Università degli Studi di Brescia

Dipartimento di Ingegneria Meccanica e Industriale

Ciclo XXXIII

Dottorato di Ricerca in Ingegneria Meccanica e Industriale
SSD: ING-INF/04

ADVANCED CONTROL TECHNIQUES FOR INDUSTRIAL MECHATRONIC SYSTEMS

Supervisor: Prof. **Antonio Visioli**
Tutor: Prof. **Giovanni Legnani**
Coordinator: Prof. **Laura Depero**

Ph.D. Candidate:
Marco Giacomelli
Marco Giacomelli

Marco Giacomelli: *Advanced control techniques for industrial mechatronic systems* ©
October 13, 2020

Ringraziamenti

I ringraziamenti per questa tesi vanno al prof. Visioli, per avere indirizzato la mia ricerca durante il dottorato, ed a Luca, con il quale ho condiviso buona parte delle gioie (e dei dolori) incontrati nel percorso che ha portato a questa tesi.

Ringrazio inoltre tutto il gruppo di sistemi di controllo (e chi intorno a questo gruppo gravita) per aver reso leggeri questi ultimi tre anni.

Gli ultimi ringraziamenti sono per le persone che mi hanno garantito la serenità per concentrare in questo percorso i miei sforzi: la mia famiglia, gli amici, e Chiara.

Contents

Abstract

In previsione di una sempre maggior dipendenza della produzione industriale dal mondo della meccatronica, alcuni aspetti dei sistemi meccatronici industriali devono essere investigati per garantire una maggiore efficienza dei sistemi produttivi stessi.

Per la necessità delle aziende di competere sul mercato, il mondo industriale sta chiedendo una sempre maggiore flessibilità degli impianti, in modo da consentire una riduzione del time to market del prodotto. Questo si scontra con la concezione classica di impianto industriale come macchina perfettamente tarata ed ottimizzata per la produzione di grandi lotti, ma sostanzialmente statica.

D'altro canto, la messa in funzione di sistemi meccatronici industriali comporta tipicamente la risoluzione di problematiche sempre nuove, molte delle quali riconducibili però a casistiche note e per le quali esistono già soluzioni, che sono spesso avanzate dal mondo accademico.

Per venire incontro a queste necessità, è necessario quindi colmare il gap esistente tra lo stato dell'arte della ricerca e le tecnologie industriali disponibili sul mercato.

Questa tesi riassume il lavoro svolto nella progettazione di alcuni dei moduli che si propongono di risolvere i problemi relativi alle elasticità nelle trasmissioni meccaniche nei sistemi meccatronici, alla definizione di uno schema di controllo e di una procedura automatica per la taratura di sistemi dedicati al motion control e ai disturbi di natura ripetitiva, che limitano la larghezza di banda del controllo.

Preface

Industrial world is moving at an increasing speed towards an interconnected, modular and flexible production paradigm. Production sites rely more and more on fully automated mechatronic systems. These kinds of systems were firstly born to perform a single task for all their lifespan, and their reconfiguration was hardly feasible. The change towards reconfigurable and flexible production flows forced to rethink the integration of cyber-physical systems, which are now required to be easily reconfigurable, interconnected and to guarantee high performance over a wide range of different applications.

While standardized solutions for the control of industrial cyber-physical systems are commercially available, their generality usually guarantees a performance that is not acceptable when the industrial market is asking for an increase of size, motion speed, precision, adaptability, self-diagnostic, connectivity, new cognitive features, etc.

Fulfillment of these requirements is essential for building smart, safe and reliable production systems. This implies completely new demands also on bottom layers of employed motion control systems which cannot be routinely handled by available commercial products. In fact, commercially available general purpose motion control systems guarantee, on most of the cases, only moderate performance as the mass market has no special requirements (such as, for example, high bandwidth position loops).

The use of these standardized control systems leads to a suboptimal performance, that can be pushed further only by means of ad-hoc solutions, which often require a significant investment on R&D.

On the contrary, many academic publications show that solutions for the most general limitations in standard industrial control hardware already exist, but such solutions cannot be applied on industrial environments due to stringent standards on reliability, robustness and safety on this field.

