0.3943
Computed s_3 $[m]$	+0.3810	+0.3943

Combined unbalance compensation

The last balancing test involves $m_1^s = 6.193$kg at $(\bar{x}_1^s, \bar{z}_1^s) = (0.75, 1.46)$m and $m_2^s = 4.185$kg at $(\bar{x}_2^s, \bar{z}_2^s) = (-0.75, 0.02)$m. To compensate for the resulting unbalances ($\Delta S_1 = 1.51\,$kgm and $\Delta J_{13} = -6.7\,$kgm^2), the nominal position of the balancing masses m_1, m_2 and m_3 should be

$$s_1 = -\frac{\Delta S_1}{m_1} = -0.33\text{m}, \; s_2 = -s_3 = \frac{\Delta J_{13} - m_1 \bar{z}_1 s_1}{m_2 \bar{x}_2} = -0.37\text{m}. \quad (6.59)$$

The data related to this test are reported in Table 6.5. Also in this case only one iteration of the HC algorithm is required to reduce the measured interface loads below the required thresholds and the values of the commanded positions are close to the expected ones.

Table 6.5: *Combined unbalance compensation test.*

Iteration		**0**	**1**
Force amplitude	$[N]$	-1.086	-0.001
Torque amplitude	$[Nm]$	-4.282	-0.029
Static unbalance	$[kgm]$	1.628	0.002
Dynamic unbalance	$[kgm^2]$	-6.419	-0.0435
Actual s_1	$[m]$	0	-0.3328
Actual s_2	$[m]$	0	-0.3539
Actual s_3	$[m]$	0	0.3539
Computed s_1	$[m]$	-0.3328	-0.3329
Computed s_2	$[m]$	-0.3539	-0.3540
Computed s_3	$[m]$	0.3539	0.3540

6.5.6 Final considerations

In this Section, the effectiveness of the proposed design has been evaluated both in simulation as well as through experiments on a dedicated breadboard. The results showed that the system is capable of reducing the force and torque induced by the unbalance at the interface between the fixed and the rotating part within predefined bounds, even in the presence of imperfect knowledge of the system parameters.

CHAPTER 7

Integrated active and passive rotorcraft vibration control

Most helicopters experience significant levels of vibration during flight. Vibrations directly affect pilot, crew, and passenger comfort and have a strong influence on the fatigue life of mechanical, structural, and electronic components, which has implications on maintenance and operating costs. The main rotor represents the principal source of vibrations. The effect on the fuselage of the generated vibratory loads is essentially periodic, being the rotor angular rate essentially constant. Indeed, it has been established that the resulting angular frequency is equal to $N\Omega$ (classically denoted as N/rev), where N is the number of blades and Ω is the rotor angular frequency, also denoted as 1/rev, meaning “per revolution”. Accordingly, the vibration suppression problem can be formulated as the compensation of a periodic disturbance of a known frequency $N\Omega$ acting at the output of a linear system. Several control strategies have been developed to solve this problem, which can be divided into passive and active vibration control techniques. Despite the large amount of research produced in the last decades, the interaction between active and passive vibration control techniques has not been analysed yet. The main contribution of this Chapter is the analysis of the interaction between the Active Control of Structural Response and a Mast Vibration Absorber (MVA). Specifically, we show that an *ad hoc* design of the control algorithm that accounts for the MVA behaviour is needed to avoid instabilities and/or degraded performance caused by the interaction between the two devices.

In this Chapter, we present a brief overview of the general vibration control problem in helicopter applications, highlighting similarities and differences among the various approaches developed over the years. Then, we model and analyse the coupling term between the active control and the passive device and we design a HC algorithm explicitly accounting for this term. In the last part of the Chapter, numerical results obtained with a virtual helicopter are discussed, comparing the proposed architecture with the classical HC algorithm.

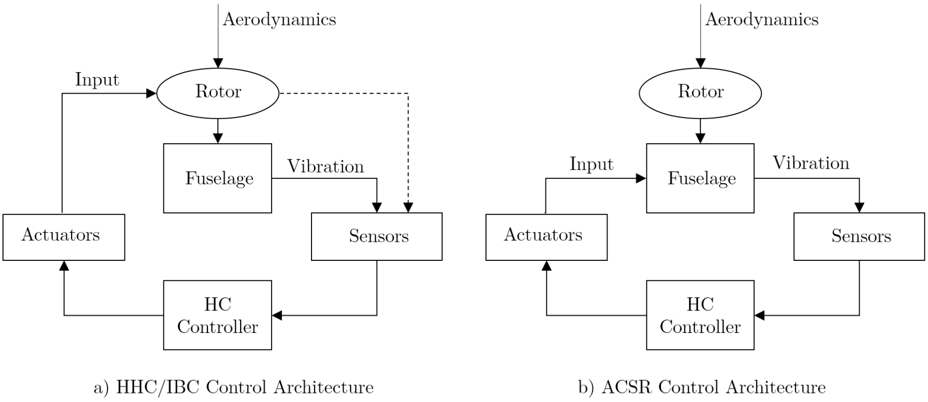
7.1 Background

In addition to their consolidated use in military applications, helicopters have gradually expanded their influence in civilian society since the *1960s*. They are used in various extreme situations, such as rescue operations or emergency air medical assistance, due to their unique characteristics and manoeuvring capabilities. However, urban communities have often expressed a strong aversion to noisy aircraft, making police and passenger helicopters still unpopular. The main aspects degrading passenger and crew comfort are the high noise and vibration levels. In addition, vibration loads constitute a significant factor in the decline of structural integrity. Therefore, helicopters should be designed to achieve the lowest possible vibrational levels, which in turn leads to improved acceptance for the commercial market. The vibratory loads transmitted to the fuselage originate in periodic airloads on the rotor blades, as a consequence of the asymmetric airflow generated by the lifting rotor in forward flight [167]. The vibratory blade airloads contain all harmonics of the rotor frequency, with a trend to lower magnitudes at higher harmonics. Many of the harmonic components have little effect on vibration because the root loads they produce on the various blades cancel out with each others. Indeed, the net vibratory hub and the resulting fuselage vibration consist primarily of blade-number multiple harmonics denoted as N/rev. From a control standpoint, several methods have been developed to minimize vibration in helicopter cabins. In particular, vibration control methods can be categorized into passive and active techniques.

Passive vibration control techniques Among the passive control techniques, one technique involves designing the blades and fuselages to separate their natural vibration frequencies from the rotor harmonic frequencies. This tuning aids in preventing resonances between the rotor harmonics and the blade and fuselage. However, as demonstrated by Nguyen *et al.* in [168], this approach necessitates extensive modelling and testing of the blade and fuselage structural characteristics throughout the aircraft development phase. Another commonly used method of vibration reduction is rotor isolation. Isolating materials, *e.g.*, rubber pads or springs, are placed between the vibrating system and its supporting structure to minimize the force transmitted from the vibrating system to the structure. Mechanical components, such as tuned-mass absorbers and tuned-mass dampers, are used in other passive methods. The former can be attached to a structure to reduce its dynamic response. The latter works similarly, but includes an additional mechanical damper. Absorbers and dampers are installed at specific locations in the fuselage and are tuned to the frequencies to be removed. Despite their simplicity, the vibration attenuation achieved by these methods is limited in the amount of energy absorption and frequency operation range. Furthermore, they involve an additional weight reducing the available payload of the helicopter [169]. Additionally, tuned-mass absorbers and dampers cannot adjust to changes in operating conditions because of their fixed structure [168].

Active vibration control techniques In order to maximize noise and vibration reduction and get beyond the limitations of the passive control techniques, active vibration reduction methods were developed. Typically, acoustic noise signals or vibrations are measured by sensors mounted on the fuselage. After that, a controller processes the measurements and uses them to produce a signal that drives the actuators, generating a vibration field as close as possible to the one that has to be eliminated but opposite in phase, leading to destructive interference. Based on the location of the actuators, existing active techniques can be classified as (see Figure 7.1):

- *Rotor location*: Individual Blade Control (IBC) and Higher Harmonic Control System (HHC);
- *Fuselage location*: Active Control of Structural Response (ACSR).



a) HHC/IBC Control Architecture

b) ACSR Control Architecture

Figure 7.1: *HC architectures classified according to the location of actuators.*

Actuators mounted on the rotor are used in the HHC [170, 171] and IBC [172, 173] techniques to attenuate vibrations. Consequently, vibrations are attenuated before propagating to the fuselage. The ACSR method, on the other hand, takes into account vibrations transmitted to the cabin [174, 151]. Nonetheless, both the approach groups rely on the HC concept (see, *e.g.*, [151] and [150]) discussed in Chapter 7. The convergence of the HHC algorithm was analysed in [150]. In particular, Patt *et al.* studied the robustness of the HHC algorithm to uncertainty in the T-matrix and introduced a relaxation coefficient into the update law to increase the robustness of the algorithm. However, their study only accounted for the steady-state representation. In [148] a continuous-time analysis of the algorithm was presented, where it was shown that the controller could be approximated using a notch filter with complex poles at the disturbance frequency. On the other hand, in [175] the entire control loop was analysed in discrete-time and in [152] the continuous-time approach was compared to the discrete one showing that neglecting the discrete components of the HHC is unconservative.

The T-matrix is usually estimated from data as it is hardly computed from first-principle models. Therefore, researchers have investigated both adaptive and robust control approaches. Johnson proposed an adaptive control approach based on Recursive Least Squares (RLS) methods to estimate the T-matrix online in [176]. In [177], on the other hand, Lovera *et al.*, after formulating the control problem in a robust control framework, used H_∞ synthesis to design fully parametrized gain matrices.

7.2 Model formulation

In this Section, we analyse the interaction between the ACSR and the Mast Vibration Absorber (MVA). First, we present the working principles of the MVA and the ACSR. Then, we model and analyse the coupling term between the active and passive device.

7.2.1 Mast Vibration Absorber

The MVA consists of a mass mounted on the rotor head of the helicopter via an elastic beam (see Figure 7.2). A detailed description of a simplified model of the MVA can be found in [179, Chapter 13.3]. The main function of the MVA, which rotates with the rotor head, is to create antiresonance in correspondence of its natural frequency, which is tuned to the same frequency of the disturbance that needs to be attenuated in the fixed frame, *i.e., N*/rev, thus, reducing the in-plane vibrations transmitted to the airframe. The relation between the MVA and the helicopter can be seen as a feedback interconnection: the passive device takes as input the rotor accelerations (a_{MR}) and returns as output the loads (f_{MVA}) that oppose to the excitation ones.

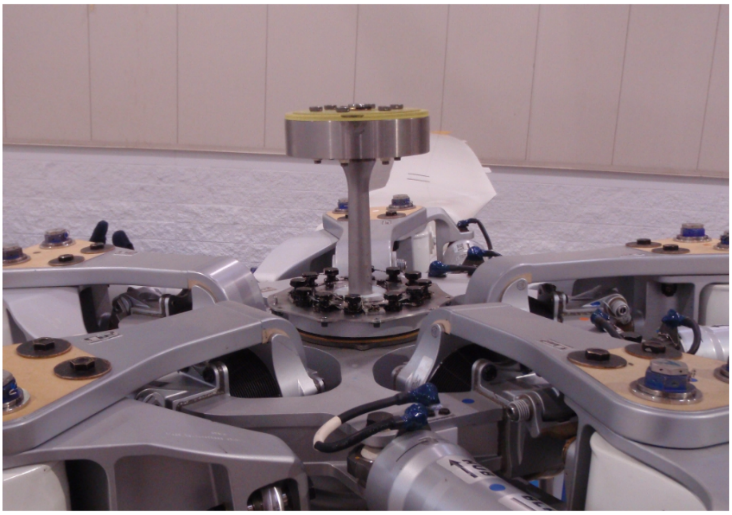


Figure 7.2: *AW139 MVA assembly [178].*

7.2.2 Active Control of Structural Response

ACSR is one of the most popular architectures for helicopter vibration reduction. Sensors are located at critical points in the fuselage, where it is necessary/desired to minimize vibration, *e.g.*, the pilot and passenger seats. The ACSR performance was examined in [180] using a coupled rotor and flexible fuselage, which revealed that the ACSR has a low power need. Furthermore, the ACSR could have fewer airworthiness problems than the HHC, because no modification to the rotor is required [149].

The ACSR is based on the superposition of the primary uncontrolled vibration response and the controlled secondary vibration response, controlled in such a way that the vibration is minimised throughout the airframe. Both the actuators and the sensors are placed on the fuselage. Since the objective of the control is to attenuate vibration at the N/rev, the helicopter is modelled at steady-state, considering only the frequency of interest. Let $y \in \mathbb{R}^p$ be the vector of measured outputs and $u \in \mathbb{R}^m$ be the vector of control inputs and define the vectors Y_N and U_N that contain the N/rev cosine and sine harmonics of the measured output and control input respectively; then, HC is a viable approach for the design of a feedback controller. In this work, we apply the control law described in Section 5.5, namely:

$$U_N(k+1) = K_M U_N(k) + K_N Y_N(k) \tag{7.1}$$

where the gain matrices are defined as:

$$K_N = -(T_u^\top Q T_u + R)^{-1} T_u^\top Q \;\text{ and }\; K_M = -T_u K_N \;, \tag{7.2}$$

where the T_u-matrix takes into account the usual steady-state response of the measured outputs to the applied harmonic controls. In the following, we consider the weighting matrices as $Q = I$ and $R = wI$, where w represents the inverse of the compensator gain, *i.e.*, the larger w is, the lower the control action.

7.2.3 Coupled model

For simulation and analysis purposes, we take into account that, as far as the N/rev vibratory response is concerned, the entry point of disturbances is the main rotor hub and that the actual disturbance vector has dimensions which are smaller than the ones of the vector of measured outputs[1], *i.e.*,

$$Y_N(k) = T_u U_N(k) + T_d D_N \;, \tag{7.3}$$

where the T_d-matrix represents the harmonic response of the measured outputs to the harmonic disturbances. This modelling approach allows taking into account very easily the role of the MVA and its interaction with the rotorcraft model and its dynamics. Indeed, the mechanical interface between the helicopter and the MVA is precisely at the level of the forces and accelerations at the main rotor hub, so that the complete model of the aircraft with MVA can be represented as in the block diagram of Figure 7.3. The main rotor hub accelerations a_{MR} have not been included as outputs of the state-space models to reflect that they are not measured and, as such, cannot be used in the implementation of the control system. It is interesting to underline that the interconnection between the helicopter

[1] From a simulation and an analysis viewpoint treating the disturbance components as independent would lead to an unrealistically challenging representation of the real situation as clearly the vibratory responses sensed at the individual measurement locations cannot be assumed to be independent of each other.

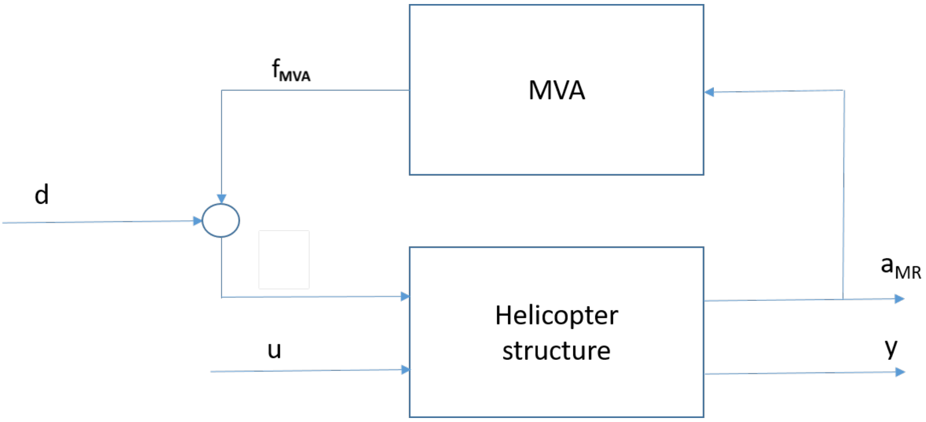


Figure 7.3: *Block diagram of input-matched disturbance model.*

structure and the MVA depicted in Figure 7.3 is a particular case of the so-called Linear Fractional Representation (LFR) widely used in the robust control engineering literature, which provides an interesting perspective in the understanding of the interaction between the MVA and the active control system to be analysed. Indeed, abstracting Figure 7.3 to Figure 7.4, we can translate the block diagram in the following set of equations:

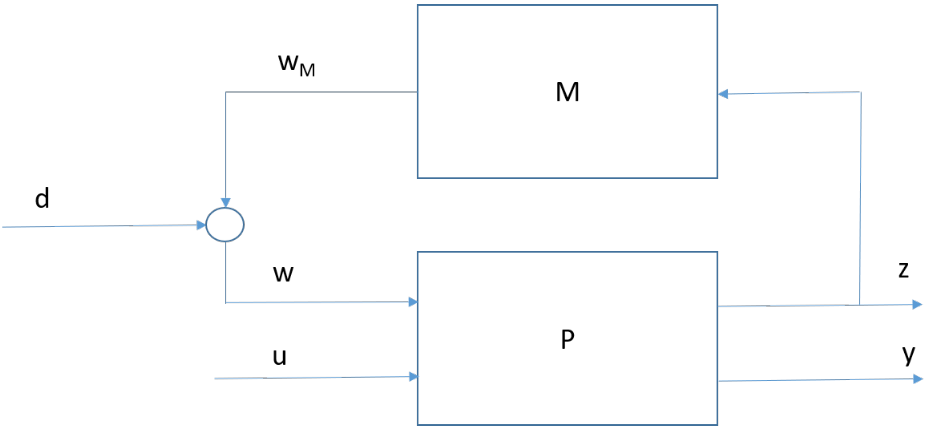


Figure 7.4: *Block diagram of input-matched disturbance model.*

$$\begin{bmatrix} z \\ y \end{bmatrix} = P \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{7.4}$$

$$w = d + w_M, \quad w_M = Mz. \tag{7.5}$$

By manipulating the above equations one can compute the explicit relations between the two input-output pairs as follows:

$$y = P_{21}d + P_{21}Mz + P_{22}u \tag{7.6}$$

$$z = (I - P_{11}M)^{-1}P_{11}d + (I - P_{11}M)^{-1}P_{12}u \tag{7.7}$$

and substituting the expression for z into the one for y one gets

$$\begin{aligned} y = P_{21}\left[I + M(I - P_{11}M)^{-1}P_{11}\right]d+ & \quad (7.8) \\ + \left[P_{22} + P_{21}M(I - P_{11}M)^{-1}P_{12}\right]u & \\ z = (I - P_{11}M)^{-1}P_{11}d + (I - P_{11}M)^{-1}P_{12}u\ . & \quad (7.9) \end{aligned}$$

Note that the above expressions can be used, *mutatis mutandis*, to model both the dynamic response of the system (interpreting P and M as transfer functions and y, z, u and w as Laplace transforms of the corresponding variables) and the corresponding T-matrices (interpreting in this case P and M as the T-matrices of the corresponding systems and y, z, u and w as the N/rev harmonics associated with the corresponding variables). Indeed considering

$$T_u = P_{22} + P_{21}M(I - P_{11}M)^{-1}P_{12} \tag{7.10}$$

$$T_d = P_{21}\left[I + M(I - P_{11}M)^{-1}P_{11}\right] \tag{7.11}$$

one can therefore obtain explicit expressions for the T_u and T_d matrices appearing in equation (7.3) as a function of the blocks of the helicopter model P and of the MVA model M.

Remark 35. *From the expression of T_u one can readily see that the presence of the MVA modifies the T-matrix of the active control system in a significant way, by introducing the term $P_{21}M(I - P_{11}M)^{-1}P_{12}$ (which is zero when the MVA is absent) which sums with the term P_{22} which corresponds to the no-MVA case. In dynamical terms, on the other hand,* *equation 7.10 clearly shows that the transfer function from the control variable u to the measurement vector y contains additional modes arising from the feedback interconnection of the helicopter structure with the MVA. Similarly, the equation* (7.9) *defining z, i.e., the acceleration at the main rotor hub, which is the forcing input for the dynamics of the MVA, shows a forcing term due to the control action u, through the function* $(I - P_{11}M)^{-1}P_{12}$. *This term accounts for the interactions between the ACSR and the MVA: indeed when the control action is non-zero it will cause a non-zero acceleration at the main rotor hub which in turn excites the MVA dynamics, leading to potentially harmful interactions.*

Remark 36. *Since the HC algorithm relies on the steady-state model* (7.3)*, the time interval Δt between the updates of the control action must be long enough to allow the transients to decay. Therefore, if the control update is faster than the dynamics of the MVA, the coupling between the two devices can destabilize the closed-loop system. To prevent this possibly harmful coupling one can think to increase the update time of the digital implementation of the HC algorithm or to increase the penalization on the control term in the LQ-like cost function. However, both solutions result in low performance: in the first case the closed-loop system shows slow response to changes in the harmonic disturbance, while in the second case low vibration attenuation is achieved.*

While these considerations are extremely useful in terms of *time-domain* analysis, as they allow to predict the onset of such interactions as a function of the controller tuning, their usefulness for control design purposes is limited as they are based on the possibility of “unravelling” the LFR associated with the dynamics and with the T-matrix starting from measured data. In other words, taking advantage of the special structure of the T-matrix in control law design would be possible only if separate measurements of the T-matrices with and without MVA were available. In this work, assuming these measurements available, we add decoupling between the MVA and the ACSR as a design requirement for the active control system.

7.3 Control design

In this Section, a novel HC law is developed to reduce the coupling between the HC action and the MVA. First, we show how knowledge of the T-matrices with (T_u) and without MVA ($\bar{T}_u$) can be leveraged to identify the subspace in which the control action does not excite the MVA. Then, we design a control action constrained to lie in this subspace, achieving a strong decoupling between the control action and the passive device.

7.3.1 Decoupling projection operator

In view of equations (7.4)-(7.5), we have:

$$T_u = P_{21}M(I-P_{11}M)^{-1}P_{12}+P_{22} \ , \tag{7.12}$$

$$\bar{T}_u = P_{22} \ . \tag{7.13}$$

Subtracting (7.13) from (7.12) leads to:

$$T_u - \bar{T}_u = P_{21}M(I-P_{11}M)^{-1}P_{12}. \tag{7.14}$$

Considering the singular value decomposition of the left-hand side

$$U\Sigma V^T = T_u - \bar{T}_u \ , \tag{7.15}$$

we define the decoupling projection operator as:

$$\Pi \ = I - V(:,1:r)V(:,1:r)^\top = I \ - V_rV_r^\top, \tag{7.16}$$

where V_r collects the first r singular vectors of V and r is the rank of Σ. Equation (7.16) defines the orthogonal projection onto the null space of P_{12} ($P_{12}\Pi = 0$). Specifically, it identifies the subspace in which the control does not excite the MVA; on the other hand $I-\Pi$ identifies the directions along which the control excites the MVA and possibly destabilizes the closed-loop system.

Remark 37. *Note that by right multiplying both side of equation* (7.14) *by the projection* Π*, we obtain* $T_u\Pi = \bar{T}_u\Pi$*, i.e., the operator* Π *makes the 'dynamics' of the helicopter with and without the MVA equivalent.*

7.3.2 Decoupled HC

Once the projection operator Π is obtained, the control input U_N can be decomposed in two components denoted as $U_{\|} = (I-\Pi)U_N$ and $U_{\perp} = \Pi U_N$

$$U_N = U_{\|} + U_{\perp} = \Pi U_N + (I-\Pi)U_N \ , \tag{7.17}$$

where ΠU is the orthogonal projection of U_N onto the null space of P_{12}, *i.e.*, the part of U that does not excite the MVA, while $(I-\Pi)U_N$ is the part of U_N that excites the MVA and can drive the system unstable. By setting $U_{\|} = 0$ a strong decoupling between the control and the passive device is achieved. The resulting algorithm is denoted Decoupled-HC (D-HC).

Remark 38. *The price to pay for a control action fully decoupled from the MVA is a reduced control authority due to the reduced space in which the control is allowed to operate. More precisely, since Π has a rank of $2m-r$, the effective T_u-matrix of the closed-loop system, $T_u\Pi$, has a reduced column space (the rank is at most $2m-r$).*

Alternatively, it is possible to control the amount of coupling using an *LQ*-based approach by choosing the control weight R as

$$R = w_1\Pi^T\Pi + w_2(I-\Pi)^T(I-\Pi). \tag{7.18}$$

Therefore, the cost function to be minimized can be rewritten as

$$\begin{aligned} J(k) = Y_N^T(k)\,QY_N(k) + w_1U_N(k)^T\,\Pi^T\Pi U_N(k) + \\ + w_2U_N(k)^T\,(I-\Pi)^T(I-\Pi)U_N(k)\ . \end{aligned} \tag{7.19}$$

Then, the control matrices K_N and K_M can be computed using equation (7.2). The weights w_1 and w_2 are the tuning parameters: the larger w_2 is, the lower the control action that excites the MVA, namely $u_{\|}$. The resulting control algorithm is denoted Partially Decoupled-HC (PD-HC). Moreover, as $w_2 \to +\infty$ the control action is fully decoupled from the MVA and we recover the D-HC algorithm.

7.3.3 Continuous time formulation of the HC algorithm

To assess the effectiveness of the proposed approach, the behaviour of the closed-loop eigenvalues associated with the MVA modes is evaluated.

This analysis aims to show that the coupling between the control action and the passive device can lead to instabilities of the closed-loop system, and that the proposed decoupling controller (D-HC) does not affect the MVA modes avoiding this dangerous behaviour. In this Section, we reformulate HC algorithm in a *time-domain* state-space model that will be used to close the loop with the virtual helicopter model described in the following Section.

Following the approach presented in [175], one obtains:[2]

$$\dot{Y}_N = AY_N + B(\Omega t)y, \tag{7.20}$$

$$u = C(\Omega t)Y_N, \tag{7.21}$$

where the system matrices are defined as:

$$A = 0_{2p\times 2p} \tag{7.22}$$

$$B = \begin{bmatrix} \cos(N\Omega t) \\ \sin(N\Omega t) \end{bmatrix} \otimes I_{p\times p} \tag{7.23}$$

$$C = -\frac{2}{\Delta t}\begin{bmatrix} \cos(N\Omega t) & \sin(N\Omega t) \end{bmatrix} \otimes I_{m\times m}K_N, \tag{7.24}$$

with $\otimes$ is the Kronecker product, Δt represents the update time of the digital implementation of the algorithm, and K_N is obtained using equation (7.2). This model has the appearance of a LTP one, however it is easy to show by means of an appropriate Lyapunov transformation (see, *e.g.*, [181]) that the demodulation and modulation carried out by the HC algorithm cancel out exactly, leaving a LTI compensator. Indeed, choosing

$$S(t) = \begin{bmatrix} -\cos(N\Omega t) & -\sin(N\Omega t) \\ \sin(N\Omega t) & -\cos(N\Omega t) \end{bmatrix} \otimes I_{p\times p}, \tag{7.25}$$

and performing the Lyapunov transformation

$$\begin{bmatrix} Y_{N1} \\ Y_{N2} \end{bmatrix} = S \begin{bmatrix} Y_{Nc} \\ Y_{Ns} \end{bmatrix}, \tag{7.26}$$

one has that the compensator can be equivalently described by the follow-

[2] For ease of notation in the following we consider a HC with a pure integral action *i.e.*, $K_M = I$.

ing model:

$$\dot{Y}_{1,2} = \left[SAS^{-1} + \dot{S}S^{-1}\right] Y_{1,2} + SBy \tag{7.27}$$

$$u = CS^{-1}Y_{1,2}. \tag{7.28}$$

By combining equations (7.27)-(7.28) with equations (7.24)-(7.25) and noting that for a LTI system K_N inherits the same well-known block structure of the T-matrix [182], that is:

$$K_N = \begin{bmatrix} K_a & K_b \\ -K_b & K_a \end{bmatrix} \tag{7.29}$$

After some manipulations one obtains:

$$\dot{Y}_{1,2} = \begin{bmatrix} 0_{p\times p} & N\Omega I_{p\times p} \\ -N\Omega I_{p\times p} & 0_{p\times p} \end{bmatrix} y_{1,2} + \begin{bmatrix} -I_{p\times p} \\ 0_{p\times p} \end{bmatrix} y \tag{7.30}$$

$$u = -\frac{2}{\Delta t}\begin{bmatrix} -K_a & K_b \end{bmatrix}. \tag{7.31}$$

The update time Δt of the digital implementation appears as the inverse of the controller gain. In the following Section, the sensitivity of the closed-loop eigenvalues is studied with respect to this parameter.

7.4 Simulation study

Numerical simulations are performed to evaluate the proposed algorithm in terms of stability, steady-state performance and transient behaviour. The helicopter model used in this study is built based on data representative of a generic, medium weight helicopter with a conventional articulated main rotor and tail rotor configuration. The model has been realized using Modern Aeroservoelastic State Space Tools (MASST), a MATLAB tool developed at Politecnico di Milano for the aeromechanical and aeroservoelastic analysis of fixed and rotary wing aircraft [183]. The model is built from subcomponents that are assembled in an overall model using the Craig Bamptons Component Mode Synthesis approach. The airframe elastic model was generated in NASTRAN while both the main rotor and the tail rotor aeroelastic models are obtained in CAMRAD/JA. For further details the reader is referred to [184]. The helicopter model is formulated as a linear system in state-space form, given by

$$\dot{x} = A_h x + B_h u \tag{7.32}$$

$$y = C_h x + D_h u, \tag{7.33}$$

where $y \in \mathbb{R}^{10}$ is the vector containing the accelerations measured by the sensors placed on the critical points of the airframe; the vector $u \in \mathbb{R}^7$ accounts for the three external forces and for the four actuator forces acting on the gearbox struts. The setup considered in the case study is described in Figure 7.5 and Figure 7.6, respectively as far as the sensor and actuator location is concerned. As can be seen from the figures, the considered setup mimics the typical arrangement of actuators and sensors of the ACSR system on the AW101.

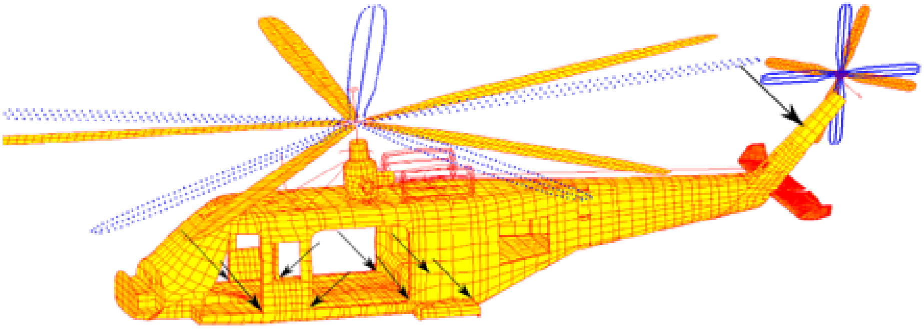


Figure 7.5: *Case study sensors location.*

7.4.1 Stability analysis results

In this Section, we show the effect of the HC algorithms on the eigenvalues associated with the MVA modes as a function of the update time Δt. The closed-loop system is obtained by combining the virtual helicopter model (7.32)-(7.33) and the continuous-time controller model

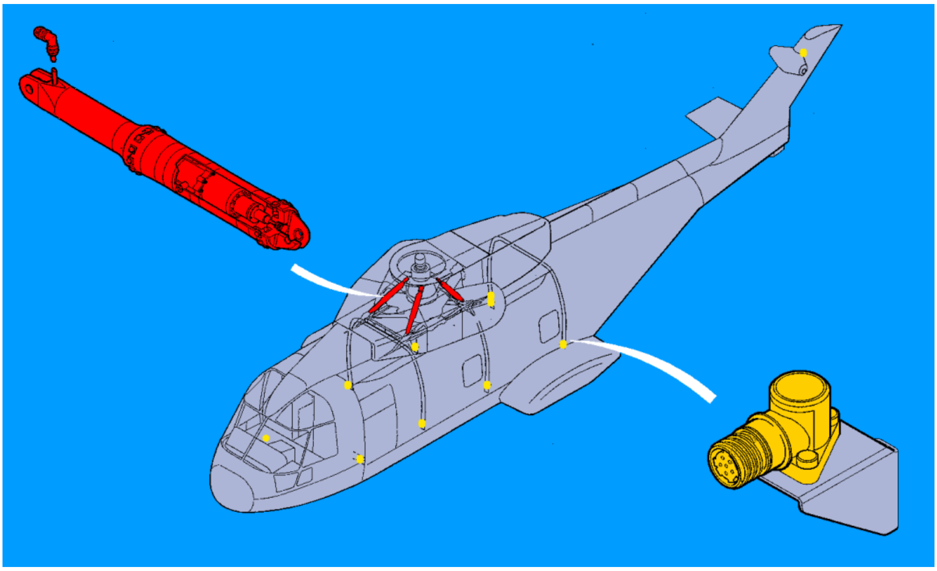


Figure 7.6: *Case study actuators location.*

(7.30)-(7.31). We compare the results obtained with the standard HC and the proposed D-HC using as weight on the control action $R = wI$, where $w = 10^{-2}$. Figure 7.7 shows the behaviour of the closed-loop MVA eigenvalues with the standard HC algorithm as Δt goes from $5\,\mathrm{s}$ to $0.1\,\mathrm{s}$, while Figure 7.8 shows the behaviour of the same eigenvalues with the proposed approach[3].

This analysis shows that fast implementations of the standard HC algorithm make the closed-loop system unstable. In this particular case, the MVA eigenvalues cross the imaginary axis at $\Delta t = 1.4\,\mathrm{s}$, hence a shorter update time ($\Delta t < 1.4\,\mathrm{s}$) brings instability. On the other hand, our modification allows a faster implementation barely affecting the MVA modes. As mentioned in Remark 38, the cost of this operation is a loss of control authority caused by the reduced space in which the control action is allowed to operate. The PD-HC law derived minimizing (7.19) provides a way to tune the amount of the interaction and, therefore, to find a trade-off between steady-state performance and transient behaviour.

[3] For ease of representation only the eigenvalues with positive imaginary part are shown.

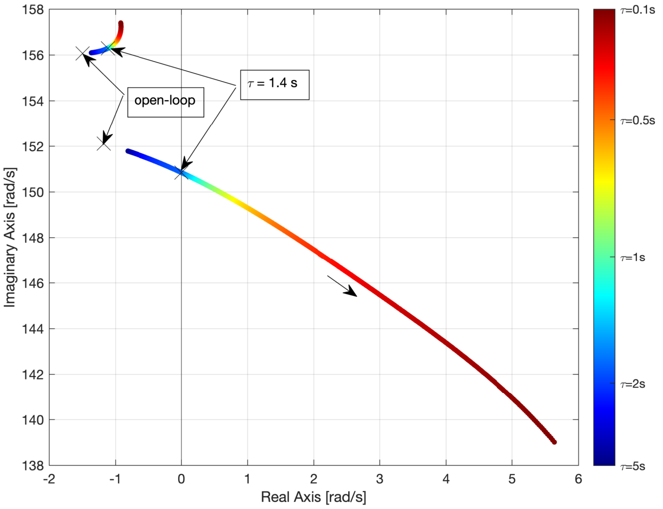


Figure 7.7: *Closed-loop MVA eigenvalues with the standard HC.*

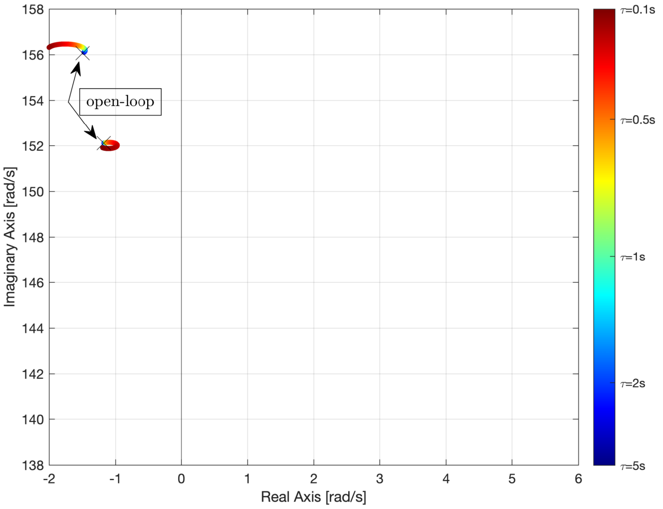


Figure 7.8: *Closed-loop MVA eigenvalues with the proposed D-HC.*

7.4.2 Steady-state performance

In this Section, the effect of tuning the control weight R on the steady-state performance is analysed in the HC and PD-HC algorithms using the steady-state model (7.3). In other words, the results obtained in this Section can be achieved using an update time much longer with respect to the settling time of the system (dominated by the MVA dynamics). Specifically, we study the sensitivity of steady-state performance to changes in w for the HC algorithm, while we assume $w_1 = 0$ and we study the sensitivity with respect to w_2 for the PD-HC algorithm. To compare easily the different configurations, the following metrics are defined[4]:

- $1 - ||Y_N||/||\bar{Y}_N||$: The vibration attenuation with respect to the baseline level $||\bar{Y}_N||$ that is evaluated at steady-state without the active control (only MVA is considered);
- $||U_N||$: The steady-state control action;
- $||U_{||}||$: The steady-state control action projected onto the subspace that excites the MVA;
- $||U_{\perp}||$: The steady-state control action projected onto the subspace that does not excite the MVA.

Figure 7.9 shows the control effort as a function of w for the standard HC algorithm. As expected, the control effort decreases as the weight w increases. Figure 7.10 shows the control effort as a function of w_2 for the PD-HC algorithm. The values reported in Figures 7.9 and 7.10 are scaled by the same reference value and consequently are comparable. Namely, all the values are scaled by the norm of the control $||U_N||$ obtained in the case $w = 0$, *i.e.*, applying the T-matrix algorithm (5.32). When $w_2 = 0$ the PD-HC algorithm behaves as the conventional HC since $w_2 = w_1 = w = 0$. As expected, $u_{||}$ decreases as w_2 is increased and it tends to zero as $w_2 \to \infty$ achieving a fully decoupled control.

Remark 39. *Note that $||U_N||$ is not a monotonic function of w_2 since minimization of the cost function* (7.19) *requires an increased demand of control in the $U_{\perp}$ direction, which is not as effective as the parallel direction $U_{||}$ in reducing the remaining vibrations.*

[4] In the following, the operator $||(\cdot)||$ will indicate the Euclidean norm of the vector $(\cdot)$.

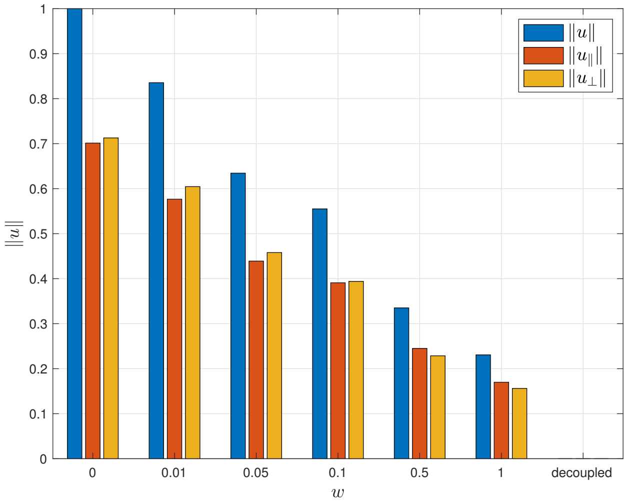


Figure 7.9: *HC algorithm: norm of the steady-state input as a function of w.*

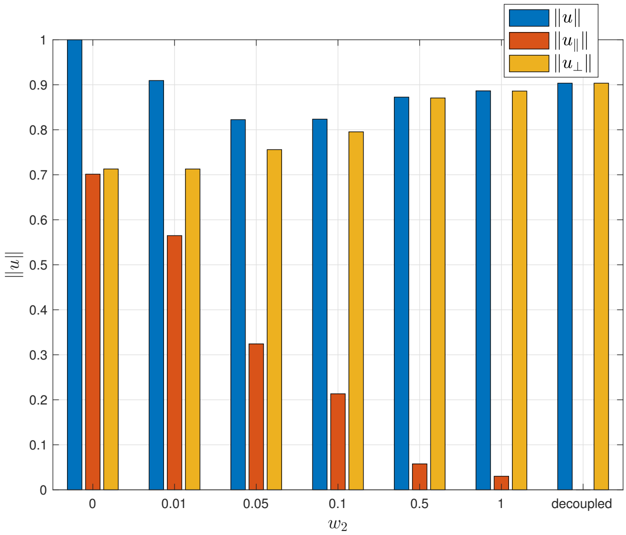


Figure 7.10: *PD-HC algorithm: norm of the steady-state input as a function of w_2.*

Figure 7.11 shows the vibration attenuation[5] achieved with the two controllers under study, obtained by varying the corresponding penalization on the control. It can be seen that by increasing the penalization on $U_{\|}$ (PD-HC case) the vibration reduction tends to the value achieved considering the fully decoupled controller (D-HC). Note that, by constraining the control, *i.e.*, $U_{\|} = 0$, we can only achieve a reduction of 70% instead of 94% achievable with the standard HC.

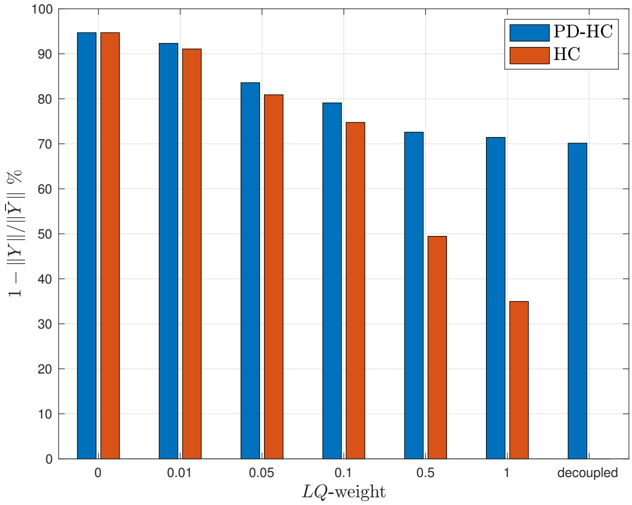


Figure 7.11: *Norm of the steady-state output as a function of the LQ-weight.*

Remark 40. *It is important to underline that the previous analysis is based on the steady-state model. As a consequence, the promising results of the standard HC algorithm requires an update time that can be too long for practical applications. In fact, rapid manoeuvres such as the transition to hover are characterised by a rapid increase in the baseline vibration during 1 to 2 seconds. If the control update time is longer than 1 second, then the increase in vibration will go unchecked until a new measurement is calculated, by which time the vibration may have further*

[5] Recall that vibration reduction levels are computed with respect to the steady-state value obtained with the MVA alone and not with respect to the bare rotorcraft (without the MVA).

increased [185]. Furthermore, if the proposed decoupling technique is not used, the coupling term $U_{\|} \neq 0$ can destabilize the overall closed-loop system during the transient as shown in Section 7.4.1.

7.4.3 Transient behaviour

Finally, we compare the transient behaviour of the standard HC algorithm with the proposed D-HC. All the results presented in the following are appropriately scaled[6]. The standard HC algorithm is implemented with two different update times $\Delta t = 1.5\,\text{s}$ and $\Delta t = 2\,\text{s}$ which respect the lower bound obtained in Section 7.4.1. On the other hand, the proposed algorithm eliminates the coupling interaction and, therefore, the choice of the update time becomes independent from the presence of the passive device. Indeed, the D-HC algorithm is implemented with an update time $\Delta t = 0.5\,\text{s}$. In all cases, the control system is activated after $5\,\text{s}$ and the response is scaled by the steady-state value achieved when the active control is turned off. For the ease of visualization, we show the results of this comparison only for a particular output/accelerometer, but the same considerations are valid for the other outputs. In particular, Figure 7.12 shows the response with the different HC algorithms, while Figure 7.13 and Figure 7.14 show the results in terms of cosine and sine components of the considered output, respectively. The standard HC response obtained with an update time $\Delta t = 1.5\,\text{s}$ presents an unpleasant oscillatory behaviour produced by the periodic excitation of the control action that couples with the MVA; by increasing the update time to $\Delta t = 2\,\text{s}$ this effect is moderated, but the slow update time weakens the effectiveness of the controller to time-varying disturbance as discussed in Remark 40. On the other hand, the D-HC algorithm can be implemented with an update time $\Delta t = 0.5\,\text{s}$ and the oscillatory behaviour produced by the coupling between the active control and the MVA is eliminated.

We can state that the proposed approach accommodates successfully the coupling term between the ACSR and the MVA. However, the price to pay to suppress this interaction is a reduced control authority due to the reduced space in which the controller is allowed to operate.

[6] Note that the amplitude of the disturbance can be arbitrarily chosen due to the linearity of the system.

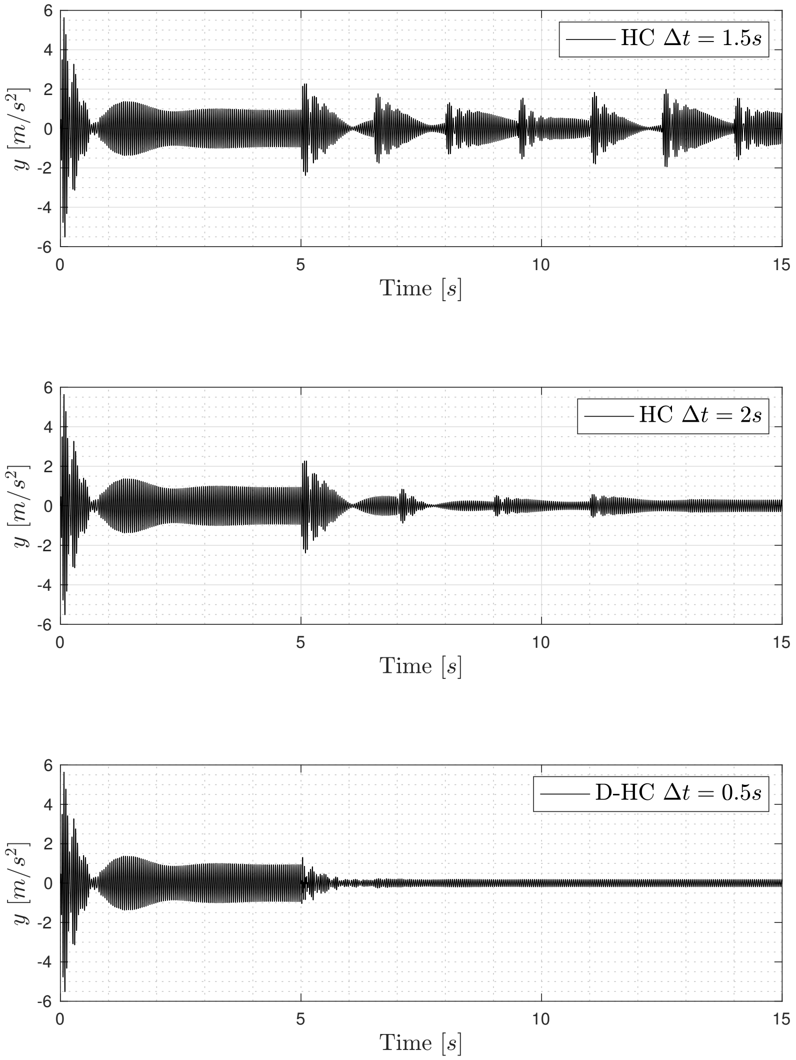


Figure 7.12: *Closed-loop response of the ACSR with different HC algorithms.*

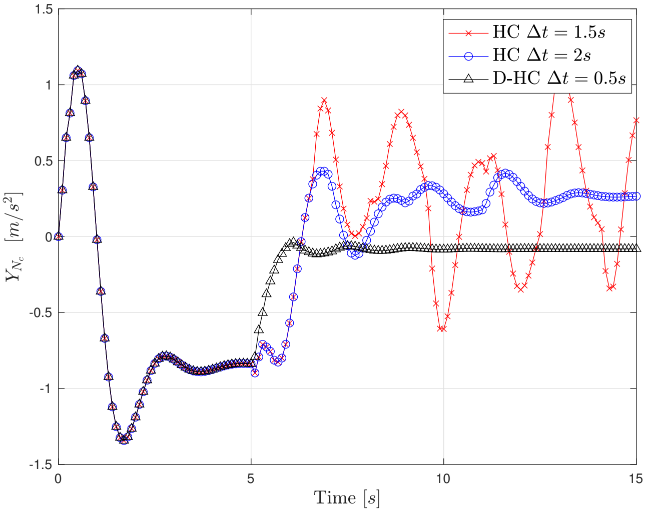


Figure 7.13: *Closed-loop response of the ACSR: cosine components.*

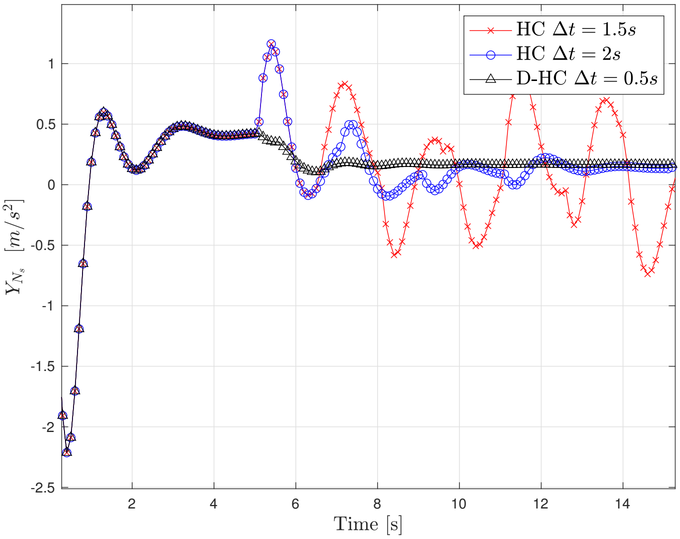


Figure 7.14: *Closed-loop response of the ACSR: sine components.*

Part III

Iterative Learning Control

CHAPTER 8

Iterative Learning Control theory

Iterative Learning control (short, ILC) is based on the idea that the performance of a system that performs the same task multiple times may be improved by learning from previous executions (iterations, trials, passes) [186]. Indeed, in describing the ILC technique, the term *iterative* is used for the recursive nature of the system operation, and the term *learning* is used for the modification of the input signal based on past performance in executing a task or trajectory. Roughly speaking, the ILC idea is to utilize the system repetitions as *experience* to improve the system control performance under incomplete *knowledge* of the system to be controlled. The fundamental ideas and aspects regarding ILC algorithms are described in this Chapter, starting from the definition of a particular class of repetitive systems, known in the literature as multipass systems.

8.1 Repetitive systems

The control designer selects the controller structure based on the dynamic characteristics of the system and the control objective. In the context of this Part, the special property of the system dynamics is the repetitive nature of its operation. Repetition is a common feature in control applications and takes several forms, described in the following.

8.1.1 Periodic signals and tonal disturbances

The simplest notion of repetition is evident in the situation where the reference signal and/or disturbances acting on the system are periodic. Periodicity can be interpreted as a repetition of these signals on the first interval $0 \leq t < T$ on subsequent intervals $kT \leq t < (k+1)T$ for all integers $k \geq 1$. Control design for such systems lies within classical control, and asymptotic tracking is achieved using feedback control with compensators based on the idea of the Internal Model Principle [187]. This research field is relevant to applications characterized by disturbances that have a known source and period, but their amplitude is unknown. Part II examined these applications in detail.

8.1.2 Multipass systems and Repetitive Control

Multipass systems evolve over a finite time interval $0 \leq t < T$ after which both the system and the clock are reset and the operation repeated. This procedure is then repeated over and over again. Repetitions are frequently referred to as trials, passes, or iterations. The particular dynamic characteristic is that the output signal from each iteration affects the next one, essentially acting as a correlated disturbance altering the next iteration dynamics. This idea can be illustrated by the example, taken from [188], of a recursive first order differential equation defined as

$$\frac{dy_{j+1}(t)}{dt} = -y_{j+1}(t) + 2y_j(t) + 1, \quad y_j(0) = 0 \quad j \geq 0 \tag{8.1}$$

with starting iteration condition $y_0(t) = 0$. The term $y_j(t)$ represents the output on j-iteration and the term $2y_j(t)$ represents the effect of j-iteration on the dynamics of the $(j+1)$-iteration. The term "$-y_{j+1}(t)$" ensures the *time-domain* stability of the response on each iteration, but a problem occurs after a large number of iterations. Specifically, assuming that the outputs converge to a limit, *i.e.*, $\lim_{j \to \infty} y_j(t) = y_\infty(t)$, this limit is the solution of the equation:

$$\frac{dy_\infty(t)}{dt} = y_\infty(t) + 1, \quad y(0) = 0 \tag{8.2}$$

We can notice that a series of stable iterations on a dynamical system can converge to an unstable, and hence unacceptable, dynamical behaviour. Effective control is, therefore, needed to ensure good system operation in a repetitive environment. A control architecture, often called *Repetitive Control*, grew out of the area of multipass systems theory [189].

8.2 Introduction to ILC

ILC is a particular case of *Repetitive Control*. It is relevant to applications involving trajectory tracking control over a finite time interval [190]. ILC focuses on problems where the interaction among iterations is about zero, but where repetition of a specific task creates the possibility of improving performance from one trial to the next. In terms of engineering systems, the mechanism of improvement is the systematic use of data collected from previous repetitions to create structured interactions that promote controlled performance. The combined plant and control system is thus a multipass process by design rather than by physical necessity. The repetitive nature of the resulting control schemes puts the topic in the area of classical control design with the added complexity of controlling iteration-to-iteration dynamics [191].

8.2.1 Historical background

The concept of ILC can be traced back to an article published in Japanese by Uchyama [192], which is sometimes cited as the origin of the topic. However, the works by Arimoto *et al.* [193, 194] provided a formal definition of ILC and are the most widely cited. The theory behind ILC is motivated by human learning. Humans learn by practicing or repeating a task until they perfect it. Arimoto *et al.* [193] studied whether similar methods could be applied to robotics, to produce a form of autonomous learning without the need for human intervention. The development of ILC grew originally from practical issues in industrial robotics, where repetitive motions appear in many applications, *e.g.*, mass production on an assembly line [195, 196, 197]. Manufacturing, robotics, and rotary systems are some examples [198]. Early works focused on analysing the convergence and robustness of ILC algorithms, mainly on linear systems [199] or the particular class of nonlinear systems represented by dynamic models of robots [200]. In recent decades, numerous ILC algorithms have been proposed and several research directions have been explored. The book by Bien and Xu [201] covers analysis, design and applications explored in the 1990s. Another commonly cited paper [202] deals with the practical aspects of design and implementation. Through this theory consolidation, ILC has also been successfully applied to maneuvering autonomous vehicles [203] and multicopters [204, 205].

In conclusion, the field of ILC is still growing, and there are many theoretical and practical issues to be investigated. The research field of ILC is covered by a range of surveys, see, *e.g.*, [201, 206, 11, 190]. In [186] Moore provides a detailed overview of the ILC area and categorizes most of the publications up to 1998, while Ahn in [198] discusses and classifies the literature published between 1998 and 2004. A topological classification of the general results is provided in [207] containing a list of references related to robotics and aerospace applications.

8.2.2 ILC architecture

The ILC main idea is to use the tracking error of the previous trials to update the control input signal for the current trial of the ILC, and the goal is to achieve better tracking performance from trial to trial. The block diagram in Figure 8.1 shows the structure of a generic ILC closed-loop system and the learning trend of related signals.