Many of the issues concerning the use of standard commercial solutions for the control of smart mechatronic systems for applications with high

performance requirements could be therefore solved by bridging the gap between the best industrial practice and latest research results.
The aim of this thesis is to make available to an industrial level a wide range of mechatronic systems solutions, both existing and new, to some of the most common problems of modern mechatronic systems in a modular and easily adaptable format, and to verify that the proposed solutions are implementable on standard off-the-shelf industrial control hardware.

The issues that are addressed in this work are the control and compensation of oscillations in underdamped mechatronic systems, the automatic tuning of control parameters for mechatronic systems and the rejection of repetitive disturbances affecting mechatronic systems.

The control of underdamped systems, discussed in Chapter 2, is treated by focusing on a key industrial application, that is the handling of heavy loads by means of industrial cranes. While this could be regarded as a specific application, the presented solutions have a general relevance for generic underactuated underdamped mechatronic systems.

The autotuning of control parameters for mechatronic systems is discussed in Chapter 3, where a set of automatic procedures for the tuning and compensation of oscillatory dynamics are presented. While a variety of approaches are present in literature, few of them focus on the respect of physical constraint during the tuning procedure, and even fewer address the issue of their implementability on industrial hardware, which is, for example, typically characterized by limitations in both computational and memory capability.
The algorithms presented in Chapter 3 are tested on the HIL setup described in Section 1.4.4, which consists of industrial off-the-shelf hardware, demonstrating their applicability to industrial applications.

Mechatronic systems can be also affected by repetitive disturbances, which can depend on the position of the system (i.e. the actuator position) or which can be periodic in time. By exploiting the information about their periodicity, it is possible to enhance the performance of the system by compensating them with approaches based on Repetitive Control (RC) and Iterative Learning Control (ILC). The application and validation of this techniques on industrial hardware is addressed in Chapter 4.

The results discussed in this thesis have been achieved in the frame of the I-MECH European Union project [13].

The results discussed in this thesis have been presented in the following scientific publications:

1. M. Giacomelli, F. Padula, L. Simoni, and A. Visioli. Simplified input-output inversion control of a double pendulum overhead crane for residual oscillations reduction. *Mechatronics, 56:37–47, 2018.*

2. M. Giacomelli, M. Faroni, D. Gorni, A. Marini, L. Simoni, and A. Visioli. Model predictive control for operator-in-the-loop overhead cranes. *In 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), volume 1, pages 589–596. IEEE, 2018.*

3. M. Giacomelli, M. Faroni, D. Gorni, A., L. Simoni, and A. Visioli. Mpc-pid control of operator-in-the-loop overhead cranes: A practical approach. *In 2018 7th International Conference on Systems and Control (ICSC), pages 321–326. IEEE, 2018.*

4. M. Giacomelli, D. Colombo, L. Simoni, G. Finzi, and A. Visioli. A fast autotuning method for velocity control of mechatronic systems. IFACPapersOnLine, 51(4):208–213, 2018.

5. M. Giacomelli, D. Colombo, G. Finzi, V. Šetka, L. Simoni, and A. Visioli. An autotuning procedure for motion control of oscillatory mechatronic systems. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 829–835, 2019.

6. M. Giacomelli, D. Colombo, G.Finzi, V. Šetka, L. Simoni, and A. Visioli. A closed-loop automatic tuning method for velocity control of

oscillatory mechatronic systems. In IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, volume 1, pages 3245–3250, 2019.

7. M. Giacomelli, L. Simoni, A. Visioli. A software tool to make primary school students aware of control systems, presented to 2020 IFAC World Congress, in press.

CHAPTER 1

Approach, methods and materials

1.1 Introduction

In this thesis, a series of solutions for smart mechatronic systems is presented and validated. While academical research on most of the topics touched by the thesis is vibrant, and new solutions are published in scientific journals on a daily basis, the approach of this thesis is to collect the most promising among them, together with some original contributions, and to convert them into modular solutions easily accessible and implementable on both greenfield and brownfield mechatronic industrial applications.

In order to do so, the V-model approach, described in Section 1.2, has been followed. Section 1.3, provides a context for the chosen approach, briefly introducing the I-MECH project. The applicability in the industrial field of the proposed techniques has then been validated by following the rapid prototyping paradigm described in Section 1.4.

1.2 The V-model approach

For the realization of the mechatronic solutions here presented, a V-model approach has been followed. The V-model paradigm, shown in Figure 1.1, consists in a pyramidal workflow. At the beginning, high-level design specifications and requirements raised by the industrial field have been identified, with a particular stress on the requirements characterized by a broad

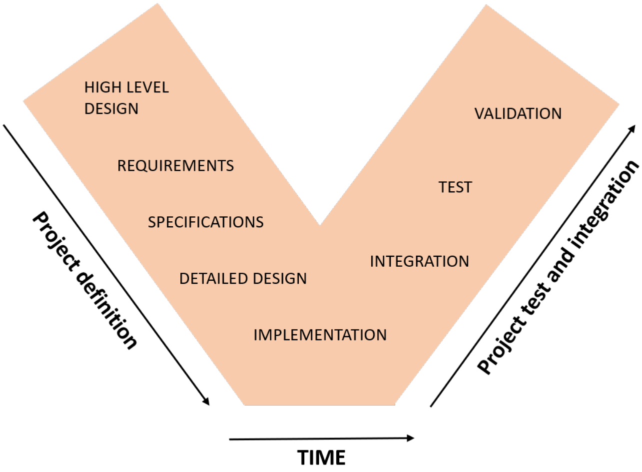


Figure 1.1: V-model approach (image after [50]).

generality among industrial mechatronic systems. The specification and implementation of the elaborated solutions were brought to completion after an accurate review of the scientific literature, so that both already existing and new implemented methodologies were scientifically consistent. In this high-level stage, the validation procedures are also defined.

The second phase of the V-model paradigm is the low-level design, in which hardware and software components are defined. Once the architecture is defined, the actual implementation takes place. In this phase, the coding of the solution must take into account all the requirements and specifications (e.g. accepted coding languages). Finally, the validation phase takes place, and the solutions are checked against the initially identified requirements and specifications. The validation can be performed through a variety of different approaches, both by using the instruments of the rapid prototyping paradigm or by actually implementing the solution on real industrial systems.

1.3 The I-MECH project

For the case of this thesis, use cases for the validation part have been found thanks to the collaboration with industrial partners in the frame of the Intelligent Motion Control Platform for Smart Mechatronic Systems (I-MECH) project, developed within the European Union H2020 program funded by ECSEL-2016-1. The consortium, composed by both technological and research professionals from 31 participating organisations and 10 European countries, was moved together by the common feeling that the limits in the control of cyber-physical systems can still be pushed further. In particular, a strong collaboration with Gefran s.p.a. allowed to have access to modern industrial hardware and software for the testing of the techniques proposed in the frame of the I-MECH project.
I-MECH project aimed to bridge the gap between the latest research results and best industrial practice by focusing on the sensors, actuators and physical control infrastructure of the motion control system more than on the application itself. Dividing the general structure of a control system in three layers, the first layer is composed by actuators and sensors, which physically interact with the plant. On the second layer the (digital) control takes place and the information coming from sensors are elaborated in order to control the plant using actuators. The third layer is the layer concerning the application, on which the system behaviour is defined in therms of, for example, the desired motion trajectory. This division into layers of a general motion control system is shown in Figure 1.2.

While the third layer can be very general, most industrial applications relies on the same stack of layers 1 and 2 and are affected by the same issues.

By developing modular and unified building blocks, each one responding to a specific need, the final outcome of the project is a modular solutions platform. In order to develop sustainable general techniques and solutions for the smart control of mechatronic systems, model-based approach was adopted, which is a standard practice and is integrated with modeling and simulation as well as digital enterprise functions.