Remark 41. *It should be pointed out that ILC is not an open-loop control operation, although ILC only modifies the input command for the next repetition. ILC closes the loop in the iteration-domain since updates are performed for the next trial using the feedback measurements of the previous one. This feedback architecture is similar to the closed-loop struc-*

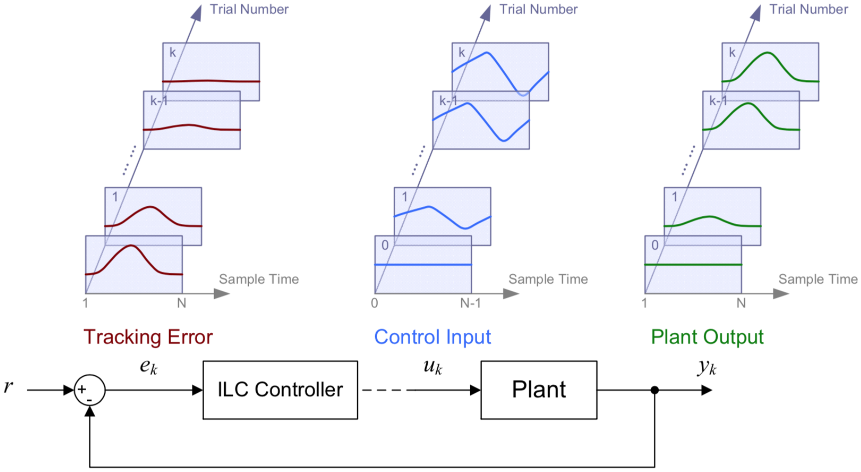


Figure 8.1: *General block diagram of ILC system [208].*

ture of conventional controllers in the time-domain, which updates the control signal of the next time step using the feedback at current or past time steps. The difference is that the ILC mechanism is in the repetition-domain, making it appears as open-loop control in the time-domain.

8.3 ILC problem formulation

The general ILC architecture is illustrated in Figure 8.2. All the signals shown are assumed to be defined on a finite interval $t \in \left[0, t_f\right]$. The subscript j indicates the trial or iteration number. These signals are stored in the memory units until the trial is over, at which point they are processed off-line by the ILC algorithm[1]. The ILC scheme works as follows: during the j-trial an input $u_j(t)$ is applied to the system, producing the output $y_j(t)$. The ILC algorithm computes a modified input signal $u_{j+1}(t)$ based on the observed error between the desired output and the actual output and stores it in memory until the next iteration, at which time the new input signal $u_{j+1}(t)$ is applied to the system. This new input $u_{j+1}(t)$ should

[1] Actually, several ILC algorithms does not require to wait until the trial end to do the processing [198].

be designed to reduce the error with respect to the previous iteration.

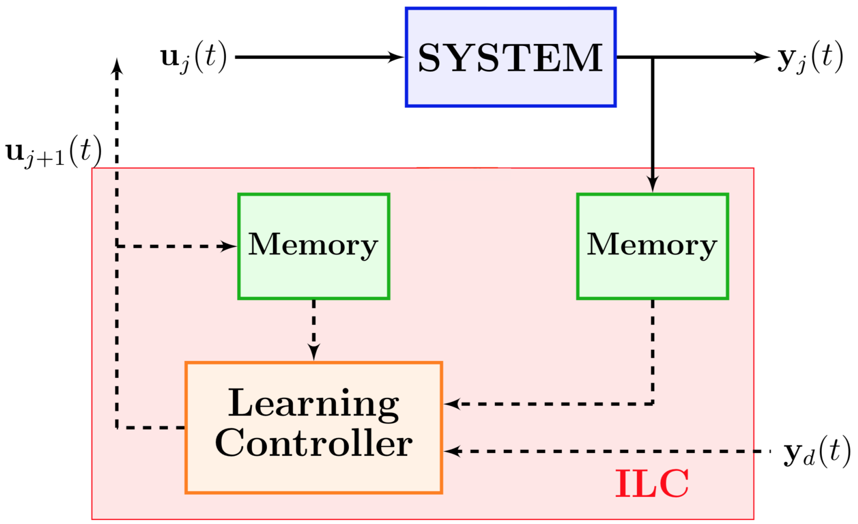


Figure 8.2: *General ILC architecture.*

The ILC approach can be formalized by introducing some additional notation [186]. We define the nonlinear operator $f : U \mapsto Y$ that maps elements in the vector space U to those in the vector space Y be written as $y = f(u)$ where $u \in U$ and $y \in Y$. Given a system S, defined by $y(t) = f_S(u(t),t)$, the ILC objective is to drive the output to a desired response defined by $y_d(t)$. This is equivalent to finding the optimal input $u^*(t)$ that satisfies

$$\min_{u(t)} \|y_d(t) - f_S(u(t),t)\| = \|y_d(y) - f_S\left(u^*(t),t\right)\| \ , \tag{8.3}$$

where $\|(\cdot)\|$ is a suitable defined norm. ILC solves this problem generating a sequence of inputs $u_j(t)$ in such a way that the sequence converges to $u^*(t)$. Namely, the input is updated with the following recursive law:

$$u_{j+1}(t) = f_L\left(u_j(\tau), y_j(\tau), y_d(\tau), t\right) \qquad \tau \in \left[0, t_f\right] \tag{8.4}$$

such that

$$\lim_{j\to\infty} u_j(t) = u^*(t) \qquad \text{for all } t \in \left[0, t_f\right] \ . \tag{8.5}$$

The general algorithm (8.4) introduces a new variable τ. This reflects the fact that there is effectively no causality restriction in the ILC operator f_L [209]. Thus one may construct the input $u_{j+1}(t_k)$ using information about what happened after the input $u_j(t_h)$ was applied with $t_h > t_k$.

Assumption 4. *A number of assumptions that underlie ILC approaches have been formulated by Arimoto in [193, 194] and are listed in the following.*

- *The iterations are finite in time, i.e., $t_f \in (0, \infty)$.*
- *The desired reference is defined a priori for all $t \in [0, t_f]$.*
- *The dynamics of the system is invariant throughout the repeated iterations or at least slowly iteration-varying [210];*
- *The initial conditions of the system are reset at the beginning of each iteration to the same value, i.e., $y_j(0) = y_0$ for all j. This is a key assumption known as the "initial reset condition".*

Remark 42. *ILC offers several advantages over conventional feedback and feedforward control, including the ability to anticipate and respond immediately to recurrent disturbances. This capability is possible because of the non-causality of the ILC algorithm. Indeed, since the entire temporal sequence of data is available from all previous iterations, a noncausal ILC algorithm is implementable in practice in contrast to the conventional conception of noncausality [190]. In [211], Goldsmith et al. demonstrated that when considering a causal ILC algorithm, the asymptotic control signal $u^*(t)$ could be obtained from a standard feedback controller. In contrast, when the ILC algorithm is noncausal, the control signal $u^*(t)$ cannot be provided by an equivalent feedback controller except for special cases, see, e.g., [212].*

There are several possible modifications to the learning algorithm (8.4). For instance, *higher-order* learning algorithms compute u_{j+1} using the error e_i and the control input u_i of N_0 previous iterations iterations $i \in \{j - N_0 + 1, \ldots, j\}$. Another modification is the *current-iteration* learning algorithm, in which the current error e_{j+1} is leveraged in real-time to compute the current control input u_{j+1}. As shown in [207], the *current-iteration* learning algorithm is equivalent to the standard ILC algorithm combined with a feedback controller (see next Section).

8.3.1 ILC properties

ILC can be categorized as an intelligent control methodology[2]. In contrast to other intelligent techniques, which are based on artificial intelligence and computer science, *e.g.*, artificial neural networks [214] or fuzzy logic [215], ILC theory is based on a system-theoretic approach. Consequently, ILC can guarantee a fast convergence, unlike other strategies that require extensive data training [191]. ILC aims to generate a control input signal that follows a desired trajectory or rejects a recurrent disturbance. Compared to a well-designed feedback and feedforward controller, ILC has a number of benefits. For instance, a feedback controller always has a lag in transient tracking because it reacts to inputs and disturbances. Only known or measurable signals, like the reference, and typically not disturbances, can be eliminated by feedforward control. In contrast, ILC is anticipatory and can react in advance to exogenous signals by leveraging experience from previous iterations. Specifically, ILC only requires that these exogenous signals (references or disturbances) repeat from one iteration to the next [190].

Remark 43. *A feedforward controller only performs well when the system dynamics is known precisely. In contrast, ILC approach computes its control signal based on experience (feedback in the iteration-domain), making the closed-loop system remarkably robust to uncertainties or unmodelled dynamics [11] and achieving low tracking errors.*

Obviously, ILC cannot provide perfect tracking in every situation. In fact, noise and non-repeating disturbances could degrade ILC performance [198]. To overcome these limitations, a feedback controller is typically employed in combination with the ILC. ILC algorithm can be integrated with a feedback loop in two different ways, as shown in Figure 8.3 and

[2] Definition from [213]: *Intelligent control uses conventional control methods to solve lower level control problems ... conventional control is included in the area of intelligent control. Intelligent control attempts to build upon and enhance the conventional control methodologies to solve new, challenging control problems.*

Figure 8.4. We refer to the first configuration as *serial*, because the ILC control input is applied before the feedback loop, *i.e.*, it alters the reference signal commanded to the closed-loop system [202]. This approach is essential when applying ILC algorithm to a pre-existing system that makes use of a commercial controller that prevents the system control signal modification. A sufficient condition for asymptotical stability of the serial configuration was derived in [216].

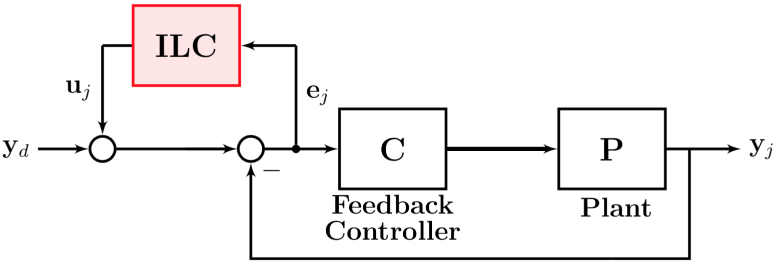


Figure 8.3: *ILC in serial arrangement.*

Instead, we refer to the second configuration as *parallel*, because the ILC control input is added to the feedback control signal, *i.e.*, it directly alters the control signal to the system [217].

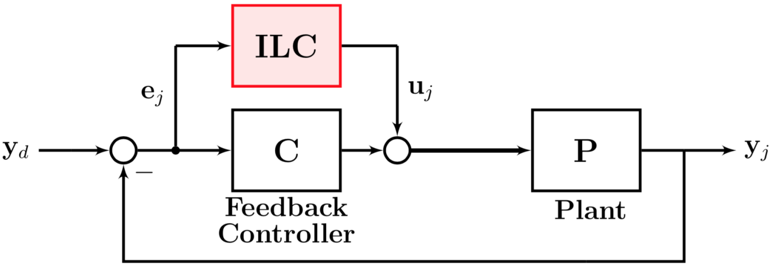


Figure 8.4: *ILC in parallel arrangement.*

Remark 44. *Note that setting the ILC input to zero in both configurations yields the typical feedback-controlled response to the reference signal. Accordingly, whenever it is necessary to perform non-repeating tasks in either of these arrangements, the ILC can be turned off.*

8.4 ILC algorithms and results

In this Section, we introduce and formalize the problem of trajectory tracking in the ILC framework. The formulation is general and can be applied to any dynamic system (including systems with underlying feedback loops). Firstly, we briefly describe the system to be controlled following the approach in [11] focusing on Linear Time Invariant Single Input Single Output (LTI SISO) systems for the sake of simplicity. Then, we introduce the most used system representations in the ILC framework. Finally, we analyse the results in terms of convergence (stability), performance, and robustness for a general ILC algorithm.

8.4.1 System description

Consider the discrete-time[3] LTI SISO 2-dimensional system

$$y_j(k) = F(\mathrm{q})u_j(k) + d(k) \quad \text{with} \quad y_j(0) = y_0 \tag{8.6}$$

where q is the forward time-shift operator $\mathrm{q}x(k) \equiv x(k+1)$, y_j is the output, u_j is the control input, and d is an exogenous signal[4] that repeats at each iteration; the plant $F(\mathrm{q})$ is a proper rational function of q with a delay, or equivalently relative degree, of m; k represents discrete-time points along the time axis and the subscript j represents the iteration trial number along the iteration axis. Notice that $y_j(0) = y_0$ for all j. We assume that $F(\mathrm{q})$ is asymptotically stable[5] and the plant delay $m = 1$. The last assumption is made only in this Section to simplify the system description without loss of generality. For the analysis with a generic plant delay the interested reader is referred to [11]. In the following, we consider the

[3] Since ILC demands storing data from previous iterations, which are frequently sampled, discrete-time is the most suitable domain for it.

[4] The term $d(k)$ encloses several effects, such as, *e.g.*, repeating disturbances and non-zero repeated initial conditions.

[5] When $F(\mathrm{q})$ is not asymptotically stable, it can be stabilized with a feedback controller, and the ILC can be applied to the closed-loop system. This agrees with the focus of the ILC algorithm that is to improve the performance of the system [209].

N-sample sequence of inputs and outputs:

$$\begin{aligned} u_j(k), &\quad k \in \{0,1,\ldots,N-1\} \\ y_j(k), &\quad k \in \{1,2,\ldots,N\} \\ d(k), &\quad k \in \{1,2,\ldots,N\} \end{aligned}$$

and the desired system output:

$$y_d(k), \quad k \in \{1,2,\ldots,N\} .$$

The performance or error signal is defined by $e_j(k) = y_d(k) - y_j(k)$.

8.4.2 General ILC Algorithm

Originally, ILC was formulated for continuous systems in a continuous-time framework. Indeed, the first ILC algorithm proposed by Arimoto is defined in continuous time [194]:

$$u_{j+1}(t) = u_j(t) + \gamma \frac{d}{dt} e_j(t) , \tag{8.7}$$

where the constant γ is the learning gain that multiplies the derivative of the continuous tracking error in time. For a large class of systems, it can be shown that this algorithm converges in the sense that, as $j \to \infty$, we have $y_j(t) \to y_d(t)$ for all $t \in \left[0, T_f\right]$ [191]. The discrete-time counterpart of the control law (8.7) is given by:

$$u_{j+1}(k) = u_j(k) + \gamma e_j(k+1) . \tag{8.8}$$

The algorithm (8.8) can be seen as a special case of the widely used ILC learning algorithm [11, 190, 198]:

$$u_{j+1}(k) = Q(\mathrm{q}) \left[u_j(k) + L(\mathrm{q})\ e_j(k+1)\right] , \tag{8.9}$$

where $Q(\mathrm{q})$ and $L(\mathrm{q})$ are LTI systems referred as to the Q-filter and learning function, respectively. This algorithm is schematized in Figure 8.5.

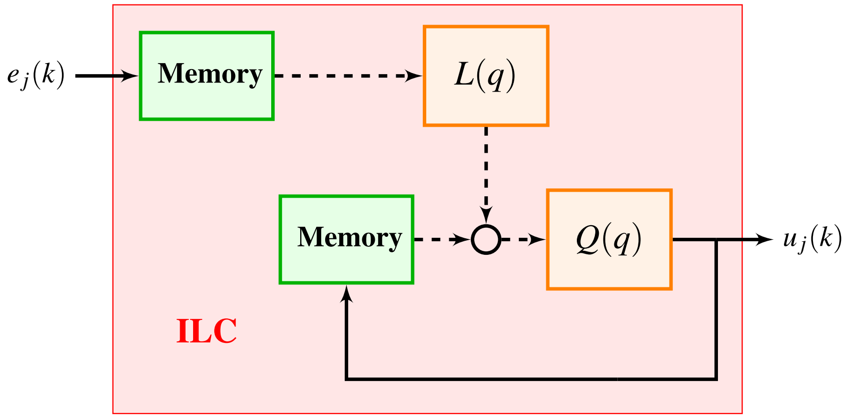


Figure 8.5: *Representation of the ILC algorithm.*

8.4.3 System representations

In this Section, we introduce and formalize the most commonly used representations in the literature:

- Lifted-system (*Time-domain*),
- z-Domain (*Frequency-domain*).

Time-domain analysis using the lifted-system framework

When the system is described in the *time-domain*, the so called *lifted*-representation is preferred in describing the input/output relation and the ILC update algorithm [11]. The lifted form, in fact, allows to write the SISO time and iteration-domain dynamic system (8.6) as a Multiple Input Multiple Output (MIMO) iteration-domain dynamic system. To obtain the lifted form, the rational LTI plant (8.6) is first expanded as an infinite power series by dividing its denominator into its numerator:

$$F(\mathrm{q}) = f_1\mathrm{q}^{-1} + f_2\mathrm{q}^{-2} + f_3\mathrm{q}^{-3} + \cdots \tag{8.10}$$

where the coefficients f_k are the Markov parameters. Note that $f_1 \neq 0$ since $m = 1$ is assumed. Considering the state space description

$$x_j(k+1) = Ax_j(k) + Bu_j(k) \tag{8.11}$$

$$y_j(k) = Cx_j(k) \tag{8.12}$$

we have that $f_k = CA^{k-1}B$. Stacking the signals in vectors, the system dynamics in (8.6) can be written equivalently as the *lifted* system:

$$\underbrace{\begin{bmatrix} y_1(1) \\ y_1(2) \\ \vdots \\ y_y(N) \end{bmatrix}}_{Y_j} = \mathbf{F} \underbrace{\begin{bmatrix} u_i(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{U_j} + \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{D} \tag{8.13}$$

with

$$\mathbf{F} := \begin{bmatrix} f_1 & 0 & \cdots & 0 \\ f_2 & f_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_N & f_{N-1} & \cdots & f_1 \end{bmatrix} \in \mathbb{R}^{N \times N} \ . \tag{8.14}$$

The components of Y_j and D are shifted by one time step to accommodate the plant delay, ensuring that the diagonal entries of $\mathbf{F}$ are non-zero. Similarly, the ILC algorithm (8.9) can be written in *lifted* form. In particular, the Q-filter and learning function can be noncausal functions with the impulse responses given as

$$Q(\mathrm{q}) = \cdots + q_{-2}\mathrm{q}^2 + q_{-1}\mathrm{q}^1 + q_0 + q_1\mathrm{q}^{-1} + q_2\mathrm{q}^{-2} + \cdots \ , \tag{8.15}$$

$$L(\mathrm{q}) = \cdots + l_{-2}\mathrm{q}^2 + l_{-1}\mathrm{q}^1 + l_0 + l_1\mathrm{q}^{-1} + l_2\mathrm{q}^{-2} + \cdots \ . \tag{8.16}$$

In *lifted* form, equation (8.9) becomes

$$U_{j+1} = \mathbf{Q}\, U_j + \mathbf{L}\, E_j \tag{8.17}$$

with

$$\mathbf{Q} := \begin{bmatrix} q_0 & q_{-1} & \cdots & q_{-(N-1)} \\ q_1 & q_0 & \cdots & q_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \in \mathbb{R}^{N \times N} , \tag{8.18}$$

$$\mathbf{L} := \begin{bmatrix} l_0 & l_{-1} & \cdots & l_{-(N-1)} \\ l_1 & l_0 & \cdots & l_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ l_{N-1} & l_{N-2} & \cdots & l_0 \end{bmatrix} \in \mathbb{R}^{N \times N} . \tag{8.19}$$

Frequency-domain analysis using the z-domain representation

The one-sided z-transformation of a signal $\{x(k)\}_{k=0}^{\infty}$ is defined as

$$\mathcal{X}(z) = \sum_{k=0}^{\infty} x(k) z^{-1} ,$$

and the z-transformation of a system is obtained by replacing q with z. The frequency response of a z-domain system is given by replacing z with $e^{j\omega}$ for $\omega \in [-\pi, \pi]$. The z-domain representations of the system dynamics in (8.6) and ILC algorithm in (8.9) are respectively

$$\mathcal{Y}_j(z) = \mathcal{F}(z)\mathcal{U}_j(z) + \mathcal{D}(z) \tag{8.20}$$

and

$$\mathcal{U}_{j+1}(z) = \mathcal{Q}(z) \left[\mathcal{U}_j(z) + z\mathcal{L}(z)\mathcal{E}_j(z) \right] , \tag{8.21}$$

where $\mathcal{E}_j(z) = \mathcal{Y}_d(z) - \mathcal{Y}_j(z)$. The z multiplying the learning function $\mathcal{L}(z)$ underlines the noncausality of the ILC algorithm in equation (8.9).

Remark 45. *The signals must be defined over an infinite time horizon, i.e., $N = \infty$, to use the z-transformation. Accordingly, since practical applications have finite trial lengths, the z-transformation yields an approximate representation of real-world ILC systems.*

8.4.4 Analysis of performance

In this Section, we summarize the properties of the ILC algorithm (8.9) in terms of convergence (stability), performance and robustness [190, 218]. The linear system (8.6) is assumed to be subject to unstructured multiplicative uncertainty. Therefore, the method takes into account a set of uncertain systems $\mathcal{F}_\Delta(z)$ given as follows:

$$\mathcal{F}_\Delta(z) = \hat{\mathcal{F}}(z)(1+\Delta(z)\mathcal{W}(z)), \quad \Delta(z) \in \mathcal{B}_\Delta \tag{8.22}$$

with

$$\mathcal{B}_\Delta = \{\Delta(z) = \text{ stable, LTI system } : \|\Delta(z)\|_\infty \leq 1\} \tag{8.23}$$

where $\|\cdot\|_\infty$ is the H_∞ norm, $\hat{\mathcal{F}}(z)$ is the nominal plant model, and the uncertainty weighting function $\mathcal{W}(z)$ determines the size of the uncertainty which is assumed to be known.

Robust convergence and convergence speed

In [218], Son *et al* demonstrated that the ILC system (8.9) achieves robust convergence if the following condition is satisfied:

$$\sup_{\Delta\in\mathcal{B}_\Delta} ||\mathcal{Q}(z)\,[1-z\mathcal{L}(z)\mathcal{F}_\Delta(z)]||\,|_\infty = \gamma^* < 1, \quad \forall\mathcal{F}_\Delta(z),\, \forall z \in \mathcal{D} \tag{8.24}$$

with $\mathcal{D} = \left\{z = e^{j\omega}|\omega \in [-\pi,\pi]\right\}$. Furthermore, following the steps in [219], the following relation is obtained for sufficiently large trial lengths:

$$\mathcal{E}_\infty(z) - \mathcal{E}_j(z) = \mathcal{Q}(z)\,[1-z\mathcal{L}(z)\mathcal{F}_\Delta(z)]\left(\mathcal{E}_\infty(z) - \mathcal{E}_j(z)\right)\ . \tag{8.25}$$

Hence, condition (8.24) also guarantees monotonic convergence. In addition, the smaller γ^*, the faster $\mathcal{E}_j(z)$ converges to the optimal $\mathcal{E}_\infty(z)$.

Robust performance

The tracking performance of an ILC system is based on the asymptotic value of the error signal in the *iteration-domain*. Robust performance ILC requires the tracking performance specifications to be met for all plants in the uncertainty set. If the ILC system (8.9) is robust convergent, the asymptotic tracking error is given by [11]:

$$e_\infty(k) = \frac{1-Q(\mathrm{q})}{1-Q(\mathrm{q})\left[1-\mathrm{q}L(\mathrm{q})F_\Delta(z)\right]}\,\left(y_\mathrm{d}(k)-d(k)\right)\ . \tag{8.26}$$

Accordingly, the robust performance condition is defined in the *frequency-domain* as

$$\left\|\left|\mathcal{W}_p(z)\frac{1-\mathcal{Q}(z)}{1-\mathcal{Q}(z)\left[1-z\mathcal{L}(z)\mathcal{F}_\Delta(z)\right]}\right|\right\|_\infty < 1,\ \forall\mathcal{F}_\Delta(z),\ \forall z\in\mathcal{D} \tag{8.27}$$

where $\mathcal{W}_p(z)$ is the performance weight selected by the designer.

Remark 46. *Typically, robust convergence* (8.24) *and asymptotic performance* (8.26) *are considered in the ILC design. Assuming exact knowledge of the system model, a trivial solution, with the learning function L as the inverse of the model and Q-filter equal to 1, achieves convergence of the error to zero in only one iteration. However, since models are never perfect and systems may be non-minimum phase, this result does not occur under real-world circumstances [11].*

8.5 Typical design techniques

Since the classic Arimoto-type ILC algorithm, numerous algorithms have been proposed and several research directions have been explored, as reflected in the categorization presented in Moore [186] and Ahn *et al.* [190]. We can divide the types of algorithms into two different categories: *basic* algorithms and *model-based* algorithms [220]. The *basic* type of algorithms is easy to implement and tune with few parameters requiring very little knowledge of the real system, such as the static gain and time delay of the system. As a result, this type is very attractive from an industrial point of view [202]. On the other hand, *model-based* algorithms require an explicit model of the system, and the methods are often more computationally demanding. However, Gunnarsson et *al.* suggested that more knowledge of the system could improve performance [221].

In this Section, some examples of *basic* and *model-based* design methods are given, limiting the discussion to linear ILC algorithms[6].

[6] Nonlinear algorithms have also received much attention in ILC research, both in analysis and algorithm

8.5.1 Basic design methods

The inherent message for designing and implementing *basic* ILC algorithms in practical applications is clearly expressed in [226] by the following words: “Try simple things first.” Indeed, the ILC control law (8.7) proposed by Arimoto involves only a simple Derivative (*D*) action. The Proportional (*P*)-, *D*-, and *PD*-type learning functions are undoubtedly the most popular learning techniques, especially for nonlinear systems [227]. For instance, the discrete-time *PD*-type control law can be written as

$$u_{j+1}(k) = u_j(k) + k_p e_j(k+1) + k_d[e_j(k+1) - e_j(k)] \ , \tag{8.28}$$

where k_p is the proportional gain and k_d is the derivative gain. These approaches do not require an accurate model for implementation and are based on fine-tuning or autotuning techniques [190]. The interested reader is referred to [228] for a comparison of different tuning methods.

8.5.2 Dynamic inversion methods

Dynamic inversion methods use models of the inverted system dynamics as the learning function $L(\mathrm{q})$. The discrete-time dynamic inversion ILC algorithm is given by

$$u_{j+1}(k) = u_j(k) + \hat{F}^{-1}(\mathrm{q})e_j(k) \ . \tag{8.29}$$

Convergence happens after just one iteration and the converged error is equal to zero if $\hat{F}(\mathrm{q})$ is an exact model of the plant. However, the implementation of such algorithms presents several practical challenges. Dealing with non-minimum phase systems, where the direct inversion of the system dynamics $F_\Delta(\mathrm{q})$ results in an unstable filter, is one of the immediate challenges with the plant inversion approach. Although finite-duration iterations guarantee bounded signals, the unstable filter produces undesirable large control signals [188]. A stable inversion method, which yields a noncausal learning function, can be used to get around this issue [229].

Furthermore, the effectiveness of the plant inversion methods depends heavily on the model accuracy, regardless of whether $F_\Delta(\mathrm{q})$ is in minimum phase or not. Poor transient behaviour can result from a mismatch

design, see *e.g.*, [222, 223, 224, 225].

between the model $\hat{F}(\mathrm{q})$ and the real dynamics $F_\Delta(\mathrm{q})$, which prevents convergence from happening in one iteration and can lead to instability. To avoid these obstacles, a low-pass Q-filter is typically used [190].

8.5.3 Frequency-domain design methods

A different design methodology is built in *frequency-domain*, which involves ILC algorithms invariant in time and iteration domain [190]. One example is the H_∞ design method, which offers a systematic approach to select the learning gains. In particular, the objective is to select the learning function $L(\mathrm{q})$ that offers the fastest convergence rate for a given Q-filter, *i.e.*, solving the optimization problem:

$$\mathcal{L}^*(z) = \arg\min_{\mathcal{L}} \| \mathcal{Q}(z)(I - z\mathcal{L}(z)\mathcal{F}_\Delta(z)) \|_\infty \qquad (8.30)$$

This problem can be recast as a Linear Fractional Transform (LFT) as

$$\mathcal{Q}(z)(I - z\mathcal{L}(z)\mathcal{F}_\Delta(z)) = \mathcal{G}_{11}(z) + \mathcal{G}_{12}\mathcal{L}(z)\left(I - \mathcal{G}_{22}(z)\mathcal{L}(z)\right)^{-1}\mathcal{G}_{21}(z)\ ,$$

where

$$\mathcal{G}(z) = \begin{bmatrix} \mathcal{G}_{11}(z) & \mathcal{G}_{12}(z) \\ \mathcal{G}_{21}(z) & \mathcal{G}_{22}(z) \end{bmatrix} = \begin{bmatrix} \mathcal{Q}(z) & \mathcal{Q}(z) \\ -z\mathcal{F}_\Delta(z) & 0 \end{bmatrix}. \qquad (8.31)$$

The resulting scheme is depicted in Figure 8.6. Exploiting this formulation, standard H_∞ synthesis tools can be employed to solve (8.30). Furthermore, a powerful feature of the H_∞ synthesis is that the formulation can be extended to include models with known uncertainty bounds [217].

Remark 47. *Note that the solutions obtained through H_∞ synthesis are optimal for a given Q-filter. Indeed, iterating over the Q-filter bandwidth is required to find the optimal trade-off between convergence rate and performance [217]. To address this drawback, Son et al. suggested in [230] an ILC architecture that solves a one-step optimization for non-causal Q-filter and learning function* $L(\mathrm{q})$ *including several ILC objectives. Namely, ILC objectives are merged into a single constrained opti-*

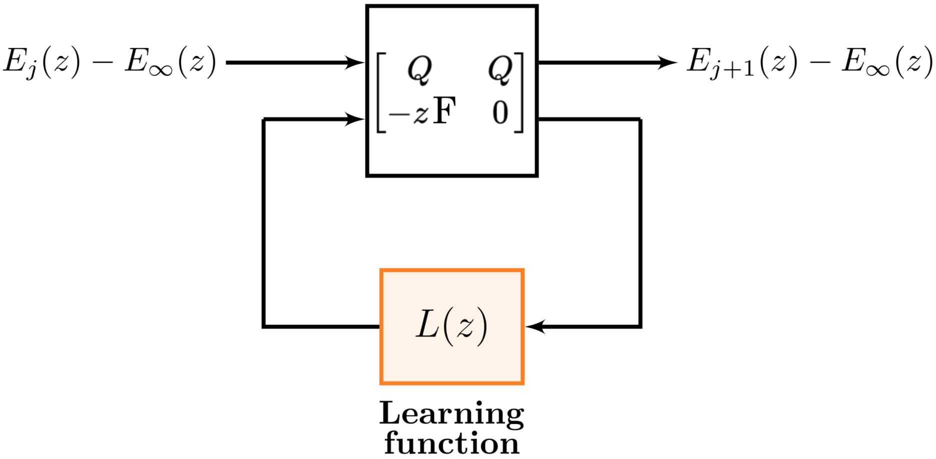

$E_j(z) - E_\infty(z)$
$\begin{bmatrix} Q & Q \\ -z\mathrm{F} & 0 \end{bmatrix}$
$E_{j+1}(z) - E_\infty(z)$
$L(z)$
Learning function

Figure 8.6: *H_∞-synthesis of ILC scheme.*

mization problem for $L(\mathrm{q})$ and $Q(\mathrm{q})$, as follows:

$$\begin{aligned} \underset{L(\mathrm{q}),Q(\mathrm{q})}{\textit{minimize}} \quad & \textit{convergence speed} \\ \textit{subject to} \quad & \textit{robust convergence} \\ & \textit{robust performance} \\ & \textit{input constraint} \end{aligned} \tag{8.32}$$

The problem was also reformulated as a convex problem in [218], ensuring an accurate and efficient computation of the global optimum.

8.5.4 Optimization-based methods

Using the lifted representation of the system and ILC algorithm, the algorithm design can be considered in the context of numerical optimisation, see, *e.g.*, [231, 232]. One example is the quadratically optimal (Q-ILC) approach, also known as *norm-optimal* ILC [233]. The Q-ILC learning functions are designed to minimize a quadratic next iteration cost function. Typically, the functions used have the following form:

$$J_{j+1}\left(U_{j+1}\right) = E_{j+1}^T W_e E_{j+1} + U_{j+1}^T W_u U_{j+1} + \Delta U_{j+1}^T W_{\Delta u} \Delta U_{j+1} \tag{8.33}$$

where $E_{j+1} = \mathbf{F}\Delta U_{j+1} + E_j$ with $\Delta U_{j+1} = U_{j+1} - U_j$, W_e is a positive-definite matrix, and $W_{\Delta u}$, W_u are positive-semidefinite matrices. The optimization formulation enables the integration of input and output constraints ($Z_u U_{j+1} \leq U_{max}$ and $Z_y Y_{j+1} \leq Y_{max}$). The resulting minimization can be formulated as a convex optimization problem which can be solved efficiently by existing commercial software tools [188]. The resulting block diagram is schematized in Figure 8.7.

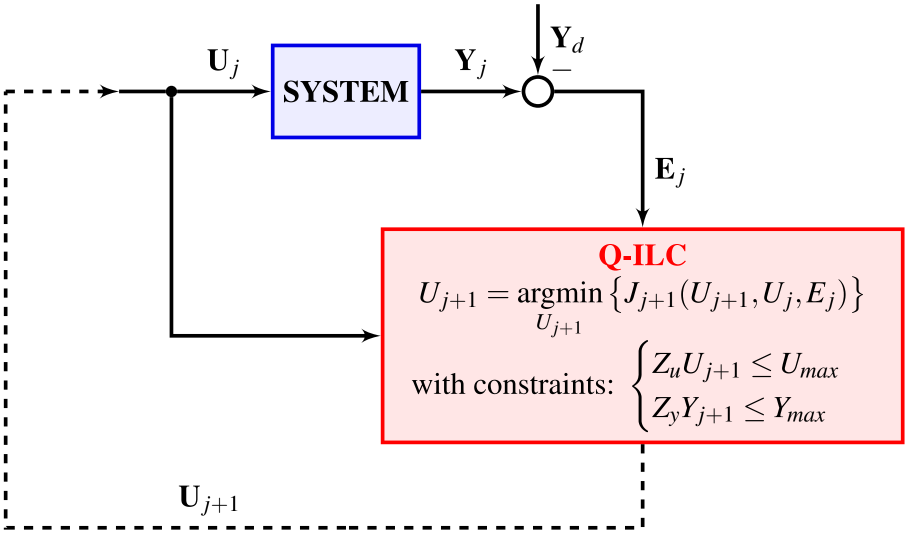

$\mathbf{U}_j$
SYSTEM
$\mathbf{Y}_j$
$\mathbf{Y}_d$
$-$
$\mathbf{E}_j$
Q-ILC
$U_{j+1} = \underset{U_{j+1}}{\text{argmin}} \{J_{j+1}(U_{j+1}, U_j, E_j)\}$
with constraints: $\begin{cases} Z_u U_{j+1} \leq U_{max} \\ Z_y Y_{j+1} \leq Y_{max} \end{cases}$
$\mathbf{U}_{j+1}$

Figure 8.7: *Q-ILC block diagram in lifted form.*

Considering the unconstrained problem and minimizing the cost function with respect to U_{j+1} yields the optimal Q-filter

$$\mathbf{Q}_{opt} = \left(\mathbf{F}^T W_e \mathbf{F} + W_u + W_{\Delta u}\right)^{-1} \left(\mathbf{F}^T W_e \mathbf{F} + W_{\Delta u}\right) \ , \tag{8.34}$$

and optimal learning function

$$\mathbf{L}_{opt} = \left(\mathbf{F}^T W_e \mathbf{F} + W_{\Delta u}\right)^{-1} \mathbf{F}^T W_e \ . \tag{8.35}$$

Remark 48. *The performance, convergence rate, and robustness of the Q-ILC are all governed by the weighting function selection. For instance,* *substituting $\boldsymbol{Q}_{opt}$ and $\boldsymbol{L}_{opt}$ into equation* (8.26)*, we obtain the converged error in terms of the weighting functions:*

$$E_\infty = \left[I - \mathbf{F}\left(\mathbf{F}^T W_e \mathbf{F} + W_u\right)^{-1} \mathbf{F}^T W_e\right](Y_d - D) \ . \tag{8.36}$$

Note that the weighting on the change in control $W_{\Delta u}$ (not present in (8.36)*) does not effect the asymptotic error. On the other hand, following the steps in [11], it can be shown that $W_{\Delta u}$ affects how quickly the ILC converges. Instead, the weight on the control W_u degrades the asymptotic performance. However, selecting $W_u = 0$ is not recommended in practice, because $W_u \neq 0$ may help limit the control action to prevent actuator saturation, particularly for non-minimum phase systems [234]. The robustness of optimization-based approaches is examined in [235] and [236].*

Remark 49. *To reduce the computational effort, a revised version, the fast norm-optimal ILC has been proposed in [220] which is significantly simpler and faster to implement. Some research has been done towards reducing the calculation complexity of the lifted system ILC design problem through the use of basis functions [237]. Furthermore, low-order Q-ILC solutions may also be found in the literature, which reduces the memory requirements associated with the massive matrix form of the Q-ILC method [238]. In particular, Dijkstra et al. proposed low-order solutions that do not depend on the iteration duration and can be extrapolated to iteration durations of arbitrary length [239].*

CHAPTER 9

Smoother-based Iterative Learning Control

Certain problems of practical interest require the generation and repeated execution of a path, where the choice of path and the related tracking accuracy can have a dramatic impact on performance. This repetitiveness creates the possibility of improving performance from one trial to the next. In view of this, the problem can effectively be formulated in the ILC framework. This chapter presents a data-based control approach to achieve high-performance trajectory tracking with Unmanned Aerial Vehicles (UAVs). While we will specifically refer to multirotor platforms for the experimental validation, the formulation can be applied to any dynamic system (including systems with underlying feedback loops). The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance to improve the learning performance.

In the first part of the Chapter, the novel *Smoother-based* ILC algorithm is presented, highlighting common points and differences concerning state of the art described in the previous Chapter. Then, a Monte Carlo analysis is carried out with the aim of showing the performance improvements and limitations of the proposed algorithm with respect to existing approaches. In the last part of the Chapter, experimental results obtained on a quadrotor performing an aggressive manoeuver are reported to show that the proposed approach is capable of remarkably reducing the tracking errors in a few iterations.

9.1 Introduction

In recent years, the study of UAVs has received increasing attention thanks to their wide range of applications. The generation and repeated execution of a path are necessary for some practical problems, where the path choice and the associated tracking accuracy can significantly impact performance. UAVs, for instance, have been widely used for performing a persistent surveillance mission over a very small domain, where they are required to precisely track a desired trajectory in order to perform the task safely and effectively [240]. Another interesting example can be the Airborne Wind Energy (AWE) systems [241]. AWE systems represent an alternative to traditional wind turbines, where the tower is replaced by tethers and a lifting body that either houses an onboard turbine or enables ground-based energy generation through cyclic spooling motion [242], [243]. Because the spool-out/spool-in process is highly repetitive, there is an opportunity to leverage knowledge from previous cycles to optimize our performance during future cycles [244, 245].

Trajectory tracking with UAVs is typically achieved using feedback control approaches [42]. Specifically, linear control techniques are widely used in commercial autopilots, but are not usually able to achieve high performance. On the other hand, non linear methods (*e.g.*, exact inputoutput feedback linearization, backstepping, *etc.*.) can yield controllers with a significantly better performance, but require a careful *ad-hoc* tuning of the parameters. Moreover, the performance of feedback control approaches is limited by the accuracy of the dynamics model and the causality of the control action that is compensating only for disturbances as they

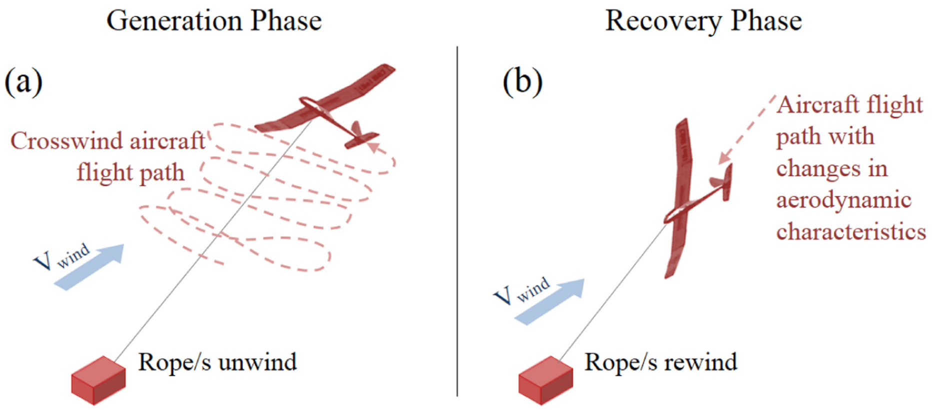

Generation Phase
(a)
Crosswind aircraft flight path
V wind
Rope/s unwind
Recovery Phase
(b)
Aircraft flight path with changes in aerodynamic characteristics
V wind
Rope/s rewind

Figure 9.1: *Scheme of the two-phase energy production for Ground Generation AWE system. (a) The energy generation phase occurs during the unwinding of the ropes as the aircraft performs a crosswind flight. (b) In the recovery phase, the aircraft is controlled to glide towards the ground station [243].*

occur. To overcome these limitations, when a system executes the same task multiple times, ILC can be used.

In recent years ILC algorithms have been also developed and applied to UAVs [246], [244], [47]. In this Chapter, we revisit the data-based control approach presented in [47]. It modifies the reference before sending it to the UAV, therefore it is an add-on algorithm that fits with any commercial controllers. Moreover it requires only the knowledge of the UAV complementary sensitivity function (the transfer function from the reference to the actual trajectory) without going into details of the open-loop dynamics and the baseline controller.

9.2 Proposed architecture

In this Section, we introduce and formalize the novel *Smoother-based* ILC algorithm highlighting the novelties with respect to the existing approaches. While we will specifically refer to multirotor platforms for the experimental validation, the formulation is general and can be applied to any dynamic system (including systems with underlying feedback loops).

9.2.1 Algorithm overview

We combine ILC with the feedback loop in a *serial* configuration, where the ILC control input is applied to the reference before the feedback loop [202] without modifying the feedback controller. As mentioned in the previous Chapter, this concept is useful when applying ILC to a pre-existing system that uses a commercial controller that does not allow to modify the control signal to the plant. Similar to [47], the ILC algorithm uses information on the input and the error at the current trial to design the next trial input that minimizes the tracking error, *i.e.*, the discrepancy between the actual and the desired output at the upcoming iteration. Namely, in [47] Schoellig *et al.* introduce a Kalman filter that estimates the repetitive disturbance based on the current input and error measurement in the *iteration-domain*[1]. This estimate is then used to update the next trial input that is generated by solving an optimization problem (with possible constraints in the input).

On the other hand, the novelty of this work lies in the inclusion of a smoother to estimate the repetitive disturbance to improve the learning performance and speed up the convergence. Namely, a fixed-interval smoothing algorithm is implemented that uses the entire batch of measurements over a fixed interval to estimate all the states in the interval. This smoother can be derived from a combination of two Kalman filters, one of which works forward over the data and the other of which works backward over the fixed interval [251]. In contrast to previous estimation-based ILC algorithms (see, *e.g.*, [252, 253]), the proposed estimator works in the *time-domain* and can be extremely helpful when accuracy is an issue exploiting the potentiality of the batch state estimation [254]. As highlighted in [255], the time-domain estimator in the ILC framework must rely on an accurate system model to not degrade the performance. Indeed, for the experimental activities involving a small quadrotor, the system model has been obtained through a black-box identification procedure using the Predictor-Based Subspace Identification (PBSID) algorithm [256].

[1] Similarly, in [247, 248, 249] the error signal is estimated using a Kalman filter in the *iteration-domain* assuming that the control error is measured directly. Instead, in [250] the error signal is estimated when the control quantity is not directly available as a measurement.

In the following, the two steps of the proposed learning algorithm are described in detail. The overall approach is schematized in Figure 9.2.

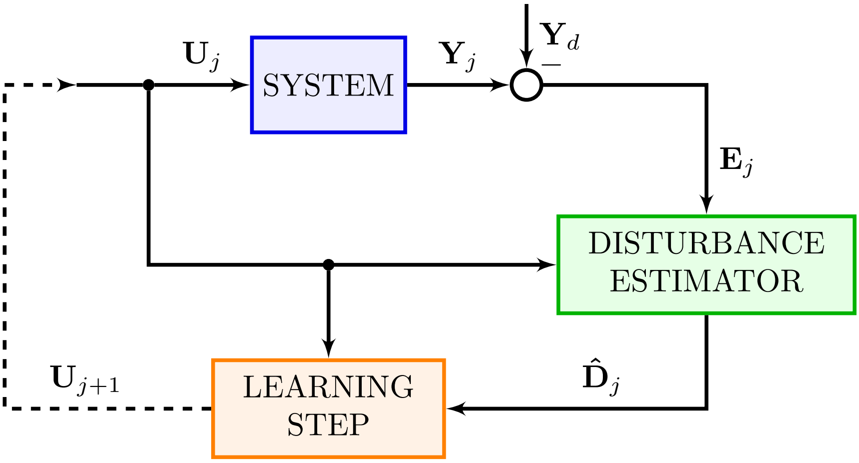


Figure 9.2: *Estimator-based ILC block diagram in the iteration domain.*

9.2.2 Disturbance estimation: Smoother

The iteration-to-iteration learning of ILC provides opportunities for advanced filtering and signal processing [11]. For instance, zero-phase filtering [257], which is non-causal, allows for high-frequency attenuation without introducing lag. Previous works estimated the repetitive disturbance using sequential measurements in the iteration domain [252, 253].

In this work, the disturbance has been estimated in the *time-domain*: since the estimation is carried out offline, a batch estimation method is used to achieve high accuracy. In fact, batch state estimation methods (also known as smoothers, since they are typically used to *smooth* out the effects of measurement noise) have the advantage of providing state estimates with a smaller error covariance than sequential ones [258]. Basically, smoothers are used to estimate the state at time t, using measurements obtained both before and after t. To accomplish this task, two filters are usually used: a forward-time filter and a backward-time filter. The first practical smoothing algorithms are attributed to Bryson and Frazier [259], as well as Rauch, Tung, and Striebel (RTS) [260].

In particular, the RTS smoothing algorithm has maintained its popularity since the initial paper, and is likely the most widely used algorithm for smoothing to date. In fact, RTS is one of the most convenient and efficient forms of the fixed-interval smoother, because it combines the backward filter and smoother into one single backward recursion. A formal proof of the stability of the RTS smoother has been provided using a Lyapunov stability analysis in [254]. A comparison of the computational requirements in the various smoother implementations is given in [261].

RTS Fixed-Interval Smoother

We consider a generic discrete-time system with the state space form

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{9.1}$$

$$y(k) = C(k)x(k) + v(k) \tag{9.2}$$

where $w(k)$ and $v(k)$ are two random vectors described as Gaussian noises with zero mean values and given variance matrices: $w(k) \sim \mathcal{N}(0,Q)$, $k(k) \sim \mathcal{N}(0,R)$. The optimal smoother is given by a combination of the estimates of two filters: one, denoted by $\hat{x}_f(k)$, is given from a filter that runs from the beginning of the data interval to time $t(k)$, and the other, denoted by $\hat{x}(k)$, that works backward from the end of the time interval. The implementation of a generic RTS smoother is given in Algorithm 2.

Remark 50. *The RTS smoother is the most widely used algorithm in practice for the following reasons [254]:*

- *The forward filter covariance updated $P_f^+(k)$ and propagated $P_f^-(k)$, as well as the state matrix $A(k)$, do not need to be stored to determine the RTS smoothed estimate $\hat{x}(k)$. This is due to the fact that the gain $\mathcal{K}(k)$ can be computed during the forward filter process and stored to be used in the smoother estimate computation.*
- *The smoother state $\hat{x}(k)$ does not involve the smoother covariance $P(k)$. Therefore, the smoother covariance should be computed only if it is required for analysis purposes.*

Algorithm 2 State estimator: Discrete-Time RTS Smoother

```
Forward Filter Initialization:
    State: $\hat{x}_f(0) = \hat{x}_{f0}$
    Covariance: $P_f(0) = E\left\{x_f(0)x_f^T(0)\right\}$

for each $k \in 1, 2, \dots, N$ do

    Kalman Gain Computation:
    $K_f(k) = P_f^-(k)C(k)^T\left[C(k)P_f^-(k)H^T(k) + R(k)\right]^{-1}$

    Forward State Update:
    $\hat{x}_f^+(k) = \hat{x}_f^-(k) + K_f(k)\left[y(k) - C(k)\hat{x}_f^-(k)\right]$

    Forward Covariance Update:
    $P_f^+(k) = \left[I - K_f(k)C(k)\right]P_f^-(k)$

    Forward State Propagation:
    $\hat{x}_f^-(k+1) = A(k)\hat{x}_f^+(k) + B(k)u(k)$

    Forward Covariance Propagation:
    $P_f^-(k+1) = A(k)P_f^+(k)A(k)^T + Q(k)$

end for

Smoother Initialization:
    State: $\hat{x}(N) = \hat{x}_f^+(N)$
    Covariance: $P(N) = P_f^+(N)$

for each $k \in N, N-1, \dots, 1$ do

    Smoother Gain:
    $\mathscr{K}(k) = P_f^+(k)A(k)^T\left(P_f^-(k+1)\right)^{-1}$
    Smoother Covariance:
    $P(k) = P_f^+(k) - \mathscr{K}(k)\left[P_f^-(k+1) - P(k+1)\right]\mathscr{K}^T(k)$
    Smoother Estimate:
    $\hat{x}(k) = \hat{x}_f^+(k) + \mathscr{K}(k)\left[\hat{x}(k+1) - \hat{x}_f^-(k+1)\right]$

end for
```

Disturbance estimation

The disturbance estimate $\hat{d}$ is obtained at each iteration j by applying the RTS smoother to the discrete-time stochastic model[2]

$$d(k+1) = d(k) + \omega(k) \tag{9.3}$$

with the measurement model

$$y(k) = F(\mathrm{q})u_j(k) + d(k) + \mu(k) \ , \tag{9.4}$$

where ω and μ are two random vectors described as Gaussian noises with zero mean values and given variance matrices: $\omega(k) \sim \mathcal{N}(0,Q)$, $\mu(k) \sim \mathcal{N}(0,R)$. In the proposed model, the disturbance d represents the state of the dynamical system, while the error e defines the output measurement. The implementation of the RTS smoother specialized for this problem is shown in Algorithm 3.

Assuming the disturbance remains constant among the different trials ($\hat{D}_{j+1} = \hat{D}_j$), the next iteration predicted error E_{j+1} can be written as a function of the disturbance estimate $\hat{D}_j$ and the next input U_{j+1}. Namely, the next iteration predicted error in lifted form is defined as

$$\hat{E}_{j+1}(U_{j+1}) = \mathbf{F}\, U_{j+1} - Y_d + \hat{D}_j \ . \tag{9.5}$$

Remark 51. *ILC in combination with the estimation of the control quantity has been extensively analysed in the literature. In [262], it has been shown that the performance of an industrial robot increases significantly when an estimate of the control quantity is used instead of measurements of a related quantity. Performance and convergence of the ILC algorithm when combined with an estimator have been addressed in [263, 264].*

Remark 52. *The use of a time-domain filter in ILC can be effective at reducing the variance of the output error resulting from random process and measurement noise [254]. However, an important issue is the influence of having an imperfect model used to design the filter [255]. This error could produce deterministic non-zero steady state errors. On the other hand, if the model is accurate, then the filter is optimal and will outperform a simpler model-free method. As a consequence, an accurate model is required to exploit this design approach.*

[2] If a more accurate model of the disturbance is available, it can be incorporated into this algorithm.

Algorithm 3 Disturbance estimator: Discrete-Time RTS Smoother

```
Forward Filter Initialization:
  Disturbance: $\hat{d}_f(0) = \hat{d}_{f0}$
  Covariance: $P_f(0) = E\left\{\hat{d}_f(0)\,\hat{d}_f^T(0)\right\}$

for each $k \in 1, 2, \ldots, N$ do

    Kalman Gain Computation:
    $K(k) = P_f^-(k)\left[P_f^-(k) + R\right]^{-1}$

    Forward State Propagation:
    $\hat{d}_f(k+1) = \hat{d}_f(k) + K(k)\left[y(k) - \hat{d}_f(k) - F(\mathrm{q})u_j(k)\right]$

    Forward Covariance Update:
    $P_f^+(k) = [I - K(k)]\,P_f^-(k)$

    Forward Covariance Propagation:
    $P_f^-(k+1) = P_f^+(k) + Q$

end for

Smoother Initialization:
  Disturbance: $\hat{d}(N) = \hat{d}_f(N)$

for each $k \in N-1, N-2, \ldots, 1$ do

    Smoother Gain:
    $\mathscr{K}(k) = P_f^+(k)\left(P_f^-(k+1)\right)^{-1}$

    Smoother Estimate:
    $\hat{d}(k) = \hat{d}_f(k) + \mathscr{K}(k)\left[\hat{d}(k+1) - \hat{d}_f(k+1)\right]$

end for
```

9.2.3 Input Update: Quadratically-Optimal Design

The learning algorithm is completed by the input update step using a Q-ILC design (see Section 8.5.4). Specifically, the next input U_{j+1} is obtained in the lifted system minimizing the quadratic cost criterion:

$$J_{j+1}(U_{j+1}) = \hat{E}_{j+1}^T W_e \hat{E}_{j+1} + \Delta U_{j+1}^T W_{\Delta u} \Delta U_{j+1} \qquad (9.6)$$

where no weight on the control is considered (W_u is a null matrix). In this application, we impose constraints on the input acceleration since it is related to the physical capabilities of the system actuators. These constraints are expressed through the following mathematical inequality:

$$\ddot{U}_{low} \leq \ddot{U}_{j+1} \leq \ddot{U}_{high} \ , \qquad (9.7)$$

where $\ddot{U}_{j+1}$ is the centered difference approximation of the second derivative of the input reference U_{j+1}. These constraint can be rearranged as linear inequality with respect to U_{j+1} as:

$$\ddot{U}_{j+1} = Z_u \, U_{j+1} \ , \qquad (9.8)$$

where

$$Z_u = \frac{1}{t_{ILC}^2} \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix} \qquad (9.9)$$

with t_{ILC} being the ILC sampling time. The asymptotic minimization over iterations of the tracking error under the proposed constrained optimization-based ILC can be guaranteed with the following assumptions [236].

Assumption 5. *The matrix **F** has full row-rank. If **F** does not have full row-rank, a projection operator must be introduced to prove convergence of the controllable part of the system [236].*

Assumption 6. *Given the input constraints* (9.8), *reference trajectory Y_d, and the actual steady-state disturbance D, the zeroing of the error is possible with an input in the feasible set [265].*

9.3 Simulation results

In this Section, we compare in simulation the proposed approach with the Kalman-filter-based ILC (K-ILC) presented in [252]. We apply these algorithms on a 1D mass-spring-damper system using the nominal model:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\,\omega_n s + \omega_n^2} \tag{9.10}$$

with natural frequency $\omega_n = 1.8\ rad/s$ and damping ratio $\xi = 0.8$. The algorithms performance is evaluated in a realistic scenario in which the system is affected by parametric uncertainties. Further, as in [252], we add a disturbance $d(t) = 0.5\sin^2(t)$ and a Gaussian noise $\mathcal{N} \sim (0, 0.01)$ to the output signal $y(t)$. The learning parameters chosen are listed in Table 9.1 and the desired trajectory is $y_d(t) = \sin(0.5\pi t)$.

Table 9.1: *Parameters used in the simulations.*

	W_u	$W_{\Delta u}$	W_e	$P(0)$	R	Q
Proposed	0_N	0.01 I_N	I_N	0.1	0.01	0.1
K-ILC	0_N	0.01 I_N	I_N	0.1 I_N	0.01 I_N	0.1 I_N

Firstly, a Monte Carlo study has been carried out with respect to uncertainty (assuming standard deviation equal to 5% of the nominal values) on the natural frequency and on the damping ratio. The learning process (30 iterations of ILC algorithm with a sampling time $t_{ILC} = 0.01\ s$) has been performed simulating different plants (100 dynamic parameters samples), maintaining the same model in the learning algorithm. For each plant the results have been averaged repeating the entire process 10 times to reduce the noise influence. The most relevant statistics (mean and standard deviation) of the error 2-norm ($||E_j||_2$) at the first iteration and at steady-state[3] are collected in Table 10.2. To visualize the algorithms rate of convergence, the error evolution in the iteration domain of a learning cycle[4] is shown in Figure 9.3.

[3] The steady-state error norm was approximated by the average error norms of iterations 20 to 30.

[4] The error evolution for most of the learning cycles has a behaviour similar to the one in Figure 9.3.