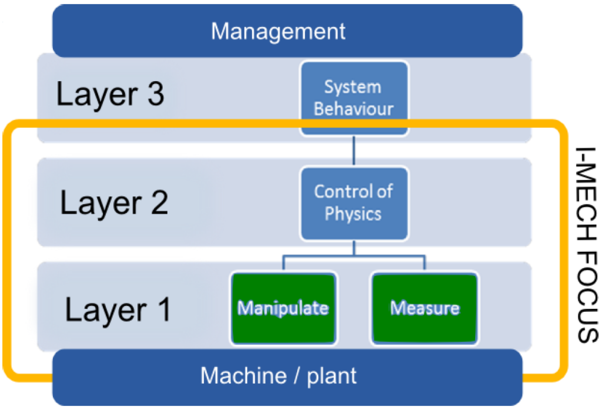


Figure 1.2: General structure of a control system.

The I-MECH Platform consists of a set of components which are referred as Building Blocks (BB). Every BB responds to a particular key objective of the I-MECH project. The list of BBs is outlined in Table 1.1. The BBs on which the University of Brescia was directly involved are BB6 (Self-commissioning velocity and position control loops), BB7 (Vibration control module) and BB9 (Iterative and repetitive control module). The structure of this thesis, as well as the methods for the validation of the techniques presented, is linked to this three BBs. As shown in Figure 1.3, all of this BBs are part of the Layer 2 (control).

1.4 Rapid Prototyping cycle

In order to fully prove the implementability and performance of the proposed solutions, the implemented technique needed to be pass Model-In-the-Loop (MIL), Simulation-In-the-Loop (SIL), Processor-In-the-Loop (PIL) and Hardware-In-the-loop (HIL) tests. Nonetheless, for practical reasons, some validations steps are often skipped, for example by testing the solution directly on HIL after the MIL validation.

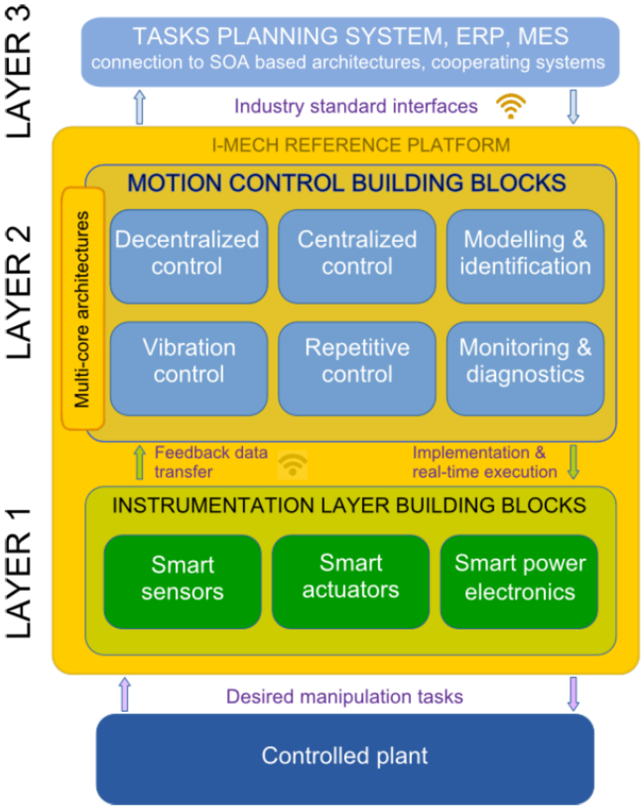


Figure 1.3: Decomposition of layers in a control system with I-MECH regions of interest.

BB1	Platform for Smart Sensors
BB2	Real-time wireless sensors
BB3	Robust condition monitoring and predictive diagnostics
BB4	High speed vision
BB5	High performance servo amplifier
BB6	Self-commissioning velocity and position control loops
BB7	Vibration control module
BB8	Robust model-based multivariable control
BB9	Iterative and repetitive control module
BB10	Control Specific Multi-many core Platform
BB11	RTOS for multi-many core platform

Table 1.1: I-MECH building blocks.

1.4.1 Model-In-the-Loop

In order to test the techniques on simulations, a combination of different software have been used along the project in order to run accurate models of both controllers and plants. While Mathworks Matlab/Simulink/Simscape has been acknowledged as a valid platform for software testing, the ability of other software (e.g., Siemens Amesim) to better describe certain dynamics has been exploited by means of co-simulation.