Table 9.2: *Error 2-norm with 5% level of uncertainty.*

Algorithm	First Iteration	Steady-State
Proposed	0.84 ± 0.10	0.23 ± 0.01
K-ILC	1.46 ± 0.21	0.23 ± 0.01

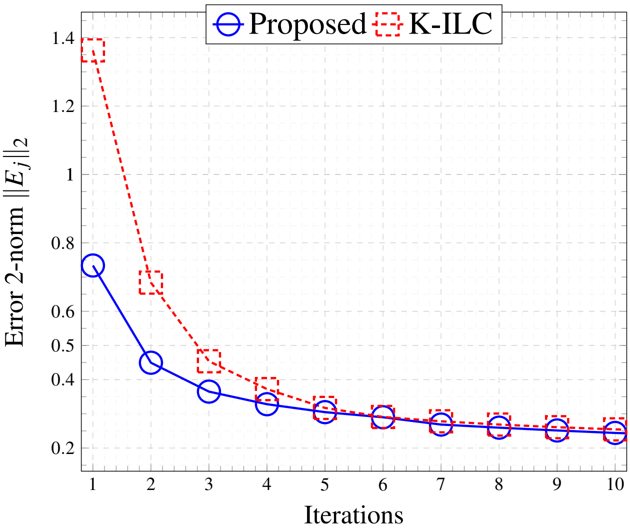


Figure 9.3: *Error 2-norm evolution in the iteration domain (first 10 iterations).*

We can see that the proposed approach achieves a faster convergence with respect to K-ILC maintaining similar steady-state performance. After that, 3 different Monte Carlo studies have been carried out with respect to larger uncertainties (standard deviation equal respectively to 10%, 20%, 40% of the nominal values) on the dynamic parameters to assess the robustness of the proposed approach. The results are summarized in Table 9.3. We can notice that the proposed approach is capable to improve the performance also with a considerable error in the parameters (20% standard deviation). However, as expected from Remark 52, when the model is very inaccurate (40% standard deviation), K-ILC outperforms our approach at steady-state due to the deterministic error caused by an imperfect knowledge of the model.

Table 9.3: *Error 2-norm with different levels of uncertainty.*

Algorithm	Uncertainty level	First Iteration	Steady-State
Proposed	±10%	1.30 ± 0.32	0.23 ± 0.01
	±20%	2.39 ± 0.65	0.23 ± 0.01
	±40%	4.33 ± 2.53	0.38 ± 0.25
K-ILC	±10%	1.99 ± 0.42	0.23 ± 0.01
	±20%	2.74 ± 0.66	0.23 ± 0.01
	±40%	4.10 ± 2.16	0.28 ± 0.18

9.4 Experimental results

In this Section, we apply the proposed ILC algorithm and the K-ILC to a multirotor UAV to achieve high performance tracking. Firstly we specialized the algorithm for this task and then we present the results obtained in the experiments.

9.4.1 Applying ILC to UAV trajectory tracking

The ILC scheme of Section 9.2 can be applied to any dynamic systems with underlying feedback loops. Specifically, it is applied to improve the tracking accuracy in executing a complex manoeuver with an UAV guided by a commercial controller (PX4 autopilot [55]). The overall architecture is schematized in Figure 9.4.

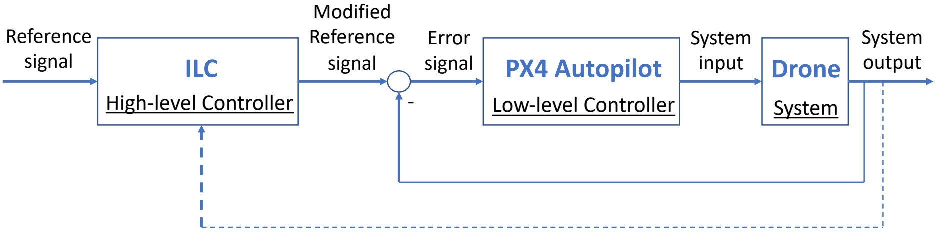


Figure 9.4: *Block diagram of the proposed architecture.*

Remark 53. *As pointed out in Section 8.3, the ILC scheme requires the disturbance to be iteration-invariant (or at least slowly iteration-varying).*

In this application, this can be reasonably assumed true, because the ILC objective is to achieve a high level of performance eliminating the unmodelled (possibly nonlinear) dynamics (e.g., aerodynamic effects). Other iteration-varying disturbances (e.g., wind) have to be counteracted by the feedback controller. Furthermore, the feedback loop allows us to use a linear model as a good approximation of the UAV closed-loop dynamics.

In this work we consider without loss of generality the case of in-plane trajectory ($y_d = [x_d,\ y_d]^T$). This is done by acting on the set-points ($u = [x_{sp},\ y_{sp}]^T$) commanded to the stock controller exploiting measurements of the position ($y = [x_m,\ y_m]^T$) to estimate the disturbance.

To analyse the learning performance, the following metrics are defined. A synthetic indicator at a specific iteration j is the *average position error* along the trajectory:

$$e_{pos,j} = \frac{1}{N} \sum_{k=1}^{N} \sqrt{(x_{m,j}(k) - x_d(k))^2 + (y_{m,j}(k) - y_d(k))^2}\,. \tag{9.11}$$

This indicator can be adimentionalized with respect the average position error at the initial iteration $\frac{e_{pos,j}}{e_{pos,0}}$. Additionally the performance indicator $\frac{e_{pos,j} - e_{pos,j-1}}{e_{pos,j}}$ has been used to highlight the difference between two consecutive iterations.

9.4.2 Experimental setup

Flight tests are carried out inside the Flying Arena for Rotorcraft Technologies (FlyART) of Politecnico di Milano (see Appendix A for more details) which is an indoor facility equipped with a Motion Capture system (Mo-Cap). The drone is a fixed-pitch quadrotor designed by ANT-X [266], with a maximum take-off weight below 300g (see Figure 9.5). A ground control station receives measurements from the Mo-Cap system, reconstructs the state of the drone and, then, computes the next iteration set-points according to the proposed approach. The overall strategy has been integrated in the PX4 autopilot [55] using the ANT-X rapid prototyping system for multirotor control.

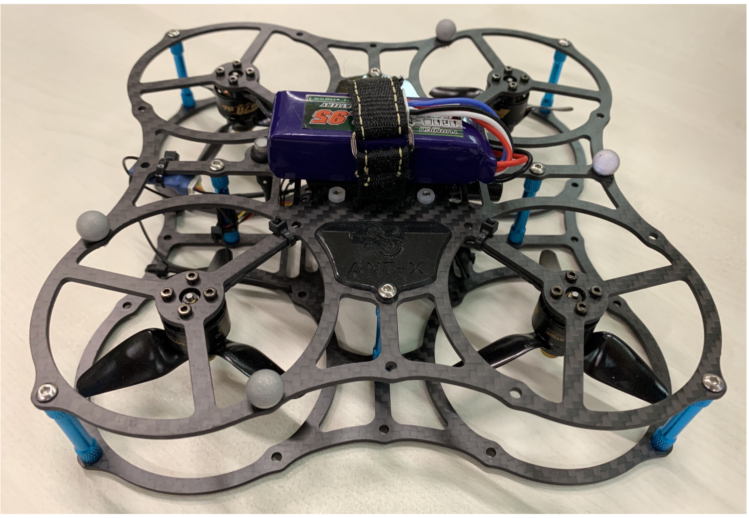


Figure 9.5: *ANT-X drone.*

9.4.3 Model Identification

The system dynamics (*i.e.*, the complementary sensitivity of the UAV) was identified as a black-box model, by applying the Predictor-Based Subspace Identification algorithm (PBSID, see also [267], [268], [269] for applications of PBSID to rotorcraft dynamics) to input-output data gathered in dedicated identification experiments.The model for the transfer function from x_{sp} to x_m is[5]

$$G_x(s) = e^{-0.25s} \cdot \frac{0.08s + 1.94}{s^2 + 1.752s + 2.01}. \tag{9.12}$$

The model was validated against flight data collected in another experiment. Figure 9.6 shows the measured response to the reference signal against the simulated response obtained with the identified model, showing a close match to the measured data.

[5] The PBSID algorithm returns a discrete time state space model, but the continuous time transfer function obtained with the Tustin approximation has been reported to give more physical insight. In fact, in this formulation we can clearly notice that the system is characterized by a natural frequency $\omega_n \approx 1.4$ rad/s and damping ratio $\xi \approx 0.6$.

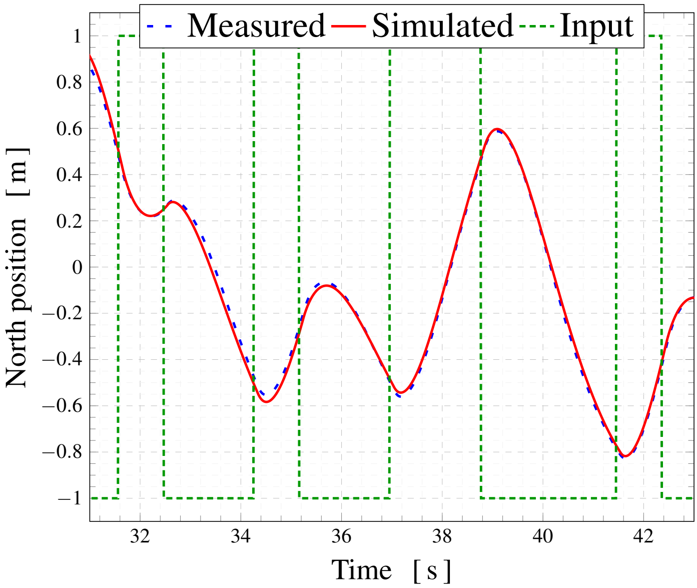


Figure 9.6: *Validation of the identified model.*

Note that due to symmetry, and based on previous experience, we can use the model identified for longitudinal dynamics (x-direction) also for the lateral dynamics (y-direction), *i.e.*, assume that $(G_x(s) = G_y(s))$. These models are used in the estimation step (time-domain) and in the input update step (iteration-domain) constructing **F** following the approach described in Section 8.4.3.

9.4.4 Results

In this Section, we present the results obtained by applying the proposed algorithm and the K-ILC on the ANT-X UAV with the same learning parameters used in Section 9.3 and with a sampling time $t_{ILC} = 0.05\ s$. In the experiment, the manoeuvre to be learned is an aggressive eight-shape trajectory flown in the horizontal plane (see Figure 9.7) characterized by a maximum velocity $v_{max} = 2.1\,\mathrm{m/s}$ and maximum acceleration

$a_{max} = 4\,\mathrm{m/s^2}$. Namely, the trajectory is defined as:

$$\begin{cases} x_d = \frac{1}{2} \cdot \sin\left(\pi(t - t_i)\right) \\ y_d = \cos\left(\frac{\pi}{2}(t - t_i)\right) \\ z_d = -2, \end{cases} \tag{9.13}$$

where x_d, y_d and z_d are the components of the desired trajectory, t is the time variable, and t_i is the time in which the eight-shape trajectory starts.

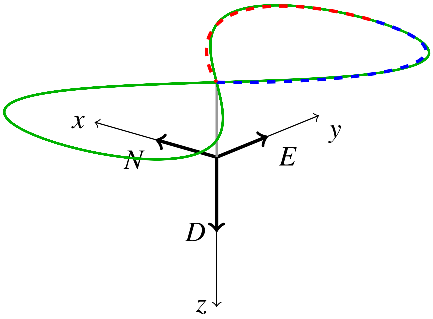


Figure 9.7: *Eight-shape trajectory.*

The quadrotor is required to hover at the beginning of the eight-shape trajectory: the learning motion starts and ends in the same hovering point. The acceleration (blue dashed line) and deceleration (red dashed line) phases must also be learned precisely at the beginning and the end of the eight-figure.

Remark 54. *ILC requires identical initial conditions at each iteration (see Assumption 4). However, obtaining identical initial conditions in real experiments is very problematic. In this work, we start from the hover position and let the UAV stabilize around the initial trajectory position until an in-plane position error of less than* 5 cm *is guaranteed.*

The data comparison between the two methods is reported in Table 9.4. We can notice that, as in Section 9.3, the proposed approach achieves faster convergence with respect to K-ILC. Due to space limitations only the trajectories obtained by to the proposed approach are plotted in the

Table 9.4: *Learning performance results for 8-shape trajectory.*

Iteration	e_{pos_j} [m]	$\frac{e_{pos_j}}{e_{pos_0}} \cdot 100$ [%]	$\frac{\lvert e_{pos_j} - e_{pos_{j-1}} \rvert}{\lvert e_{pos_{j-1}} \rvert} \cdot 100$ [%]
$j = 0$	0.422	100.0	-
		Proposed approach	
$j = 1$	0.040	9.59	90.41
$j = 2$	0.018	4.29	55.33
		K-ILC	
$j = 1$	0.072	17.2	82.84
$j = 2$	0.050	11.9	30.52
$j = 3$	0.044	10.5	12.13

following figures. Specifically, in Figure 9.8 the evolution in *xy*-plane is depicted without the acceleration and deceleration phases, while in Figure 9.9 the time evolution of the North position is plotted (the East direction shares similar behaviour). We can state that the proposed approach is very effective in improving the tracking performance of the UAV, while being robust to small initial errors in the positioning, unmodelled system dynamics, process and measurement noises.

9.5 Final considerations

In this Chapter, we tackled the problem of high-performance tracking of UAVs for the solution of which an ILC-based control approach has been developed. This approach implements an efficient learning algorithm that uses the dynamics model of the system and the experience gained through repeating a given trajectory to improve tracking performance. Specifically, the proposed ILC algorithm modifies the reference signal offline after each trajectory repetition. However, this approach requires an accurate model of the system dynamics and cannot effectively handle non-repetitive disturbances. To overcome these limitations, in the following Chapter, we combine the proposed algorithm with an adaptive controller that is able to force the system to behave close to a *reference model* despite the presence of iteration-varying disturbances and unmodelled dynamics.

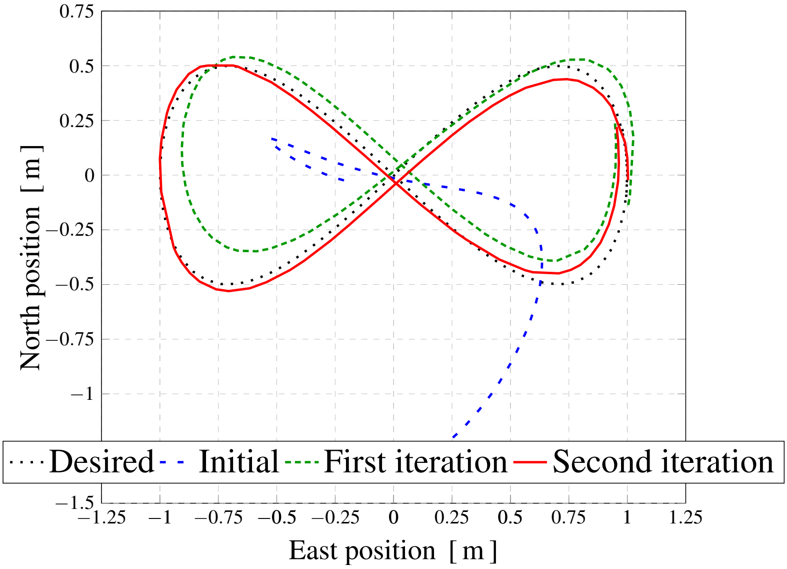


Figure 9.8: *8-shape trajectories for the iterations* 0, 1 *and* 2.

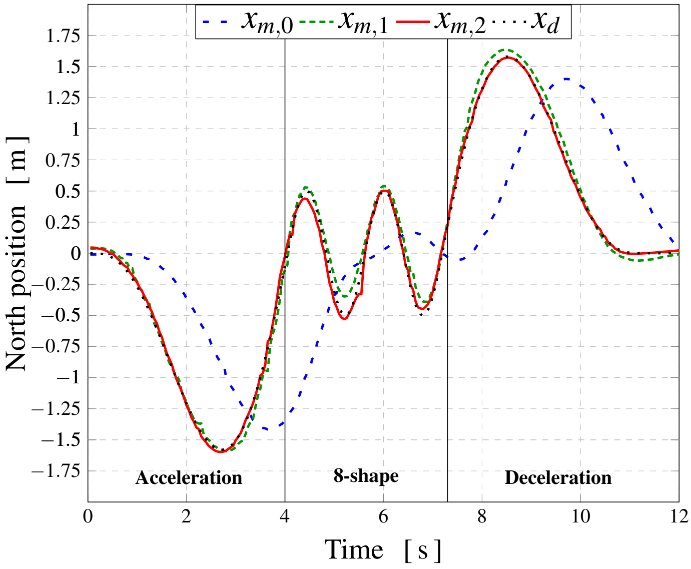


Figure 9.9: *UAV North-position for iterations* 0, 1 *and* 2.

CHAPTER 10

H_∞-based Transfer Learning

The ILC approach presented in Chapter 9 has been capable of reaching exceptional tracking performance. However, the learning phase needed to apply such technique is related to each specific system, thus making the application of ILC poorly scalable. To overcome this limitation, we present in this Chapter a novel H_∞-optimisation-based definition of a map that allows transforming the input signal learnt on a *source* system to the input signal needed for a *target* system to execute the same task.

10.1 Introduction

The downside of ILC is that, in order to actually learn the task, a time-consuming training phase is needed, which is specific for each system and for each task. As a consequence, whenever a new system and/or task is considered, the whole process of learning has to restart from scratch.

To widen the field of application of ILC to a large number of systems operating in different environments like UAVs, there is the need for a systematic approach that allows transferring knowledge acquired in previous training phases to new systems and/or tasks. This problem goes under the name of Transfer Learning (short, TL) in the literature. For details, the readers are referred to the survey papers [270], [271] and references therein. This work focuses on multi-system TL, whose objective is to transfer the knowledge acquired on a *source* system to a *target* system that needs to perform the same task. This transfer of data may be beneficial if the *source* system is less costly, difficult, or hazardous to operate than the *target* system. Furthermore, if multiple, similar robots have to perform the same task, it is more cost effective if one robot learns to perform the task and transfers its knowledge to the other robots [272].

The application of TL to ILC is a recent research topic. In [273], Hamer *et al.* designed a TL approach based on the assumption that a linear mapping around the desired trajectories can approximate the learning process of ILC. In [265], Pereida *et al.* developed a multi-robot multi-task TL approach by using a dynamic transfer map in combination with an adaptive controller. Another aspect that has been analysed is the nature of the transfer map: the use of a dynamic map rather than a scalar one appears to be the most reasonable choice, and, in fact, in [274], Sorocky *et al.* proved that the optimal TL map between two robots is, in general, a dynamic system. However, the effects of such maps on the transfer error have not been analytically investigated yet. On the other hand, scalar transformations have already been analysed, like in [275], in which Raimalwala *et al.* derived an analytical upper bound on the transformation error. In [276], Helwa *et al.* proposed a novel approach to build a dynamical TL map based on a grey-box model identification procedure: the structure of the map is derived from knowledge of the order and relative degree of the *source* and *target* systems, and then the map is identified through simple experiments performed on both the *source* and *target* system. Nevertheless, this procedure has an important drawback: experiments are needed to identify the map. In this way, the time which would be saved for the training of the *target* system is not actually saved.

10.2 H_∞-based transfer learning

In this Section, we introduce and formalize the proposed TL approach. The problem of multi-system TL in the framework of ILC consists in finding a dynamical transfer map which transforms the input learnt on a *source* system (*i.e.*, which allows the source system to track the desired trajectory) into the input needed for a *target* system to track the same trajectory. The expected result of this approach is a reduction of the first iteration error when the *target* system performs the same task. Specifically, a novel H_∞-optimisation-based TL architecture is derived from a classical control problem, namely the *model matching* problem. While we will specifically refer to multirotor platforms for the experimental validation, the formulation is general and can be applied to any dynamic system (including systems with underlying feedback loops).

10.2.1 Model matching

Considering the block diagram of Figure 10.1, the *model matching* goal is to design $C(s)$ to minimise the error e when the *source* and *target* systems, namely $G_1(s)$ and $G_2(s)$, are excited by the same reference signal r. The matching error is defined as:

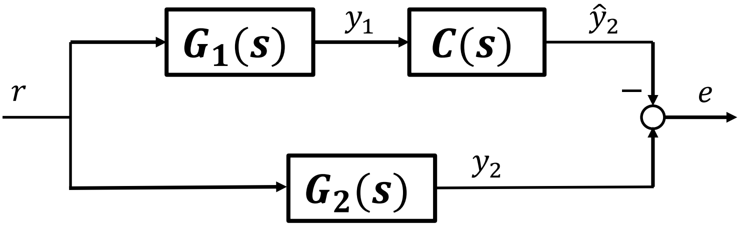


Figure 10.1: *Model matching problem.*

$$\begin{aligned} e(s) &= y_2(s) - \hat{y}_2(s) = y_2(s) - C(s)y_1(s) \\ &= (G_2(s) - C(s)G_1(s))r(s) \ . \end{aligned} \tag{10.1}$$

The *model matching* problem can be treated in the H_∞ framework [277] as an optimization problem that minimize a measure of the deviation between the two models by exploiting invariants of the system, *i.e.*, the transfer functions. Specifically, the problem can be formulated in terms of H_∞ norm of the difference between the two models:

$$\min_{M(s)} ||G_2(s) - C(s)G_1(s)||_\infty \,. \tag{10.2}$$

The non-smooth optimization of the H_∞ norm has been studied extensively in the last decade and for which reliable computational tools are presently available [278]; these tools are embedded in the *Robust Control Toolbox* of MATLAB, making use of the `systune` function. The method implemented in these tools allows the H_∞ design of a controller; since there are infinitely possible architectures, the method considers a *standard form*, in which any block diagram can be rearranged. Considering the performance inputs w (*e.g.*, reference signals, disturbances and noise) and outputs z (*e.g.*, tracking errors, control actions, plant outputs subject to disturbances), the *standard form* is depicted in Figure 10.2 and consists of two main components:

- LTI model $P(s)$ which combines all fixed (non tunable) blocks in the control system;
- Controller $K(s)$ which combines all tunable control elements.

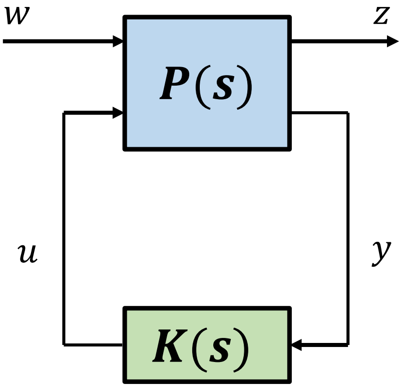


Figure 10.2: *Standard form for H_∞ synthesis.*

By considering the closed-loop transfer function $T_{w\rightarrow z}(P,K)$, the optimization problem is formulated as follows:

$$\min_{K(j\omega)} \max_{\omega\in[0,\infty]} \bar{\sigma}\left(T_{w\to z}(P(j\omega),K(j\omega))\right) \tag{10.3}$$
$$\text{subject to: } K(s) \text{ stabilizes } P(s) \text{ internally}$$

where $\bar{\sigma}(\cdot)$ denotes the maximum singular value associated with its argument: in the SISO case, this norm is just the peak gain over frequency, while it measures the peak 2-norm of the frequency response over frequency in the MIMO case. The problem (10.3) is solved iteratively starting from initial guess, by constructing a tangent model around the current iterate that approximates locally the original problem: an adequate descent direction is then computed by solving a convex quadratic program; further details on the state-of-the-art optimizer can be found in [277]. Other optimization methods based on convex formulation often rely on conservative and expensive relaxations or resort to biconvex schemes [278].

It is worth noticing that the *model matching* problem formulated in equation (10.2) is a special case of the one typically used for control system tuning (10.3), but without the stability constraint, which is meaningless for model matching [277]. In fact, it is very simple to obtain an equivalent representation of the *model matching* problem in the *standard form* (see Figure 10.3) by dividing the plant from the tunable element.

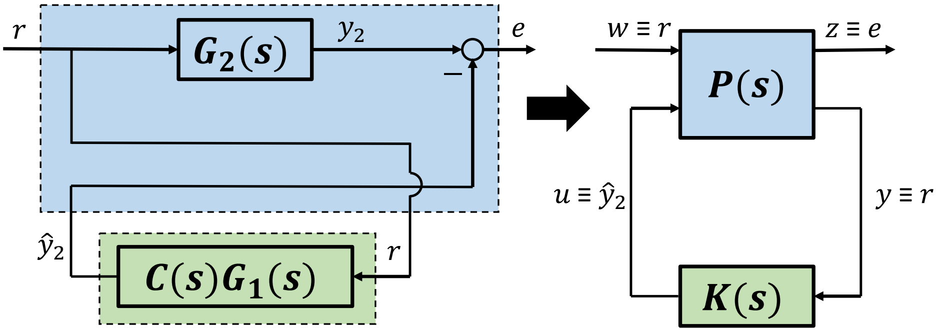


Figure 10.3: *Standard form of the model matching problem.*

Specifically, the performance input is the reference signal while the performance output is the matching error, the measured output is substituted by the reference signal, while the control input is $\hat{y}_2$, which should be as close as possible to the output of the *target* model $G_2(s)$. Finally, the augmented plant and the tunable controller are respectively:

$$P(s) = \begin{bmatrix} G_2(s) & -1 \\ 1 & 0 \end{bmatrix} \quad , \quad K(s) = C(s)G_1(s) \ . \tag{10.4}$$

10.2.2 Multi-system transfer learning

Multi-system TL is an analogous problem, but, instead of transforming the output, the goal is to transform the input that excites the target system, as shown in Figure 10.4. The map $M(s)$ has to be tuned to minimise the error between the outputs of the two systems when the *source* one is fed by the learnt input u_1^L and the *target* one by the transferred input u_2, where $u_2(s) = M(s)u_1^L(s)$.

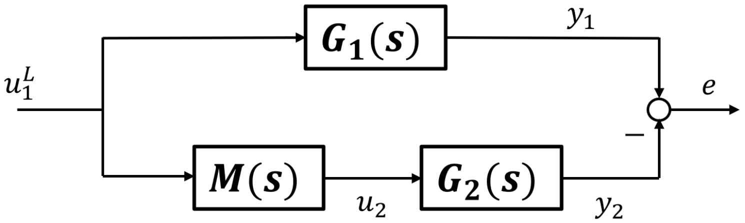


Figure 10.4: *Transfer learning problem.*

By looking at the structure of the block diagram in Figure 10.4 and assuming that $G_2(s)$ is a non-minimum phase system, a trivial solution to solve the matching problem is:

$$M_{basic}(s) = G_2^{-1}(s)G_1(s) \ . \tag{10.5}$$

This choice allows to ideally “cancel” the *target* dynamics and “substitute” it with the *source* dynamics, thus producing a *target* output identical to the *source* one. However this choice would work only in an ideal situation in which the two transfer functions are not affected by uncertainty. In other words, this non-robust approach would not be exploitable in a real-world application. For this reason an optimisation-based approach like the H_∞ one has to be preferred.

Similar to the previous Section, the block diagram of Figure 10.4 can be manipulated in the *standard form* (see Figure 10.5) considering:

$$P(s) = \begin{bmatrix} G_1(s) & -1 \\ 1 & 0 \end{bmatrix} \quad , \quad K(s) = G_2(s)M(s) \ . \tag{10.6}$$

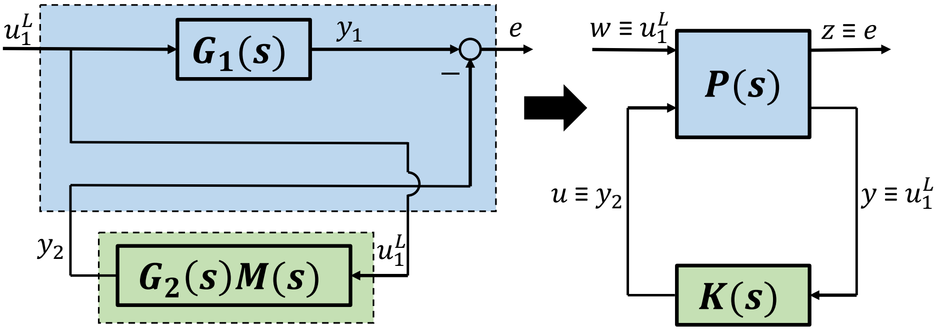


Figure 10.5: *Standard form of the transfer learning problem.*

Moreover, the plant $P(s)$ can be augmented using frequency weights, namely frequency-dependent functions used to enforce requirements in the optimisation problem. In the context of TL, a weighting function $W(s)$ should be defined to enforce a desired shape for the transfer function from the *source* learnt setpoint u_1^L to the error e. By using this weight, the objective of the optimisation problem can be written as

$$\min_{M(s)} ||W(s)(G_1(s) - G_2(s)M(s))||_\infty \ . \tag{10.7}$$

A powerful feature of the H_∞ framework is the possibility of including model uncertainty in the optimisation problem. The result will be a robust transfer map which allows to maintain the desired performance (described by the weighting function) in the worst-case scenario. Furthermore, the structure H_∞ synthesis has the possibility of enforcing a desired structure to the tunable elements [277]. In the TL problem a rough structure of $M(s)$ is usually known *a priori* based on the model structure of the *source* and *target* system, even if these models are uncertain.

10.2.3 Proposed TL architecture

The result of H_∞-optimization-based TL scheme is a dynamic transfer map which is treated as a standard dynamic model, receiving as input the set-point learnt on the *source* system (u_1^L) and producing as output the input needed for the *target* system (u_2) to reproduce the same task. In this work, the set-point u_1^L has been obtained performing N_S iterations of the *Smoother-based* ILC (S-ILC) algorithm presented in Chapter 9. After that, the learning is continued on the *target* system (additional N_S iterations of the S-ILC algorithm) using as set-point ($u_2(s) = M(s)u_1^L(s)$). The expected result of the proposed approach is a reduction of the first iteration error when the *target* system performs the same task.

The overall approach is summarised in Algorithm 4. Note that if TL is successful, then the learning phase on the *target* is not needed. However, few iterations can still allow to further improve the performance.

10.3 Simulation results

In this Section we validate in simulation the proposed approach on two 1D mass-spring-damper systems. Concerning the characteristics of the two systems sharing data, in [274] the authors analytically demonstrate that TL is more likely to guarantee positive results if it is performed from a more aggressive system to a less aggressive one. If this concept is considered in the framework of UAVs, this means that it is better to choose a smaller (and faster) drone as source system and a bigger (and slower) drone as the target one. This choice is beneficial in the aeronautical framework, because tests performed on smaller vehicles are easier and safer to perform.

The two (nominal) systems are therefore selected with this rationale:

$$G_1(s) = \frac{\omega_S^2}{s^2 + 2\xi_S\omega_S s + \omega_S^2} \ , \ \omega_S = 4 \ , \ \xi_S = 0.7 \tag{10.8}$$

$$G_2(s) = \frac{\omega_T^2}{s^2 + 2\xi_T\omega_T s + \omega_T^2} \ , \ \omega_T = 1 \ , \ \xi_T = 0.8 \ . \tag{10.9}$$

The TL performance is evaluated in a realistic scenario in which the *target* system is affected by parametric uncertainties. Furthermore, we add

Algorithm 4 H_∞ Transfer Learning Approach

```
Input: $G_1(s)$ and $G_2(s)$ with the related uncertainty levels.
Dynamical transfer map
$H_\infty$ optimisation $\Rightarrow M(s)$

Source system
for $j = 1 : N_S$ do

    Apply $U_j$
    Measure $Y_j$
    S-ILC: Compute optimal $U_{j+1}$ using $\hat{D}_j$

end for

Result: Optimal input $u_1^L = U_{j+1}$

Transfer learning
Transferred setpoint: $u_2(s) = M(s)u_1^L(s)$

Target system
Apply $U_{1,T} = u_2$
Measure $Y_{1,T}$

for $j = 2 : N_T$ do

    S-ILC: Compute optimal $U_{j,T}$ using $\hat{D}_{j,T}$
    Apply $U_{j,T}$
    Measure $Y_{j,T}$

end for
```

a disturbance $D(t) = 0.1\sin(2t)$ and a Gaussian noise $\mathcal{N} \sim (0, 0.01)$ to the output signals of both *source* $Y_s(t)$ and *target* system $Y_t(t)$. The desired trajectory is $Y_d(t) = 0.5\sin(t)$ and the S-ILC parameters (which are referred to the nomenclature of Chapter 9) are listed in Table 10.1.

Table 10.1: *Parameters used in the simulations.*

	W_u	$W_{\Delta u}$	W_e	$P(0)$	R	Q
Proposed	0_N	$0.1\ \mathrm{I}_N$	I_N	10	0.01	0.1

A Monte Carlo study has been carried out with respect to uncertainty (assuming standard deviation equal to 20% of the nominal values) on the natural frequency and on the damping ratio of the *target* model (see Figure 10.6 for the Bode plot of the resulting uncertain model).

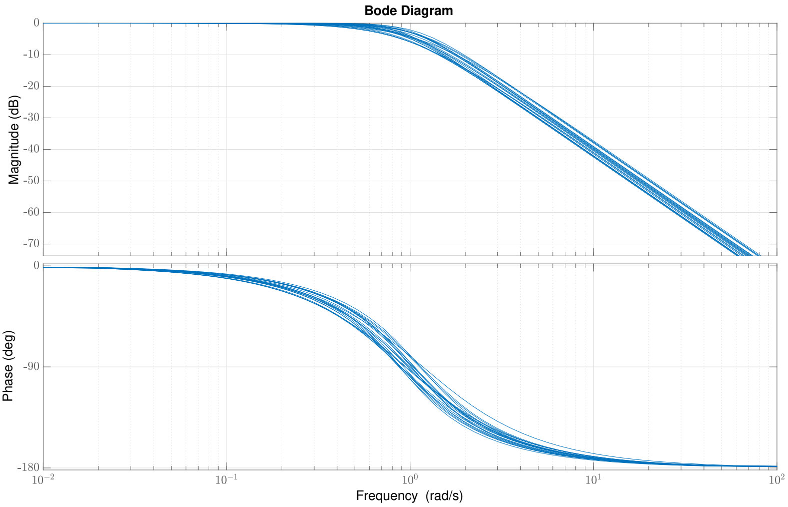


Figure 10.6: *Bode plot of the uncertain target model.*

By using these uncertainty levels, the structured H_∞ model matching is solved by using MATLAB function `systune`. The frequency weight $W(s)$ is defined using the function `makeweight`, which enables to create a first order, continuous-time weight with the following properties: $|W(0)| = 40$, $\lim_{\omega\to\infty}|W(j\omega)| = 0.99$ and the gain cross-over frequency equal to

2 rad/s. Given the models of the *source* and *target* systems, the structure assigned to the transfer map is:

$$M(s) = \frac{a_1 s^2 + a_2 s + 1}{b_1 s^2 + b_2 s + 1} , \tag{10.10}$$

where $a_1, a_2, b_1, b_2 \in \mathbb{R}$ are the optimization variables. This choice is justified by the previous considerations about the trivial solution of the matching problem: even if the optimal solution cannot be $M_{basic}(s) = G_1(s)G_2^{-1}(s)$, it is still reasonable to choose the transfer map such that $n_z = n_{z,s} + n_{p,t}$ and $n_p = n_{p,s} + n_{z,t}$, where n_z is the number of zeros and n_p is the number of poles. The optimization result is the transfer map:

$$M(s) = \frac{0.823s^2 + 1.464s + 1}{0.043s^2 + 0.272s + 1} . \tag{10.11}$$

The procedure followed in the simulation environment is explained in the following. Firstly, the learning process is applied on the *source* system involving 10 iterations of the S-ILC algorithm. Then, the transferred setpoint is computed using the learnt input u_1^L and the computed transfer map $M(s)$. Finally, the *target* system undergoes the learning process both with and without using TL. Namely, 10 iterations of S-ILC have been performed simulating different *target* systems (100 natural frequency and damping ratio samples), maintaining the same model in the learning algorithm. The most relevant statistics (mean and standard deviation) of the error 2-norm ($||E_j||_2$) at the first iteration on the *target* system are collected in Table 10.2. We can notice that an average reduction of 82% is obtained using the proposed TL approach despite the uncertainty.

Table 10.2: *Relevant statistics of the first iteration error 2-norm with/without TL.*

	TL	No TL
Error 2-norm	1.19 ± 0.48	6.65 ± 0.26

Furthermore, the evolution of the error mean of the *target* system in the iteration domain is shown in Figure 10.7. We can see that the proposed approach achieves a faster convergence with respect to the case without TL maintaining similar steady-state performance.

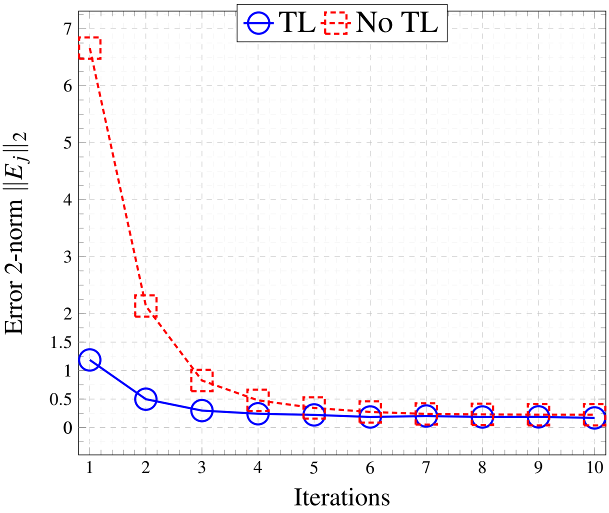


Figure 10.7: *Mean error 2-norm evolution in the iteration domain.*

10.4 Experimental results

In this Section, we apply the proposed TL approach to two different-scale multirotor UAVs to achieve high performance tracking. Firstly, we describe the controller architectures for this task and, then, we present the results obtained in the experiments exploiting the proposed TL algorithm.

10.4.1 UAV controller architectures

The controller architecture presented in Chapter 9 is applied to improve the tracking accuracy in executing a complex manoeuvre with two different-scale UAVs. Namely, the controller architecture implemented on the *source* UAV is a modification of the nonlinear cascaded controller for position stabilization implemented in the PX4 autopilot, where the commercial controller is augmented using the $\mathcal{L}_1$ adaptive control [279]. The overall architecture is schematized in Figure 10.8. Instead, the *target* UAV controller is the stock version of the commercial one.

Remark 55. *The combination of adaptive control and ILC is inspired* *from previous works [280, 281]. The adaptive controller drives dynamically different nonlinear systems to behave as a defined linear model over iterations, whereas ILC increases tracking performance over iterations.*

Specifically, the *source* system dynamics is a feedback loop driven by $\mathcal{L}_1$ adaptive control that follows the ILC reference setpoint U_j behaving as a desired linear model despite uncertainties and/or disturbances. $\mathcal{L}_1$ adaptive control is based on the MRAC architecture with the addition of a low-pass filter that decouples robustness from adaptation. This decoupling allows for an arbitrary increase of the estimation rate, limited only by the available hardware, while robustness is limited by the available control channel bandwidth and can be addressed by conventional methods from classical and robust control. The main feature of this configuration is the guaranteed transient performance and guaranteed robustness in the presence of fast adaptation [41]. As in the previous Chapter, we consider the case of in-plane trajectory ($y_d = [x_d,\ y_d]^T$). This is done by acting on the set-points ($u = [x_{sp},\ y_{sp}]^T$) commanded to the controller exploiting measurements of the position ($y = [x_m,\ y_m]^T$) to estimate the repetitive disturbance. To analyse the learning performance, the metrics defined in Section 9.4.1 are used.

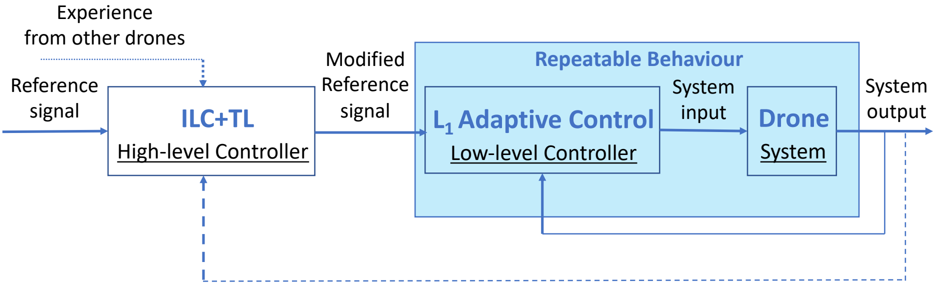


Figure 10.8: *Block diagram of the proposed TL architecture.*

10.4.2 Experimental setup

Flight tests are carried out inside the Flying Arena for Rotorcraft Technologies of Politecnico di Milano (see Appendix A). Two different-scale quadrotors (shown in Figure 10.9) are used. Namely, the *source* system is ANT-X, a small-scale UAV with in-plane dimensions $20 \times 20\,\text{cm}$ for a takeoff weight of $0.250\,\text{kg}$, and the *target* system is ADAM-0, with in-plane dimensions $40 \times 40\,\text{cm}$ for a takeoff weight of $1.4\,\text{kg}$. The overall strategy has been integrated in the PX4 autopilot [55] using the ANT-X [266] rapid prototyping system for multirotor control.

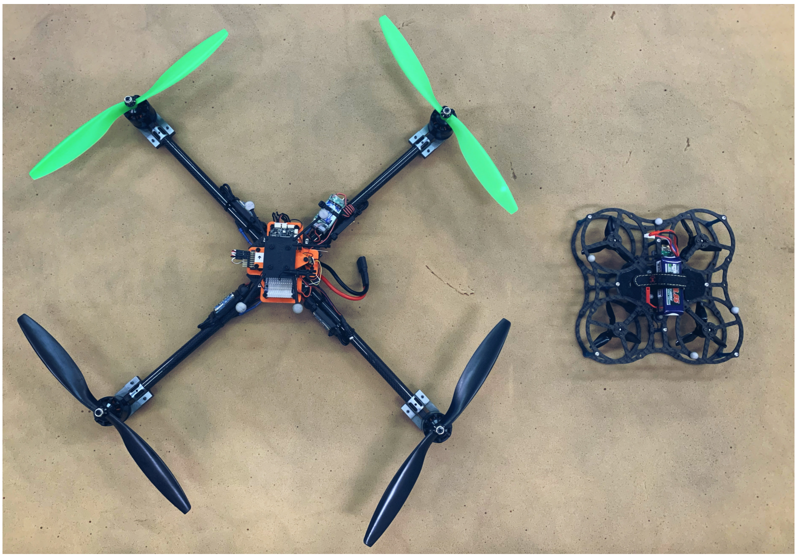


Figure 10.9: *The two different quadrotors used in the experiments.*

10.4.3 Model Identification

The complementary sensitivity of the UAVs (*i.e.*, the system dynamics) were identified as a black-box model, by applying the PBSID algorithm to input-output data gathered in dedicated identification experiments. The models for the transfer function from x_{sp} to x_m are[1]

$$G_1(s) = G^x_{ANT-X}(s) = e^{-0.25s} \cdot \frac{0.08s + 1.94}{s^2 + 1.75s + 2.01}\,, \tag{10.12}$$

[1] The PBSID algorithm returns a discrete-time state space model, but the continuous-time transfer functions obtained with the Tustin approximation has been reported to give more physical insight.

$$G_2(s) = G^x_{ADAM-0}(s) = e^{-0.20s} \cdot \frac{0.02s + 1.97}{s^2 + 1.92 + 1.94} \,. \tag{10.13}$$

Note that due to symmetry, and based on previous experience, we can use the models identified for longitudinal dynamics (x-direction) also for the lateral dynamics (y-direction), *i.e.*, assume that ($G^x_{(\cdot)}(s) = G^y_{(\cdot)}(s)$).

10.4.4 Dynamical transfer map

Based on the identified models, the structure assigned to the transfer map is changed with respect to equation (10.10), opting for a third order polynomial on both numerator and denominator. Two dynamical transfer maps are computed: $M_{nom}(s)$ considering no uncertainty on the identified models, and $M_{unc}(s)$ considering uncertainty on the *target* model (assuming standard deviation equal to 10% for the denominator coefficients and equal to 15% for the numerator ones). Using the proposed H_∞-based TL approach with the same performance weight function $W(s)$ of Section 10.3, the obtained transfer maps are:

$$M_{nom}(s) = \frac{0.14s^3 + 1.79s^2 + 1.79s + 1}{0.15s^3 + 1.66s^2 + 1.78s + 1} \,, \tag{10.14}$$

$$M_{unc}(s) = \frac{0.50s^3 + 1.34s^2 + 1.60s + 1}{0.49s^3 + 1.35s^2 + 1.50s + 1} \,. \tag{10.15}$$

10.4.5 Results

In this Section, we present the results obtained by applying the proposed TL algorithm. In the experiment the manoeuvre to be learned is an eight-shape trajectory flown in the horizontal plane. The quadrotors are required to hover at the beginning of the eight-shape trajectory: the learning motion starts and ends in the same hovering point. The acceleration and deceleration phases at the beginning and the end of the eight-figure must also be learned precisely. The flight tests with the two quadrotors have been carried out in two phases. Firstly, we perform several ILC iterations on the *source* system (until the average position error becomes less than 5 cm). The data related to the S-ILC algorithm on the *source* system is reported in Table 10.3. We can notice that only three iterations are needed to obtain the average position error below 5 cm ($e_{pos_3} = 3\,\text{cm}$).

Table 10.3: *S-ILC results on the* source *system.*

Iteration	e_{pos_j} [m]	$\frac{e_{pos_j}}{e_{pos_0}} \cdot 100$ [%]	$\frac{\lvert e_{pos_j} - e_{pos_{j-1}} \rvert}{\lvert e_{pos_{j-1}} \rvert} \cdot 100$ [%]
$j = 0$	0.7675	100.0	-
$j = 1$	0.0902	11.75	88.25
$j = 2$	0.0296	3.86	67.18

Then, the performance of the *target* system is evaluated in three different scenarios: without TL, using the nominal transfer map $M_{nom}(s)$ and the uncertain transfer map $M_{unc}(s)$. In Table 10.4 the results obtained with the *target* system are listed highlighting the performance improvement with respect the case without TL ($e_{pos_{NoTL}}$).

Table 10.4: *TL results on the* target *system.*

Case	e_{pos_j} [m]	$\frac{e_{pos_j}}{e_{pos_{NoTL}}} \cdot 100$ [%]
No TL	0.7244	100.0
TL with $M_{nom}(s)$	0.0814	11.2
TL with $M_{unc}(s)$	0.0989	13.6

As expected[2], the performance is slightly better using the nominal transfer map $M_{nom}(s)$, but also using $M_{unc}(s)$ a consistent error reduction (86.4%) is obtained with respect to the case without TL. This reduction confirms the potentiality of the proposed approach that takes into account the models' uncertainty in a systematic way. In Figure 10.10 and Figure 10.11 the time evolution of the North and East position are plotted respectively. Furthermore, in Figure 10.12 the evolution in the xy-plane is depicted (without the acceleration and deceleration phases). We can state that the proposed TL approach is very effective in improving the tracking performance of a UAV transferring knowledge between different-scale UAVs.

[2] In this case, despite the identified models being very accurate, a high level of uncertainty is considered in the H_∞ optimization to validate the proposed approach. As a consequence, the optimised dynamical transfer map is more conservative than needed.

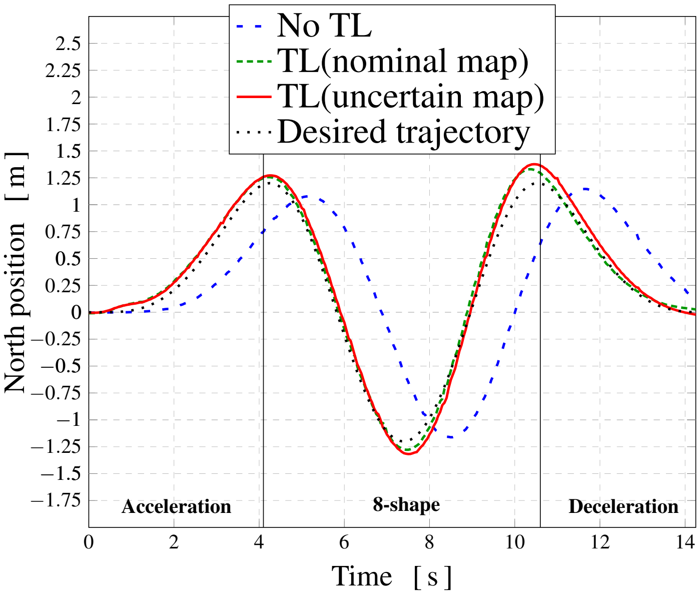


Figure 10.10: *UAV North position with and without TL.*

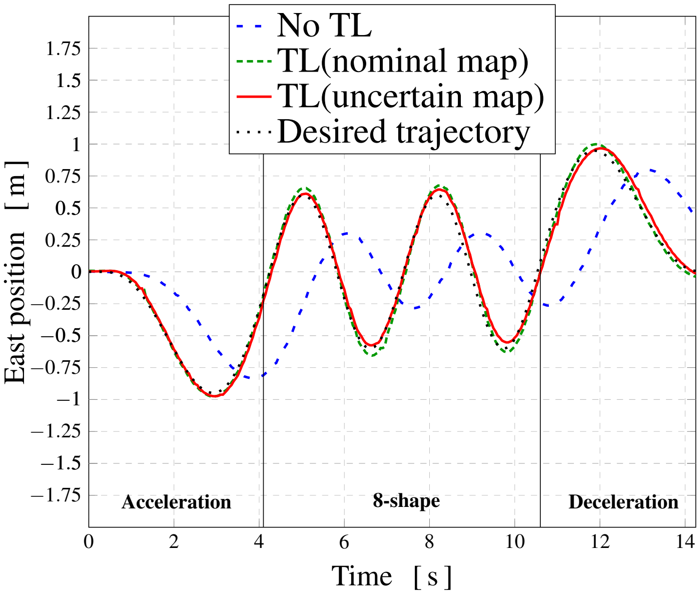


Figure 10.11: *UAV East position with and without TL.*

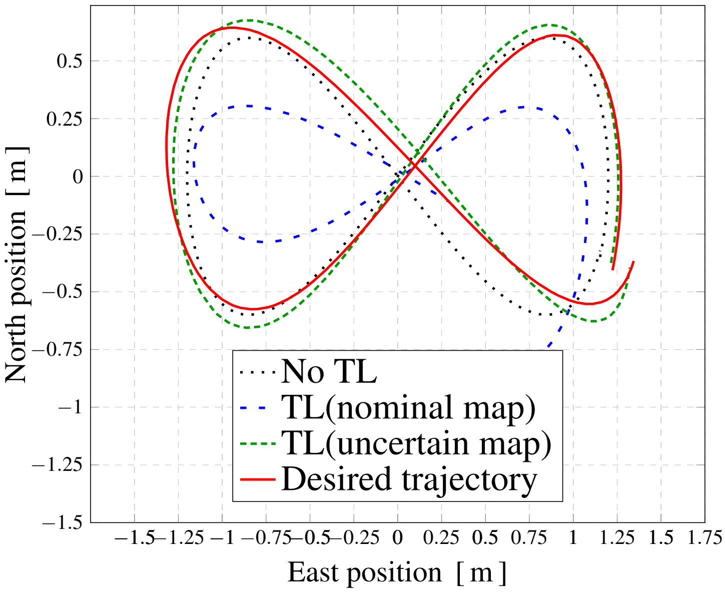


Figure 10.12: *8-shape trajectories with and without TL.*

10.5 Final considerations

In this Chapter, we define a robust and systematic approach to transfer data acquired during the learning phase of a drone (*source*) and to use it to train a different-scale one (*target*) to perform the same task. The developed approach is based on the H_∞ optimisation of a transfer function and takes advantage of the H_∞ control design framework by adapting the formulation to the TL problem. Furthermore, we combine the $\mathcal{L}_1$ adaptive control with the S-ILC proposed in the previous Chapter. The adaptive controller drives dynamically different nonlinear systems to behave as a defined linear model over iterations, whereas the ILC algorithm increases tracking performance over iterations. The validation of the proposed approach has been performed using two identified models of the *source* and *target* system. However, in certain applications, only one identified model could be available. In these cases, a dynamic scaling approach (see, *e.g.*,

[282]) can be applied to obtain the other model starting from the identified one, and the obtained model can be used to build the transfer map and tune the ILC algorithms. Finally, future works could exploit the proposed H_∞ framework to derive a procedure that allows us to guarantee that the transferred experience will lead to improved performance (*i.e.*, positive transfer) on the *target* system prior to transferring experience from the *source* system.

Conclusions

This dissertation focuses on the development, simulation and experimental validation of various control algorithms for aerospace systems, ranging from small UAVs to satellites with large rotating antennas. The main contribution is the development of systematic approaches that integrate the experience gathered during the operation (*data-driven* knowledge) into a classic (*model-based*) control framework. Specifically, some innovative control architectures are proposed and discussed.

The first Part of this work addresses two challenges for verifiable adaptive control systems, *i.e.*, the presence of actuator dynamics and systems with uncertain time-varying parameters. The resulting methods make it possible to check and satisfy different conditions such that the adaptive control system has performance and robustness guarantees. Finally, the proposed algorithms are specialized to a UAV position controller and an attitude controller for a spacecraft with time-varying inertial parameters.

The second Part of this work is devoted to presenting and discussing two applications of HC algorithms. Specifically, an active balancing system for rotating orbital devices has been designed and a HC algorithm has been used to command the positions of the actuated masses. After extensive numerical simulations, a dedicated breadboard has been developed, and experimental validation of the control law has been carried out.

Then, a second application of HC with a slightly different paradigm is presented. In particular, the HC framework is specialized in the helicopter rotor-induced vibration problem and the analysis of the interaction between the active control of structural response and a mast vibration absorber is done using a virtual helicopter model.

The last Part of this work presents the novel *Smoother-based* ILC algorithm that takes advantage of the repetition of a specific task to improve performance from one trial to the next. This algorithm has been specialized to achieve high-performance trajectory tracking with UAVs, and an experimental campaign involving a small quadrotor has been carried out. However, the learning phase needed to apply such a technique is related to each specific system, thus making the application of ILC poorly scalable. To overcome this limitation, a novel H_∞-optimization-based definition of a transfer map is proposed that allows transferring the knowledge acquired on a *source* system to a *target* system that needs to perform the same task. This transfer of data may be beneficial if the *source* system is less costly, difficult, or hazardous to operate than the *target* system. Furthermore, if multiple similar robots have to perform the same task, it is more cost-effective if one robot learns to perform the task and transfers its knowledge to the other robots. Experimental results demonstrate the effectiveness of the proposed approach in improving the tracking performance of a UAV transferring knowledge between different-scale UAVs.

Bibliography

[1] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*, vol. 1. Springer, 2015.

[2] P. Liu, R. Yang, and Z. Xu, "How safe is safe enough for self-driving vehicles?," *Risk Analysis*, vol. 39, no. 2, pp. 315–325, 2019.

[3] G. E. Musgrave, A. Larsen, and T. Sgobba, *Safety design for space systems*. Butterworth-Heinemann, 2009.

[4] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. M. Gregory, "$\mathcal{L}_1$ adaptive control for safety-critical systems," *IEEE Control Systems Magazine*, vol. 31, no. 5, pp. 54–104, 2011.

[5] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice-Hall, Inc., 1996.

[6] N. Sebe, I. Cohen, A. Garg, and T. S. Huang, *Machine learning in computer vision*, vol. 29. Springer Science & Business Media, 2005.

[7] F. Olsson, *A literature survey of active machine learning in the context of natural language processing*. Swedish Institute of Computer Science, 2009.

[8] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, “Safe learning in robotics: From learning-based control to safe reinforcement learning,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[9] P. A. Ioannou and J. Sun, *Robust adaptive control*, vol. 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[10] V. Nannen, *A Feasibility Study of Helicopter Vibration Reduction by Self-optimizing Higher Harmonic Blade Pitch Control*. PhD thesis, Massachusetts Institute of Technology, USA, 1967.

[11] D. Bristow, M. Tharayil, and A. Alleyne, “A survey of iterative learning control,” *IEEE Control Systems Magazine*, vol. 26, pp. 96–114, June 2006.

[12] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Courier Corporation, 2012.

[13] R. F. Drenick and R. A. Shahbender, “Adaptive servomechanisms,” *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 76, no. 5, pp. 286–292, 1957.

[14] V. Chalam, *Adaptive control systems: Techniques and applications*. Routledge, 2017.

[15] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, M. A. Bolender, and E. Lavretsky, “Connections between adaptive control and optimization in machine learning,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4563–4568, 2019.

[16] E. Lavretsky and K. A. Wise, *Robust and adaptive control with aerospace applications*. Springer London, 2013.

[17] A. M. Annaswamy and A. L. Fradkov, “A historical perspective of adaptive control and learning,” *Annual Reviews in Control*, vol. 52, pp. 18–41, 2021.