In Figure 1.4 the general scheme of the MIL simulation is shown. The controller is implemented and run on Matlab/Simulink and a model of the plant can run in Matlab/Simulink as well, or in co-simulation on another software, depending on the model accuracy required for the testing.

1.4.2 Software-In-the-Loop

In SIL testing, the control technique is compiled into code and the embedded software is tested within a simulated environment model, but without any

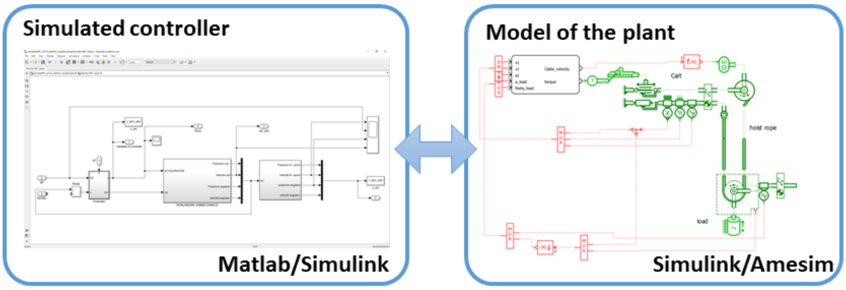


Figure 1.4: MIL scheme.

specific hardware.
As Gefran IDE does not support SIL testing nor co-simulation with other software, SIL testing has not been implemented: once the techniques have passed the MIL testing they were implemented on the industrial hardware and PIL testing was directly executed.

1.4.3 Processor-In-the-Loop

PIL simulation is essential part of prototyping where the control system is fully installed into the final hardware platform and can interact with the plant through the proper Inputs/Outputs. The machine model is running on the real-time system with I/O simulations (contains emulators of sensors and simulated actuators).
The techniques were implemented in IEC 61131-3 ST language and the program was downloaded into the drive motherboard along with simple models of the plants. This kind of testing is necessary to debug the translation of Matlab/Simulink controller into ST language and to check the implementability of the techniques in standard industrial hardware. A scheme of the PIL scheme is shown in Figure 1.5.

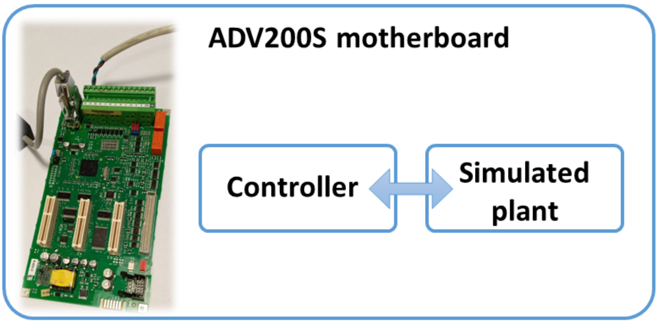


Figure 1.5: PIL scheme.

1.4.4 Hardware-In-the-Loop

HIL simulation has been demonstrated to be a powerful instrument for testing the software on industrial hardware before the test on the actual plant. This prevents the arising of errors from damaging the real plant [40].
The group of the University of Brescia, thanks to a strict collaboration with Gefran S.p.a., relied on a Hardware-In-the-Loop setup fully composed by industrial off-the-shelf hardware [62].
The HIL setup consists of two Gefran brushless servomotors (Gefran SBM series). The Motor Under Test (MUT), on which the control techniques are tested, has a rated nominal power of 1.51 [kW], while the motor simulating the plant dynamics (SM) has a rated nominal power of 3.77 [kW]. The two motors are controlled by two Gefran ADV200S inverter and controller units. The characteristics of the actuators are shown in Table 1.2. The HIL setup has been built in order to be flexible and easily reconfigurable. The two motors can be linked together by means of a rigid shaft or by a series of elastic transmissions composed by two belts and pulleys. The pulleys are linked together by means of a rigid shaft, but their diameter is different, so that a complete turn of the SM corresponds to a quarter of turn on MUT. Moreover, it is possible to change the inertia of the system composed by the two pulleys by adding a series of metal discs. This highly flexible and

	Motor Simulator	Motor Under Test
Type	SBM73303	SBM53303
Nominal power [kW]	3.77	1.51
Stall torque [Nm]	15.3	5.4
Stall current [A]	9.71	3.52
Nominal torque [Nm]	12	4.8
Nominal current [A]	7.62	3.13
Nominal voltage [V]	318	310
Nominal velocity [rpm]	3000	3000
Torque constant [Nm/A]	1.58	1.53
Poles	8	8

Table 1.2: Characteristics of the motors of the HIL setup.

easily reconfigurable setup can be used to simulate the behaviour of a great variety of mechanical systems. Figure 1.6 shows the mechanical hardware setup used for HIL validation. The University of West Bohemia provided an industrial PC running their proprietary *REXYGEN* software [61]. This industrial PC was used as a gateway between the host PC and the drives using EtherCAT standard and ethernet. This allowed for a faster communication and an easier and more powerful data acquisition overcoming the limits of Modbus communication. A scheme of the setup is shown in Figure 1.7.

1.4.4.1 HIL setup dynamics

Even if the reconfigurability of the system allows for the introduction of physical inertia and elasticities by including pulleys and belts, it is also possible to simulate the whole plant by using only the SM.
In order to effectively simulate the dynamic of the plant, the dynamic of the HIL setup must be known in order to compensate for it. In particular,

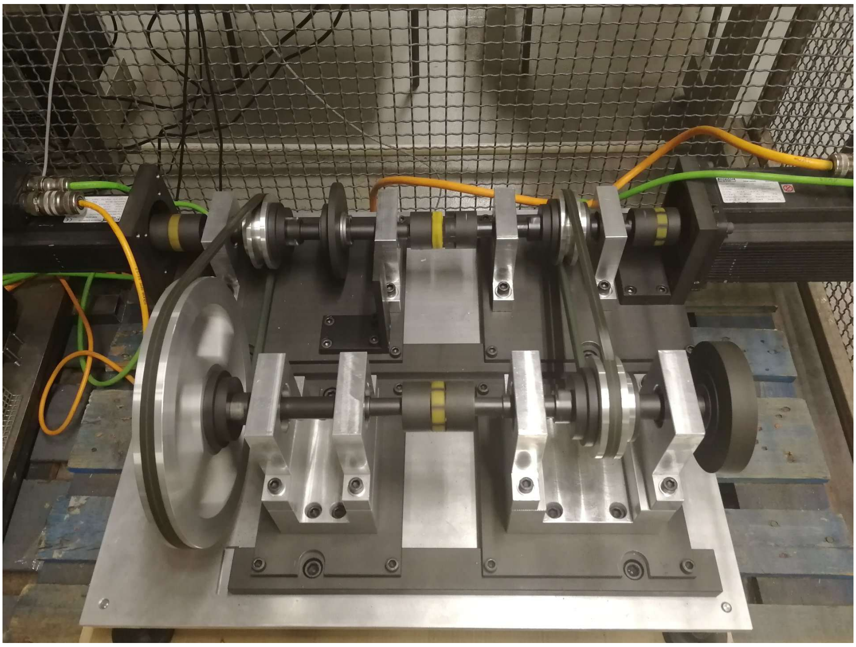


Figure 1.6: Mechanical setup of the HIL.