[18] P. C. Gregory, “Air research and development command plans and programs,” in *Proceedings of Self Adaptive Flight Control Symposium*, pp. 8–15, OH: Wright-Patterson Air Force Base, 1959.

[19] E. Mishkin and L. Braun, *Adaptive control systems.* McGraw-Hill, 1961.

[20] H. P. Whitaker, J. Yamron, and A. Kezer, *Design of model-reference adaptive control systems for aircraft.* Report: Massachusetts Institute of Technology, Instrumentation Laboratory, 1958.

[21] P. Kokotovic, J. Medanic, M. Vuskovic, and S. Bingulac, “Sensitivity method in the experimental design of adaptive control systems,” in *Proceeding of 3rd IFAC World Congress*, vol. 1, pp. 1–12, 1966.

[22] P. Osburn, *New Developments in the Design of Model Reference Adaptive Control Systems.* Report: Institute of the Aerospace Sciences, 1961.

[23] R. E. Kalman, “Design of a self-optimizing control system,” *Transactions of the American Society of Mechanical Engineers*, vol. 80, no. 2, pp. 468–477, 1958.

[24] K. Åström, “Adaptive control around 1960,” *IEEE Control Systems Magazine*, vol. 16, no. 3, pp. 44–49, 1996.

[25] L. W. Taylor and E. J. Adkins, “Adaptive control and the X-15,” in *Princeton University Conference on Aircraft Flying Qualities, Princeton, NJ*, p. 26, 1965.

[26] P. H. Phillipson, “Design Methods for Model-Reference Adaptive Systems,” *Proceedings of the Institution of Mechanical Engineers*, vol. 183, no. 1, pp. 695–706, 1968.

[27] B. Shackcloth and R. But Chart, “Synthesis of Model Reference Adaptive Systems by Lyapunov’s Second Method,” *IFAC Proceedings Volumes*, vol. 2, no. 2, pp. 145–152, 1965.

[28] K. S. Narendra and P. Kudva, “Stable Adaptive Schemes for System Identification and Control - Part I,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-4, no. 6, pp. 542–551, 1974.

[29] K. Narendra, Y.-H. Lin, and L. Valavani, “Stable adaptive controller design, part II: Proof of stability,” *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 440–448, 1980.

[30] B. D. Anderson, “Adaptive systems, lack of persistency of excitation and bursting phenomena,” *Automatica*, vol. 21, no. 3, pp. 247–258, 1985.

[31] B. Egardt, *Stability of adaptive controllers*. Springer, 1979.

[32] C. Rohrs, L. Valavani, M. Athans, and G. Stein, “Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics,” *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 881–889, 1985.

[33] P. Ioannou and K. Tsakalis, “A robust direct adaptive controller,” *IEEE Transactions on Automatic Control*, vol. 31, no. 11, pp. 1033–1043, 1986.

[34] K. Narendra and A. Annaswamy, “A new adaptive law for robust adaptation without persistent excitation,” *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, 1987.

[35] R. Ortega and Y. Tang, “Robustness of adaptive controllers A survey,” *Automatica*, vol. 25, no. 5, pp. 651–677, 1989.

[36] R. Kosut, B. Anderson, and I. Mareels, “Stability theory for adaptive systems: Method of averaging and persistency of excitation,” *IEEE Transactions on Automatic Control*, vol. 32, no. 1, pp. 26–34, 1987.

[37] G. Tao, “Multivariable adaptive control: A survey,” *Automatica*, vol. 50, no. 11, pp. 2737–2764, 2014.

[38] D. Seo and M. R. Akella, “Non-certainty equivalent adaptive control for robot manipulator systems,” *Systems & Control Letters*, vol. 58, no. 4, pp. 304–308, 2009.

[39] H. Dong, Q. Hu, M. R. Akella, and H. Yang, “Composite adaptive attitude-tracking control with parameter convergence under finite excitation,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2657–2664, 2020.

[40] A. Astolfi and R. Ortega, “Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 590–606, 2003.

[41] N. Hovakimyan and C. Cao, *$\mathcal{L}_1$ Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM, 2010.

[42] M.-H. Hua, T. Hamel, P. Morin, and C. Samson, “Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles,” *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, 2013.

[43] D. Invernizzi, S. Panza, and M. Lovera, “Robust tuning of geometric attitude controllers for multirotor unmanned aerial vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 7, pp. 1332–1343, 2020.

[44] Z. Zuo and S. Mallikarjunan, “$\mathcal{L}_1$ adaptive backstepping for robust trajectory tracking of UAVs,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 2944–2954, 2017.

[45] T. Madani and A. Benallegue, “Backstepping sliding mode control applied to a miniature quadrotor flying robot,” in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pp. 700–705, 2006.

[46] R. Naldi, M. Furci, R. G. Sanfelice, and L. Marconi, “Robust global trajectory tracking for underactuated VTOL aerial vehicles using inner-outer loop control paradigms,” *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 97–112, 2017.

[47] A. Schoellig and R. D’Andrea, “Optimization-based Iterative Learning Control for trajectory tracking,” in *2009 European Control Conference (ECC)*, pp. 1505–1510, 2009.

[48] S. Meraglia and M. Lovera, “Smoother-Based Iterative Learning Control for UAV Trajectory Tracking,” *IEEE Control Systems Letters*, vol. 6, pp. 1501–1506, 2021.

[49] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, “Adaptive control of quadrotor UAVs: A design trade study with flight evaluations,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1400–1406, 2013.

[50] B. C. Gruenwald, T. Yucelen, K. M. Dogan, and J. A. Muse, “Expanded reference models for adaptive control of uncertain systems with actuator dynamics,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 3, pp. 475–489, 2020.

[51] B. C. Gruenwald, D. Wagner, T. Yucelen, and J. A. Muse, “Computing actuator bandwidth limits for model reference adaptive control,” *International Journal of Control*, vol. 89, no. 12, pp. 2434–2452, 2016.

[52] R., Mahony , V., Kumar, V. and P., Corke, “Multirotor aerial vehicles: Modeling, estimation and control of quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[53] S. Formentin and M. Lovera, “Flatness-based control of a quadrotor helicopter via feedforward linearization,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 6171–6176, 2011.

[54] A. Roza and M. Maggiore, “A class of position controllers for underactuated VTOL vehicles,” *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2580–2585, 2014.

[55] *PX4 community*. Dronecode Project, Inc., San Francisco, CA, USA. [Online]. Available: `https://docs.px4.io/en/`.

[56] S. Bertrand, N. Gunard, T. Hamel, H. Piet-Lahanier, and L. Eck, “A hierarchical controller for miniature VTOL UAVs: Design and stability analysis using singular perturbation theory,” *Control Engineering Practice*, vol. 19, no. 10, pp. 1099–1108, 2011.

[57] D. Invernizzi, M. Lovera, and L. Zaccarian, “Geometric tracking control of underactuated VTOL UAVs,” in *American Control Conference*, (Milwaukee (WI), USA), July 2018.

[58] K. J. Åström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

[59] E. N. Johnson, *Limited authority adaptive flight control*. School of Aerospace Engineering, Georgia Institute of Technology. Georgia Institute of Technology, 2000.

[60] B. C. Gruenwald, T. Yucelen, J. A. Muse, and D. Wagner, “Computing stability limits for adaptive control laws with high-order actuator dynamics,” *Automatica*, vol. 101, pp. 409–416, 2019.

[61] E. N. Johnson and A. J. Calise, “Limited authority adaptive flight control for reusable launch vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 906–913, 2003.

[62] S. Panza, D. Invernizzi, M. Giurato, and M. Lovera, “Design and characterization of the 2DoF drone: a multirotor platform for education and research,” *IFAC-PapersOnLine*, vol. 54, no. 12, pp. 32–37, 2021.

[63] D. Invernizzi, S. Panza, M. Giurato, G. Yang, K. Chen, M. Lovera, and T. Parisini, “Integration of experimental activities into remote teaching using a quadrotor test-bed,” *IFAC-PapersOnLine*, vol. 54, no. 12, pp. 49–54, 2021.

[64] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[65] G. Tao, *Adaptive control design and analysis*, vol. 37. John Wiley & Sons, 2003.

[66] A. Astolfi, D. Karagiannis, and R. Ortega, *Nonlinear and adaptive control with applications*, vol. 187. Springer, 2008.

[67] G. Goodwin and E. Khwang Teoh, “Adaptive control of a class of linear time varying systems,” *IFAC Proceedings Volumes*, vol. 16,

no. 9, pp. 1–6, 1983. IFAC Workshop on Adaptive Systems in Control and Signal Processing 1983, San Francisco, CA , USA, 20-22 June 1983.

[68] R. Middleton and G. Goodwin, "Adaptive control of time-varying linear systems," *IEEE Transactions on Automatic Control*, vol. 33, no. 2, pp. 150–155, 1988.

[69] K. Chen and A. Astolfi, "Adaptive control for systems with time-varying parameters," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 1986–2001, 2021.

[70] K. Tsakalis and P. Ioannou, "Adaptive control of linear time-varying plants: a new model reference controller structure," *IEEE Transactions on Automatic Control*, vol. 34, no. 10, pp. 1038–1046, 1989.

[71] R. Marino and P. Tomei, "Adaptive control of linear time-varying systems," *Automatica*, vol. 39, no. 4, pp. 651–659, 2003.

[72] R. Marino and P. Tomei, "Robust adaptive regulation of linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1301–1311, 2000.

[73] K. Narendra and J. Balakrishnan, "Improving transient response of adaptive control systems using multiple models and switching," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1861–1866, 1994.

[74] B. D. O. Anderson, T. S. Brinsmead, F. De Bruyne, J. Hespanha, D. Liberzon, and A. S. Morse, "Multiple model adaptive control. part 1: Finite controller coverings," *International Journal of Robust and Nonlinear Control*, vol. 10, no. 11-12, pp. 909–929, 2000.

[75] J. Hespanha, D. Liberzon, A. Stephen Morse, B. D. O. Anderson, T. S. Brinsmead, and F. De Bruyne, "Multiple model adaptive control. part 2: switching," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 5, pp. 479–496, 2001.

[76] K. Narendra and J. Balakrishnan, “Adaptive control using multiple models,” *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 171–187, 1997.

[77] K. Narendra and Z. Han, “A new approach to adaptive control using multiple models,” *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 8, pp. 778–799, 2012.

[78] B. D. O. Anderson, T. Brinsmead, D. Liberzon, and A. Stephen Morse, “Multiple model adaptive control with safe switching,” *International Journal of Adaptive Control and Signal Processing*, vol. 15, no. 5, pp. 445–470, 2001.

[79] Z. Han and K. S. Narendra, “New concepts in adaptive control using multiple models,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 78–89, 2012.

[80] J. E. Gaudio, A. M. Annaswamy, E. Lavretsky, and M. A. Bolender, “Fast parameter convergence in adaptive flight control,” in *AIAA Scitech 2020 Forum*, 2020.

[81] J. E. Gaudio, A. M. Annaswamy, E. Lavretsky, and M. Bolender, “Parameter estimation in adaptive control of time-varying systems under a range of excitation conditions,” *IEEE Transactions on Automatic Control*, pp. 1–1, 2021.

[82] E. Arabi and T. Yucelen, “Set-theoretic model reference adaptive control with time-varying performance bounds,” *International Journal of Control*, vol. 92, no. 11, pp. 2509–2520, 2019.

[83] J.-X. Xu, “A new periodic adaptive control approach for time-varying parameters with known periodicity,” *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 579–583, 2004.

[84] O. S. Patil, R. Sun, S. Bhasin, and W. E. Dixon, “Adaptive control of time-varying parameter systems with asymptotic tracking,” *IEEE Transactions on Automatic Control*, pp. 1–1, 2022.

[85] B. Xian, D. Dawson, M. de Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1206–1211, 2004.

[86] W. Lin and C. Qian, "Adaptive control of nonlinearly parameterized systems: the smooth feedback case," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1249–1266, 2002.

[87] W. Lin and C. Qian, "Adding one power integrator: a tool for global stabilization of high-order lower-triangular systems," *Systems and Control Letters*, vol. 39, no. 5, pp. 339–351, 2000.

[88] K. Chen and A. Astolfi, "Adaptive control of linear systems with time-varying parameters," in *2018 Annual American Control Conference (ACC)*, pp. 80–85, 2018.

[89] K. Chen and A. Astolfi, "Adaptive control for nonlinear systems with time-varying parameters and control coefficient," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3829–3834, 2020. 21st IFAC World Congress.

[90] Y.-Y. Chen, K. Chen, and A. Astolfi, "Adaptive formation tracking control for first-order agents with a time-varying flow parameter," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2481–2488, 2022.

[91] K. Chen and A. Astolfi, "I&I Adaptive Control for Systems with Varying Parameters," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2205–2210, 2018.

[92] K. Chen and A. Astolfi, "Output-Feedback *I&I* Adaptive Control for Linear Systems with Time-Varying Parameters," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1965–1970, 2019.

[93] K. Chen and A. Astolfi, "Adaptive control for systems with time-varying parametersa survey," *Trends in Nonlinear and Adaptive Control*, pp. 217–247, 2022.

[94] J.-B. Pomet and L. Praly, “Adaptive nonlinear regulation: estimation from the Lyapunov equation,” *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 729–740, 1992.

[95] J. Zhou and C. Wen, *Adaptive backstepping control of uncertain systems: Nonsmooth nonlinearities, interactions or time-variations*. Springer, 2008.

[96] R. Marino and P. Tomei, “An adaptive output feedback control for a class of nonlinear systems with time-varying parameters,” *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2190–2194, 1999.

[97] Y. Zhang, B. Fidan, and P. Ioannou, “Backstepping control of linear time-varying systems with known and unknown parameters,” *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 1908–1925, 2003.

[98] K. Chen and A. Astolfi, “Identification-based adaptive control for systems with time-varying parameters,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1083–1088, 2021.

[99] P. Parks, “Lyapunov redesign of model reference adaptive control systems,” *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 362–367, 1966.

[100] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, NJ: Prentice-Hall, 2002.

[101] J. Cortes, “Discontinuous dynamical systems,” *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[102] H. Khalil, “Improved performance of universal integral regulators,” *Journal of optimization theory and applications*, vol. 115, no. 3, pp. 571–586, 2002.

[103] R. J. Wallsgrove and M. R. Akella, “Globally stabilizing saturated attitude control in the presence of bounded unknown disturbances,” *Journal of guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 957–963, 2005.

[104] P. Ioannou and P. Kokotovic, "Robust redesign of adaptive control," *IEEE Transactions on Automatic Control*, vol. 29, no. 3, pp. 202–211, 1984.

[105] G. J. Balas, "Linear, parameter-varying control and its application to aerospace systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 763–796, 2002.

[106] S. Ortiz, C. T. Calafate, J.-C. Cano, P. Manzoni, and C. K. Toh, "A UAV-Based Content Delivery Architecture for Rural Areas and Future Smart Cities," *IEEE Internet Computing*, vol. 23, no. 1, pp. 29–36, 2019.

[107] Z. Guang, Z. Heming, and B. Liang, "Attitude dynamics of spacecraft with time-varying inertia during on-orbit refueling," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 8, pp. 1744–1754, 2018.

[108] S. Platt, R. Maple, and M. Franke, "Parachute extraction of a generic munition from a C-130 aircraft," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, p. 457, 2006.

[109] J.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1148–1162, 1991.

[110] M. Krstic and P. Tsiotras, "Inverse optimal stabilization of a rigid spacecraft," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1042–1049, 1999.

[111] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*, vol. 1286. Springer, 2014.

[112] S. S. Gao, K. Clark, J. Zackrisson, K. Maynard, L. Boccia, and J. Xu, *Antennas for Small Satellites*, ch. 15, pp. 596–628. John Wiley & Sons, Ltd, 2012.

[113] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[114] J. Michael, K. Chudej, and J. Pannek, "Modelling and optimal control of a docking maneuver with an uncontrolled satellite," *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 1135–1140, 2012. 7th Vienna International Conference on Mathematical Modelling.

[115] A. Sanyal, A. Fosbury, N. Chaturvedi, and D. S. Bernstein, "Inertia-free spacecraft attitude tracking with disturbance rejection and almost global stabilization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1167–1178, 2009.

[116] W. MacKunis, K. Dupree, N. Fitz-Coy, and W. Dixon, "Adaptive satellite attitude control in the presence of inertia and cmg gimbal friction uncertainties," *The Journal of the Astronautical Sciences*, vol. 56, no. 1, pp. 121–134, 2008.

[117] H. Leduc, D. Peaucelle, and C. Pittet, "LMI-based design of a robust direct adaptive attitude control for a satellite with uncertain parameters," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 367–372, 2016. 20th IFAC Symposium on Automatic Control in Aerospace 2016.

[118] J. K. Zelina, R. J. Prazenica, and T. Henderson, "Adaptive control for nonlinear time-varying rotational systems," in *AIAA Scitech 2021 Forum*, 2021.

[119] J. K. Zelina, K. M. Dogan, R. J. Prazenica, and T. Henderson, "Adaptive control for time varying systems with actuator dynamics and sensor noise," in *AIAA Scitech 2022 Forum*, 2022.

[120] P. M. Tiwari, S. Janardhanan, and M. un Nabi, "Attitude control using higher order sliding mode," *Aerospace Science and Technology*, vol. 54, pp. 108–113, 2016.

[121] B. Wu, D. Wang, and E. K. Poh, "High precision satellite attitude tracking control via iterative learning control," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 3, pp. 528–534, 2015.

[122] Y. Park, "Robust and optimal attitude control of spacecraft with inertia uncertainties using minimal kinematic parameters," *Aerospace Science and Technology*, vol. 54, pp. 276–285, 2016.

[123] Q. Hu, X. Tan, and M. R. Akella, “Finite-time fault-tolerant spacecraft attitude control with torque saturation,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2524–2537, 2017.

[124] R. Jin, X. Chen, Y. Geng, and Z. Hou, “LPV gain-scheduled attitude control for satellite with time-varying inertia,” *Aerospace Science and Technology*, vol. 80, pp. 424–432, 2018.

[125] A. Weiss, I. V. Kolmanovsky, and D. S. Bernstein, “Inertia-free attitude control of spacecraft with unknown time-varying mass distribution,” in *62nd International Astronautical Congress, Cape Town, SA*, 2011.

[126] D. Thakur, S. Srikant, and M. R. Akella, “Adaptive attitude-tracking control of spacecraft with uncertain time-varying inertia parameters,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 41–52, 2015.

[127] R. Marino, “High-gain feedback in non-linear control systems,” *International Journal of Control*, vol. 42, no. 6, pp. 1369–1385, 1985.

[128] J. Han, “From PID to Active Disturbance Rejection Control,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[129] J. Hong and D. Bernstein, “Bode integral constraints, collocation, and spillover in active noise and vibration control,” *IEEE Transactions on Control Systems Technology*, vol. 6, no. 1, pp. 111–120, 1998.

[130] M. Badlani, W. Kleinhenz, and C. Hsiao, “On vibration of rotating shafts,” *Mechanism and Machine Theory*, vol. 13, no. 5, pp. 555 – 564, 1978.

[131] E. D. Bishop and R. Parkinson, “On the use of balancing machines for flexible rotors,” *Journal of Engineering for Industry*, vol. 94, p. 561, 1972.

[132] S. Zhou and J. Shi, "Active balancing and vibration control of rotating machinery: A survey," *The Shock and Vibration Digest*, vol. 33, pp. 361–371, 2001.

[133] J. Shaw and N. Albion, "Active control of the helicopter rotor for vibration reduction," *Journal of the American Helicopter Society*, vol. 26, no. 3, pp. 32–39, 1981.

[134] P. Friedmann and T. A. Millott, "Vibration reduction in rotorcraft using active control: A comparison of various approaches," *Journal of Guidance, Control, and Dynamics*, vol. 18, pp. 664–673, 1995.

[135] D. Teves, G. Niesl, A. Blaas, and S. Jacklin, "The role of active control in future rotorcraft," in *Proc. of the 21st European Rotorcraft Forum, Saint Petersburg, Russia*, 1995.

[136] R. Mura, *Robust Harmonic Control For Disturbance Rejection: Methods And Applications*. PhD thesis, Politecnico di Milano, 2015.

[137] B. Bamieh, J. Pearson, and J. Boyd, "A general framework for linear periodic systems with applications to H_∞ sampled-data control," *Automatic Control, IEEE Transactions on*, vol. 37, pp. 418 – 435, 1992.

[138] P. G. Voulgaris, M. Dahleh, and L. Valavani, "H_∞ and H_2 optimal controllers for periodic and multirate systems," *Automatica*, vol. 30, pp. 251–263, 1991.

[139] H. Sandberg, E. Mollerstedt, and Bernhardsson, "Frequency-domain analysis of linear time-periodic systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 1971–1983, 2005.

[140] N. M. Wereley and S. R. Hall, "Frequency response of Linear Time Periodic systems," in *29th IEEE Conference on Decision and Control*, vol. 6, pp. 3650–3655, 1990.

[141] M. Lovera, P. Colaneri, C. Malpica, and R. Celi, "Closed-Loop Aeromechanical Stability of Hingeless Rotor Helicopters with

Higher Harmonic Control," *Journal Of Guidance, Control and Dynamics*, vol. 29, no. 1, pp. 179–189, 2006.

[142] H. C. Li, "A note on Toeplitz matrices," *World Academy of Science, Engineering and Technology*, vol. 62, pp. 343–346, 2010.

[143] D. Kucerovsky, K. Mousavand, and A. Sarraf, "On some properties of toeplitz matrices," *Cogent Mathematics*, pp. 1–17, 2016.

[144] N. Wereley, *Analysis and control of linear periodically time varying systems*. PhD thesis, Massachusetts Institute of Technology, 1990.

[145] F. J. McHugh and J. Shaw, "Helicopter Vibration Reduction with Higher Harmonic Blade Pitch," 1978.

[146] J. Shaw, *Higher harmonic blade pitch control : a system for helicopter vibration reduction*. PhD thesis, Massachusetts Institute of Technology, 1980.

[147] J. Shaw, N. Albion, E. J. Hanker, and R. S. Teal, "Higher Harmonic Control: Wind Tunnel Demonstration of Fully Effective Vibratory Hub Force Suppression," 1989.

[148] N. Wereley and S. Hall, "Linear control issues in the Higher Harmonic Control of helicopter vibrations," in *45th Forum of the American Helicopter Society*, 1989.

[149] J. Pearson and R. Goodall, "Adaptive schemes for the active control of helicopter structural response," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 61–72, 1994.

[150] D. Patt, L. Liu, J. Chandrasekar, D. S. Berstein, and P. Friedmann, "Higher-Harmonic-Control Algorithm for Helicopter Vibration Reduction Revisited," *Journal of Guidance Control and Dynamics*, vol. 28, 2005.

[151] J. Chandrasekar, L. Liu, D. Patt, P. Friedmann, and D. S. Bernstein, "Adaptive harmonic steady-state control for disturbance rejection," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 993 – 1007, 2006.

[152] M. Lovera, P. Colaneri, C. Malpica, and R. Celi, “Discrete-time, closed-loop aeromechanical stability analysis of helicopters with Higher Harmonic Control,” *Journal of Guidance, Control and Dynamics*, vol. 30, pp. 1249–1260, 2007.

[153] L. Valsecchi, P. Apollonio, M. Molina, M. Olivier, F. Bernelli-Zazzera, M. Lovera, and S. Bittanti, “Design and testing of an active damping system for the reduction of vibrations induced by a rotating device on the ISS,” in *54th Internation Astronautical Congress, Bremen, Germany*, 2003.

[154] D. Entekhabi and E. G. Njoku, “The Soil Moisture Active Passive (SMAP) mission,” *Proceedings of the IEEE*, vol. 98, no. 5, pp. 704–716, 2010.

[155] F. Vanin, P. Laberinti, C. Donlon, B. Fiorelli, I. Barat, M. P. Sole, M. Palladino, P. Eggers, T. Rudolph, and C. Galeazzi, “Copernicus Imaging Microwave Radiometer (CIMR): System aspects and technological challenges,” in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 6535–6538, 2020.

[156] R. Gonçalo, “Deployable structures activities at the European Space Agency’s structures section,” in *Proceeding of ECSSMET 2016-14th European Conference on Spacecraft Structures, Materials and Environmental Testing, Toulouse, France*, 2016.

[157] M. Mobrem, E. Keay, G. C. Marks, and E. Slimko, “Development of the large aperture reflector/boom assembly for the SMAP spacecraft,” 2012.

[158] C. Donlon, “Mission requirements document: Copernicus imaging microwave radiometer,” ESA Technical report, 2019.

[159] O. S. Alvarez-Salazar, D. Adams, M. Milman, R. Nayeri, S. Ploen, L. Sievers, E. Slimko, and R. Stephenson, “Pointing Architecture of SMAP’s Large Spinning Antenna,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.

[160] W. Qiuxiao and W. Fei", "A new vibration mechanism of balancing machine for satellite-borne spinning rotors," *Chinese Journal of Aeronautics*, vol. 27, no. 5, pp. 1318 – 1326, 2014.

[161] ESA, "Applications: CIMR." `https://www.esa.int/ESA_Multimedia/Images/2020/11/CIMR`.

[162] S. Meraglia, D. Invernizzi, and M. Lovera, "Active balancing systems for rotating orbital devices," in *25th Conference of the Italian Association of Aeronautics and Astronautics*, pp. 369–376, 2019.

[163] J. K. Jae and B. Agrawal, "Automatic mass balancing of air-bearing-based three-axis rotational spacecraft simulator," *Journal of Guidance Control and Dynamics*, vol. 32, pp. 1005–1017, 2009.

[164] D. Invernizzi, "Modeling and attitude control of spacecraft with an unbalanced rotating device," *IEEE Control Systems Letters*, vol. 7, pp. 466–471, 2023.

[165] M. De Stefano, H. Mishra, A. M. Giordano, R. Lampariello, and C. Ott, "A relative dynamics formulation for hardware-in-the-loop simulation of on-orbit robotic missions," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3569–3576, 2021.

[166] J. Zhou and T. Hagiwara, "H_2 and H_∞ norm computations of linear continuous-time periodic systems via the skew analysis of frequency response operators," *Automatica*, vol. 38, pp. 1381–1387, 2002.

[167] S. Stupar, A. Simonovic, and M. Jovanovic, "Measurement and analysis of vibrations on the helicopter structure in order to detect defects of operating elements," *Scientific Technical Review*, vol. 62, no. 1, pp. 58–63, 2012.

[168] K. Q. Nguyen, "Higher Harmonic Control analysis for vibration reduction of helicopter rotor systems," NASA Technical report, 1994.

[169] G. Reichert, "Helicopter vibration control: A survey," *Vertica*, vol. 5, 1991.

[170] C. A. Malpica, *Contributions to the dynamics of helicopters with active rotor control.* PhD thesis, University of Maryland, 2008.

[171] F. H. Fan, *Improved Continuous-Time Higher Harmonic Control Hinf Methods.* PhD thesis, Massachusetts Institute of Technology, 2013.

[172] N. Ham, "Helicopter Individual Blade Control and its applications," in *American Helicopter Society 39th Annual Forum*, pp. 613 – 623, 1983.

[173] C. Kessler, "Active rotor control for helicopters: Individual Blade Control and swashplateless rotor designs," *CEAS Aeronautical Journal*, vol. 1, no. 1-4, pp. 23–54, 2011.

[174] R. G. J.T. Pearson and I. Lyndon, "Active control of helicopter vibration," *Computing & Control Engineering Journal*, pp. 277–284, December 1994.

[175] M. Lovera, "Invariant and periodic filters for helicopter vibration attenuation," in *16th IFAC Symposium on Automatic Control in Aerospace, Saint Petersburg, Russia*, 2004.

[176] W. Johnson, *Self-tuning regulators for multicyclic control of helicopter vibration.* NASA Technical Paper, 1982.

[177] M. Lovera and R. Mura, "Baseline vibration attenuation in helicopters: Robust MIMO-HHC," in *IFAC Proceedings Volumes*, vol. 47, pp. 8855–8860, 2014.

[178] L. M. Bottasso, L. Medici, G. Bernardi, and E. Fosco, "A tunable Mast Vibration Absorber for variable RPM rotorcraft," *42nd European Rotorcraft Forum 2016*, vol. 2, no. 3, pp. 808–819, 2016.

[179] T. Krysinski and F. Malburet, *Mechanical Vibrations.* 2007.

[180] R. Cribbs, P. Friedmann, and T. Chiu, "Coupled helicopter rotor/flexible fuselage aeroelastic model for control of structural response," *AIAA journal*, vol. 38, no. 10, pp. 1777–1788, 2000.

[181] T. Kailath, *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[182] W. Johnson, *Rotorcraft Aeromechanics*. Cambridge Aerospace Series, Cambridge University Press, 2013.

[183] P. Masarati, V. Muscarello, and G. Quaranta, "Linearized aeroservoelastic analysis of rotary-wing aircraft," in *36th European Rotorcraft Forum*, pp. 099–1, 2010.

[184] A. Tamer, V. Muscarello, P. Masarati, and G. Quaranta, "Evaluation of vibration reduction devices for helicopter ride quality improvement," *Aerospace Science and Technology*, vol. 95, 2019.

[185] A. E. Staple, "An Evaluation of Active Control of Structural Response as a Means of Reducing Helicopter Vibration," in *15th European Rotorcraft Forum*, 1990.

[186] K. Moore, "Iterative Learning Control: An Expository Overview," in *Applied and Computational Control, Signals, and Circuits: Volume 1*, pp. 151–214, Boston, MA: Birkhäuser Boston, 1999.

[187] B. Francis and W. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[188] D. Owens and J. Hätönen, "Iterative learning control: An optimization paradigm," *Annual Reviews in Control*, vol. 29, no. 1, pp. 57–70, 2005.

[189] L. Cuiyan, Z. Dongchun, and Z. Xianyi, "A survey of repetitive control," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 2, pp. 1160–1166 vol.2, 2004.

[190] H. Ahn, Y. Chen, and K. Moore, "Iterative Learning Control: Brief Survey and Categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, 2007.

[191] K. L. Moore, *Iterative learning control for deterministic systems*. Springer Science & Business Media, 1993.

[192] M. Uchiyama, “Formation of high-speed motion pattern of a mechanical arm by trial,” *Transactions of the Society of Instrument and Control Engineers*, vol. 14, no. 6, pp. 706–712, 1978.

[193] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning,” *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[194] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems,” in *The 23rd IEEE Conference on Decision and Control*, pp. 1064–1069, 1984.

[195] D. Kim and S. Kim, “An iterative learning control method with application for cnc machine tools,” in *Conference Record of the 1993 IEEE Industry Applications Conference Twenty-Eighth IAS Annual Meeting*, pp. 2106–2111 vol.3, 1993.

[196] M. Pandit and K.-H. Buchheit, “Optimizing iterative learning control of cyclic production processes with application to extruders,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 3, pp. 382–390, 1999.

[197] S. Garimella and K. Srinivasan, “Application of iterative learning control to coil-to-coil control in rolling,” in *Proceedings of 1995 American Control Conference - ACC’95*, vol. 2, pp. 1230–1234 vol.2, 1995.

[198] H. Ahn, K. Moore, and Y. Chen, *Iterative learning control: robustness and monotonic convergence for interval systems*. Springer Science & Business Media, 2007.

[199] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, “Stability of learning control with disturbances and uncertain initial conditions,” *IEEE Transactions on Automatic Control*, vol. 37, no. 1, pp. 110–114, 1992.

[200] S. Kawamura, F. Miyazaki, and S. Arimoto, “Applications of learning method for dynamic control of robot manipulators,” in *1985*

24th IEEE Conference on Decision and Control, pp. 1381–1386, 1985.

[201] Z. Bien and J.-X. Xu, *Iterative learning control: analysis, design, integration and applications.* Springer Science & Business Media, 1998.

[202] R. Longman, "Iterative learning control and repetitive control for engineering practice," *International journal of control*, vol. 73, no. 10, pp. 930–954, 2000.

[203] Y. Q. Chen and K. L. Moore, "A practical iterative learning path-following control of an omni-directional vehicle," *Asian Journal of Control*, vol. 4, no. 1, pp. 90–98, 2002.

[204] R. Adlakha and M. Zheng, "An optimization-based iterative learning control design method for uavs trajectory tracking," in *2020 American Control Conference (ACC)*, pp. 1353–1359, 2020.

[205] M. Hehn and R. D'Andrea, "A frequency domain iterative feed-forward learning scheme for high performance periodic quadrocopter maneuvers," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2445–2451, 2013.

[206] Y. Chen, Z. Gong, and C. Wen, "Analysis of a high-order Iterative Learning Control algorithm for uncertain nonlinear systems with state delays," *Automatica*, vol. 34, no. 3, pp. 345–353, 1998.

[207] K. Moore, "Multi-loop control approach to designing iterative learning controllers," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 1, pp. 666–671 vol.1, 1998.

[208] Z. Cai, D. Tong, C. T. Freeman, and E. Rogers, "Application of newton-method based ilc to 3d stroke rehabilitation," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 4851–4856, 2011. 18th IFAC World Congress.

[209] Y. Wang, F. Gao, and F. Doyle, “Survey on iterative learning control, repetitive control, and run-to-run control,” *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[210] S. Arimoto, “A brief history of iterative learning control,” in *Iterative Learning Control*, pp. 3–7, Springer, 1998.

[211] P. Goldsmith, “On the equivalence of causal LTI iterative learning control and feedback control,” *Automatica*, vol. 38, no. 4, pp. 703 – 708, 2002.

[212] D. Owens and E. Rogers, “Comments on ”On the equivalence of causal LTI iterative learning control and feedback control”,” *Automatica*, vol. 40, no. 5, pp. 895 – 898, 2004.

[213] P. J. Antsaklis, “Intelligent learning control,” *IEEE Control Systems Magazine*, vol. 15, no. 3, pp. 5–7, 1995.

[214] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[215] L. A. Zadeh, “Fuzzy logic,” *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[216] Y. Wang and F. J. Doyle, “Stability analysis for set-point-related indirect iterative learning control,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 5702–5707, 2009.

[217] D. D. Roover and O. H. Bosgra, “Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system,” *International Journal of Control*, vol. 73, no. 10, pp. 968–979, 2000.

[218] T. D. Son, G. Pipeleers, and J. Swevers, “Multi-objective iterative learning control using convex optimization,” *European Journal of Control*, vol. 33, pp. 35–42, 2017.

[219] G. Pipeleers and K. L. Moore, “Unified analysis of iterative learning and repetitive controllers in trial domain,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 953–965, 2014.

[220] J. D. Ratcliffe, J. J. Htnen, P. L. Lewin, E. Rogers, T. J. Harte, and D. H. Owens, "P-type Iterative Learning Control for systems that contain resonance," *International Journal of Adaptive Control and Signal Processing*, vol. 19, no. 10, pp. 769–796, 2005.

[221] S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011–2016, 2001.

[222] Y. Jiang, D. Clements, and T. Hesketh, "Betterment learning control of nonlinear systems," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 2, pp. 1702–1707 vol.2, 1995.

[223] T. Sugie and T. Ono, "An Iterative Learning Control law for dynamical systems," *Automatica*, vol. 27, no. 4, pp. 729–732, 1991.

[224] T.-J. Jang, C.-H. Choi, and H.-S. Ahn, "Iterative Learning Control in feedback systems," *Automatica*, vol. 31, no. 2, pp. 243–248, 1995.

[225] C.-J. Chien and J.-S. Liu, "A P-type Iterative Learning Controller for robust output tracking of nonlinear time-varying systems," *International Journal of Control*, vol. 64, no. 2, pp. 319–334, 1996.

[226] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 636–641, 2002.

[227] J.-X. Xu and Y. Tan, "Robust optimal design and convergence properties analysis of iterative learning control approaches," *Automatica*, vol. 38, no. 11, pp. 1867–1880, 2002.

[228] R. W. Longman and S.-L. Wirkander, "Automated tuning concepts for iterative learning and repetitive control laws," *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 1, pp. 192–198, 1998.

[229] K. Kinosita, T. Sogo, and N. Adachi, "Iterative learning control using adjoint systems and stable inversion," *Asian Journal of Control*, vol. 4, no. 1, pp. 60–67, 2002.

[230] T. D. Son, G. Pipeleers, and J. Swevers, "Robust monotonic convergent iterative learning control," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1063–1068, 2016.

[231] M. Togai and O. Yamano, "Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach," in *24th IEEE Conference on Decision and Control*, pp. 1399–1404, 1985.

[232] K. L. Barton and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.

[233] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *International Journal of Control*, vol. 65, no. 2, pp. 277–293, 1996.

[234] N. Amann, D. H. Owens, and E. Rogers, "Predictive optimal iterative learning control," *International Journal of Control*, vol. 69, no. 2, pp. 203–226, 1998.

[235] D. Gorinevsky, "Loop shaping for Iterative Control of batch processes," *IEEE Control Systems Magazine*, vol. 22, no. 6, pp. 55–65, 2002.

[236] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.

[237] J. van de Wijdeven and O. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," *International Journal of Control*, vol. 83, no. 4, pp. 661–675, 2010.

[238] B. Dijkstra and O. Bosgra, "Convergence design considerations of low order q-ilc for closed loop systems, implemented on a high

precision wafer stage," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 2494–2499, 2002.

[239] B. Dijkstra and O. Bosgra, "Extrapolation of optimal lifted system ilc solution, with application to a waferstage," in *Proceedings of the 2002 American Control Conference*, vol. 4, pp. 2595–2600, 2002.

[240] J. Scherer and B. Rinner, "Persistent multi-UAV surveillance with energy and communication constraints," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1225–1230, 2016.

[241] P. Bechtle, M. Schelbergen, R. Schmehl, U. Zillmann, and S. Watson, "Airborne wind energy resource analysis," *Renewable Energy*, vol. 141, pp. 1103–1116, 2019.

[242] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "Iterative Learning-based waypoint optimization for repetitive path planning, with application to Airborne Wind Energy systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2698–2704, 2017.

[243] A. Cherubini, A. Papini, R. Vertechy, and M. Fontana, "Airborne Wind Energy Systems: A review of the technologies," *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1461–1476, 2015.

[244] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "An Iterative Learning Approach for Online Flight Path Optimization for Tethered Energy Systems Undergoing Cyclic Spooling Motion," in *2019 American Control Conference (ACC)*, pp. 2164–2170, 2019.

[245] M. Cobb, M. Wu, K. Barton, and C. Vermillion, "Flexible-time economic iterative learning control: A case study in airborne wind energy," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5580–5586, 2019.

[246] R. Adlakha and M. Zheng, "Two-Step Optimization-Based Iterative Learning Control for Quadrotor UAVs," *Journal of Dynamic Systems, Measurement, and Control*, 2021.

[247] H.-S. Ahn, K. Moore, and Y. Chen, "Kalman filter-augmented iterative learning control on the iteration domain," in *2006 American Control Conference*, pp. 6 pp.–, 2006.

[248] K. S. Lee and J. H. Lee, "Design of quadratic criterion-based iterative learning control," in *Iterative Learning Control*, pp. 165–192, Springer, 1998.

[249] M. Zhu, L. Ye, and X. Ma, "Estimation-Based Quadratic Iterative Learning Control for Trajectory Tracking of Robotic Manipulator With Uncertain Parameters," *IEEE Access*, vol. 8, pp. 43122–43133, 2020.

[250] P. Axelsson, R. Karlsson, and M. Norrlöf, "Estimation-based norm-optimal Iterative Learning Control," *Systems & Control Letters*, vol. 73, pp. 76–80, 2014.

[251] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[252] N. Degen and A. Schoellig, "Design of norm-optimal Iterative Learning controllers: The effect of an iteration-domain Kalman filter for disturbance estimation," in *53rd IEEE Conference on Decision and Control*, pp. 3590–3596, 2014.

[253] M. Norrlöf, "An adaptive Iterative Learning Control algorithm with experiments on an industrial robot," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 245–251, 2002.

[254] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, 2011.

[255] B. Panomruttanarug and R. Longman, "Using Kalman filter to attenuate noise in learning and repetitive control can easily degrade performance," in *2008 SICE Annual Conference*, pp. 3453–3458, 2008.

[256] G. V. der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods:

an overview," *IET Control Theory and Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[257] H. Elci, R. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Simple learning control made practical by zero-phase filtering: applications to robotics," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 753–767, 2002.

[258] J. L. Crassidis and F. L. Markley, "Minimum model error approach for attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 6, pp. 1241–1247, 1997.

[259] A. Bryson and M. Frazier, "Smoothing for linear and nonlinear dynamic systems," in *Proceedings of the optimum system synthesis conference*, pp. 353–364, 1963.

[260] H. E. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[261] S. R. McReynolds, "Fixed interval smoothing - revisited," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 5, pp. 913–921, 1990.

[262] J. Wallén, S. Gunnarsson, R. Henriksson, S. Moberg, and M. Norrlöf, "ILC applied to a flexible two-link robot model using sensor-fusion-based estimates," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 458–463, 2009.

[263] P. Axelsson, R. Karlsson, and M. Norrlöf, "Estimation-based ILC using particle filter with application to industrial manipulators," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1740–1745, 2013.

[264] J. Wallén, M. Norrlöf, and S. Gunnarsson, "A framework for analysis of observer-based ILC," *Asian Journal of Control*, vol. 13, no. 1, pp. 3–14, 2011.

[265] K. Pereida, M. Helwa, and A. Schoellig, “Data-Efficient Multi-Robot, Multi-Task Transfer Learning for Trajectory Tracking,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1260–1267, 2018.

[266] *ANT-X Website*. [Online]. Available: `https://antx.it/`.

[267] M. Bergamasco and M. Lovera, “Continuous-time predictor-based subspace identification for helicopter dynamics,” in *37th European Rotorcraft Forum*, Gallarate, Italy, 2011.

[268] J. Wartmann and S. Seher-Weiss, “Application of the Predictor-Based Subspace Identification Method to Rotorcraft System Identification,” in *39th European Rotorcraft Forum*, Moscow, Russia, 2013.

[269] M. Bergamasco and M. Lovera, “Identification of Linear Models for the Dynamics of a Hovering Quadrotor,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1696–1707, 2014.

[270] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[271] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[272] K. Raimalwala, B. Francis, and A. Schoellig, “An upper bound on the error of alignment-based transfer learning between two linear, time-invariant, scalar systems,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5253–5258, 2015.

[273] M. Hamer, M. Waibel, and R. D’Andrea, “Knowledge Transfer for High-Performance Quadrocopter Maneuvers,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1714–1719, 2013.

[274] M. Sorocky, S. Zhou, and A. Schoellig, "To Share or Not to Share? Performance Guarantees and the Asymmetric Nature of Cross-Robot Experience Transfer," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 923–928, 2020.

[275] K. Raimalwala, B. Francis, and A. Schoellig, "A preliminary study of transfer learning between unicycle robots," in *Proceedings of the AAAI Spring Symposium Series*, pp. 53–59, 2016.

[276] M. K. Helwa and A. P. Schoellig, "Multi-robot transfer learning: A dynamical system perspective," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4702–4708, 2017.

[277] P. Apkarian and D. Noll, "Nonsmooth H_∞ Synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[278] P. Gahinet and P. Apkarian, "Structured H_∞ Synthesis in MATLAB," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1435–1440, 2011. 18th IFAC World Congress.

[279] G. Bressan, A. Russo, D. Invernizzi, M. Giurato, S. Panza, and M. Lovera, "Adaptive augmentation of the attitude control system for a multirotor UAV," *Journal of Aerospace Engineering*, vol. 234, no. 10, pp. 1587–1596, 2018.

[280] K. Pereida, D. Kooijman, R. Duivenvoorden, and A. Schoellig, "Transfer learning for high-precision trajectory tracking through $\mathcal{L}_1$ adaptive feedback and iterative learning," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 2, pp. 388–409, 2019.

[281] K. Barton, S. Mishra, and E. Xargay, "Robust Iterative Learning Control: $\mathcal{L}_1$ adaptive feedback control in an ILC framework," in *Proceedings of the 2011 American Control Conference*, pp. 3663–3668, 2011.

[282] C. M. Ivler, E. S. Rowe, J. Martin, M. J. Lopez, and M. B. Tischler, "System identification guidance for multirotor aircraft: Dynamic

scaling and test techniques," *Journal of the American Helicopter Society*, vol. 66, no. 2, pp. 1–16, 2021.

[283] "Aerospace System and Control Laboratory - ASCL." http://ascl.daer.polimi.it/. Accessed: 03-11-2022.

[284] "Department of AERospace science and technology - DAER." https://www.aero.polimi.it/. Accessed: 03-11-2022.

[285] "Motive." https://optitrack.com/products/motive/. Accessed: 03-11-2022.

[286] "Optitrack motion capture system." https://optitrack.com/. Accessed: 03-11-2022.

[287] "ROS." http://www.ros.org. Accessed: 03-11-2022.

[288] "MATLAB." https://www.mathworks.com/products/matlab.html. Accessed: 03-11-2022.

[289] "Pixhawk Mini." https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk_mini.html. Accessed: 03-11-2022.

[290] "MAVLink." https://mavlink.io/en/. Accessed: 03-11-2022.

[291] "PX4." https://github.com/PX4/Firmware. Accessed: 03-11-2022.

[292] "QGroundControl." http://qgroundcontrol.com/. Accessed: 03-11-2022.

[293] "NanoPi NEO Air." http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Air. Accessed: 03-11-2022.

[294] "Mavros." https://github.com/mavlink/mavros. Accessed: 03-11-2022.

APPENDIX A

The Flying Arena for Rotorcraft Technologies

The Flying Arena for Rotorcraft Technologies (FlyART) is an indoor facility (see Figure A.1) designed and developed by the Aerospace Systems and Control Laboratory (ASCL [283]) which is the scientific laboratory of the Department of Aerospace Science and Technology of Politecnico di Milano (DAER [284]) devoted to systems and control research. FlyART facility allows the indoor testing and prototyping of small UAV concepts and new subsystems (*e.g.*, guidance, navigation and control systems) for single and formation flight. The FlyART has a flight volume of $12\,\mathrm{m} \times 6\,\mathrm{m} \times 4\,\mathrm{m}$ equipped with a motion capture system. In this Chapter, we describe the systems which compose the facility.

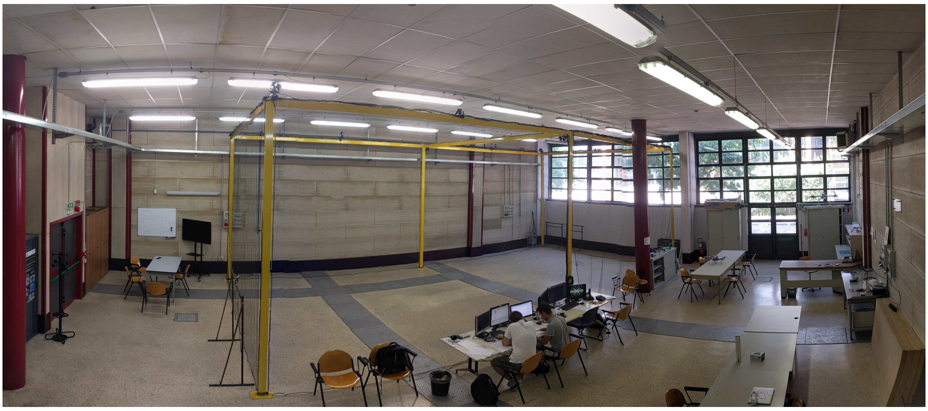


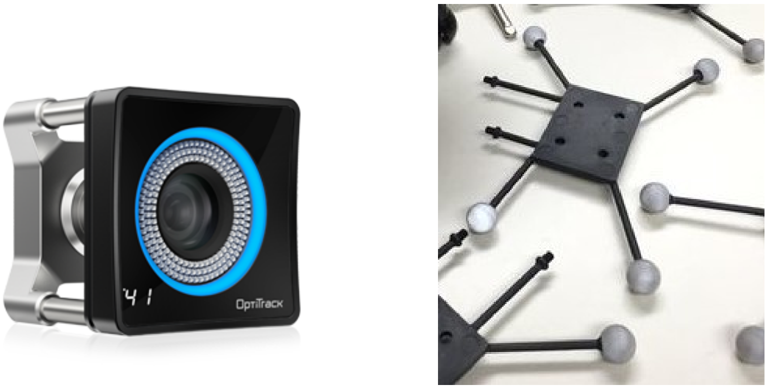
Figure A.1: *The FlyART facility.*

A.1 Motion Capture system

The Motion Capture system (Mo-Cap) is composed by 12 Infra-Red (IR) sensitive Optitrack cameras (Figure A.2(a)) which incorporate IR flood lights. The cameras, mounted on the FlyART, are fixed at calibrated positions and orientation so that the measurement subject is into the field of view of multiple cameras. Through markers sensitive to IR light (Figure A.2(b)) mounted on top of the drones, it is possible to estimate and track their attitude and positions inside the flight volume. Each UAV mounts a different marker layout to be uniquely identified when several drones fly at the same time. To control the Mo-Cap system, the Motive software [285] is installed on the ground station. It allows the user to calibrate the system and provides interfaces for capturing and processing 3D data that can be recorded or live-streamed. The accuracy of the position estimated onboard by the UAV depends on the frequency with which the position information is sent to it, which can be selected in the range $30-240\,\text{Hz}$.

A.2 Ground Control System

The Ground Control System (GCS) is dedicated to send and retrieve information from the UAVs during a flight session. The main task of the GCS is to read the attitude and position information provided by the mo-



(a) Infrared camera from [286].

(b) Infrared markers from [286].

Figure A.2: *Motion Capture System components.*

tion capture system and send it to the drones (at a frequency of 100 Hz). Furthermore, it is also possible to define a trajectory to be followed using specific way points (usually sent with a frequency of 20 Hz) and view telemetry data in real time. The GCS architecture involves two different computers: the former, which runs Windows 10, is used to execute Motive while the latter, which runs Linux OS (more precisely Ubuntu 16.04), is used to execute ROS [287] and MATLAB [288]. The GCS architectural division was done because Motive is a Windows software while ROS integrates better in a Linux environment.

A.3 Drones HW/SW architecture

Each UAV flying inside the indoor facility mounts two electronic boards, the Flight Control Unit (FCU) and the Flight Computer Companion (FCC).

Flight Control Unit

The FCU is an electronic board which runs the control and the navigation algorithms. The adopted FCU is the electronic board Pixhawk Mini [289] which integrates the inertial sensors, such as 3-axes accelerometer, 3-axes gyroscope, magnetometer and barometer.

Figure A.3: *Pixhawk Mini FCU from [289].*

The main features of the Pixhawk Mini are:

- *Processor*: main STM32F427 Rev 3 based on 32 bit ARM Cortex® M4 core with 180 *MHz* CPU and an IO processor STM32F103 based on the Cortex® M3 core with 72 *MHz* CPU;
- *Interfaces*: UART serial port for GPS, spektrum DSM/DSM2/DSM-X® satellite compatible RC input, Futaba S BUS® compatible RC input, PPM sum signal RC input, I2C for digital sensors, CAN for digital motor control with compatible controllers, ADC for analog sensors and micro USB port;
- *Weight*: 15.8 g;
- *Dimensions*: 38 mm × 43 mm × 12 mm.

The MAVLink [290] protocol is used for serial communication between FCU and FCC, PX4 firmware [291] is the one supported by Pixhawk Mini, and QGroundcontrol [292] is the software used to configure the Pixhawk Mini and to retrieve the drone status in real-time.

Flight Computer Companion

The FCC computer (Figure A.4) is the part of the drone system used to interface and communicate with PX4 using the MAVLink protocol. It enables a variety of functionalities, such as the possibility to execute processes that require heavy CPU load. During the flight test in the FlyART, the companion computer is used to receive from the GCS the position and attitude information (coming from the Mo-Cap system), and the commands to be executed. This information is sent to the FCU through the serial communication. On the drones the NanoPi NEO Air (see Figure A.4) is used with the characteristics reported in the following:

- *CPU name*: Quad-core Cortex-A7 1.2 *GHz*;
- *RAM*: 512 *MB*;
- *Wireless module*: 2.4 *GHz* 802.11 $b/g/n$;
- *Dimensions*: 40 mm × 40 mm;
- *Weight*: 7.9 g;
- *Power*: 5 V - 2 A.

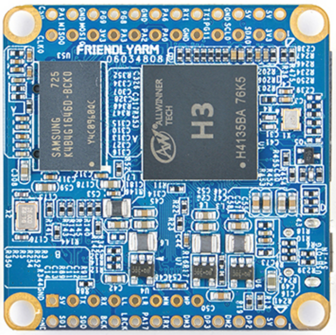


Figure A.4: *NanoPi NEO Air from [293].*

From a software point of view the Robot Operating System (ROS) [287] is used to communicate with GCS through ROS messages. In particular Mavros [294], that is a ROS package, provides communication driver for Pixhawk Mini autopilot with MAVLink communication protocol. Additionally, it provides User Datagram Protocol (UDP) MAVLink bridge for GCS (*e.g.*, QGroundControl).

A.4 UAV platforms

The aim of this Section is to describe the UAV platforms which have been used during the experimental activities of this Thesis.

ANT-X

The ANT-X drone (see Figure A.5) is a fixed-pitch quadrotor designed by ANT-X spin-off [266] with a maximum take-off weight below 300g. Its main features are:

- *Take Off Weight*: 0.270 kg;
- *Frame size*: $0.2 \times 0.2 \times 0.04$ m;
- *Diameter (motor-to-motor distance)*: 0.16 m;
- *Motors*: 4;
- *Propellers*: 4× three-bladed propellers of 3 inch;
- *Rotor configuration*: X configuration;
- *Battery*: LiPo 950 mAh 3S;
- *Flight time (hovering)*: 6.5 min.

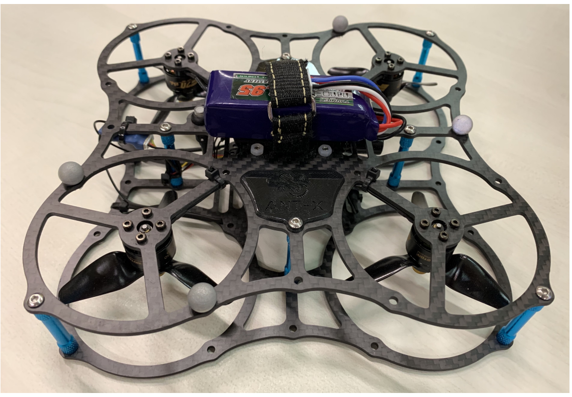


Figure A.5: *ANT-X quadcopter.*

ADAM-0

The ADAM-0 drone (see Figure A.6) is a quadrotor built with commercial off-the-shelf components in the FlyART. Its main features are:

- *Take Off Weight*: 1.4 kg;
- *Frame size*: 0.4 × 0.4 × 0.08 m;
- *Diameter (motor-to-motor distance)*: 0.55 m;
- *Motors*: 4× RCtimer High-performance series (HP2814) 810KV;
- *Propellers:* 4× two-bladed propellers 1245;
- *Rotor configuration*: X configuration;
- *Battery*: LiPo 4000 mAh 3S;
- *Flight time (hovering)*: 10 min.

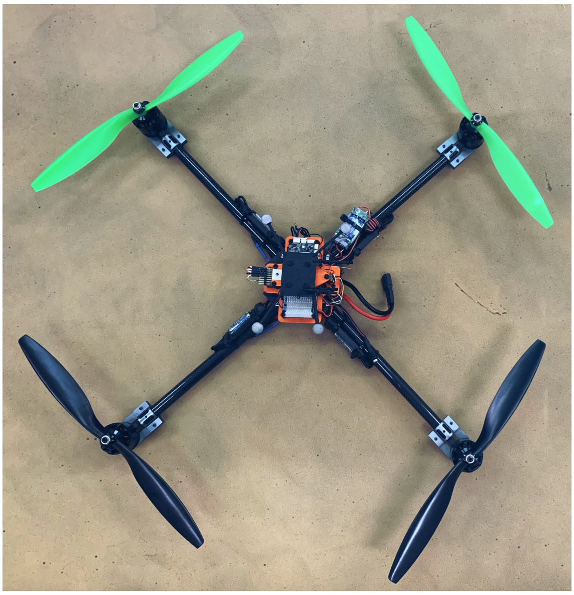


Figure A.6: *ADAM-0 quadcopter.*