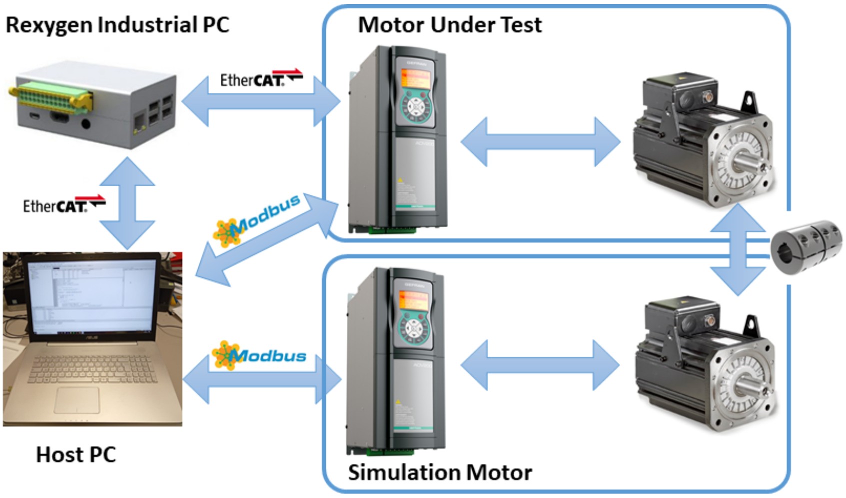


Figure 1.7: HIL setup scheme.

the rigid configuration can be approximated with a motor and load system so that the parameters that describe its dynamics are the total inertia of the system J_{tot} [kgm^2], the dynamic friction B_m [Nm/rad] and the static friction K_f [Nm]. The system of differential equations that describes the dynamics of the system is

$$\begin{cases} J_T\dot{\omega}(t) = \tau_m(t) + \tau_s(t) - B_m\omega(t) - K_f\text{sign}(\omega(t)) \\ \dot{\phi}(t) = \omega(t) \end{cases} \tag{1.1}$$

where ω [rad/s] is the angular velocity, ϕ [rad] the angular position, τ_m [Nm] and τ_s [Nm] the MUT and SM torques respectively. The estimation of these parameters as well as the methods for the compensation of the HIL setup dynamics will be presented in detail in Chapter 3.

Chapter 2

Control of industrial cranes

In this chapter a set of methods for the control of industrial cranes will be presented. In particular, different techniques for both open-loop and closed-loop control of oscillating underactuated systems are proposed.

Industrial cranes, thanks to their capability of moving heavy payloads despite their relative simplicity and low cost, are by far the most diffused Material Handling Systems (MHS) in the industrial and construction fields. Their simple design allows for the motion of heavy loads over wide ranges in clustered spaces.

Nonetheless, cranes are underactuated oscillatory systems, and the oscillation that affect the load position is highly underdamped. This poses issues both in terms of performance and safety. For this reason, the control of industrial cranes has raised increasing interest in both academical and industrial research [1, 57].

While expert operators learn to cope with the oscillatory dynamics of the crane, it has been demonstrated that the use of simple control techniques improves the manoeuvre performance of human operated cranes [39].

In this chapter, different models of industrial cranes are described and discussed. The described models are then exploited for the definition of both open and closed-loop approaches for the control of industrial cranes.

The chapter is structured as follows: in Section 2.1 mathematical models for the most diffused types of industrial cranes are derived; Section 2.2 motivates the need of scientific research in the field of the control of industrial cranes and reviews the available literature on the subject; in Section 2.3 open-loop control techniques for oscillatory systems are presented and tested, while closed-loop approaches are covered in Section 2.4. Section 2.5

concludes the chapter.

2.1 Cranes models

In order to apply innovative control techniques to industrial cranes, different models of the system have been exploited.
A crane consists of a support mechanism, that can be either a trolley mounted on a girder (e.g. for portal cranes) or a boom (e.g. for rotary cranes) and a hoisting mechanisms composed of a hoisting line and a hook. Based on their kinematics, the most diffused industrial cranes can be classified in four main classes, as shown in Figure 2.1.

While cranes can be considered as relatively simple systems, they can be represented by very different models depending on the desired degree of accuracy and the aim of the control. In fact, accurate models of a crane must include nonlinearities in order to describe its dynamics, and very accurate models can also include the elasticity of the hoisting cable in the equations. Nonetheless, for some applications complex models are not only not needed, but would require a parameters estimation phase that is not trivial and should be rerun at every change in the crane layout. Moreover, the models can also differ because of the different inputs and outputs considered. For example, for completely autonomous cranes, the payload position reference must be tracked, so that, in order to apply a model-based control technique, the model of the crane should have the payload position as the output. On the contrary, for human operators manoeuvred cranes a velocity output of the model is the most intuitive and by far the most used. In this section the equations and models for the different kinds of industrial cranes will be presented with a focus on the linearizations and simplifications that are necessary for control purposes.

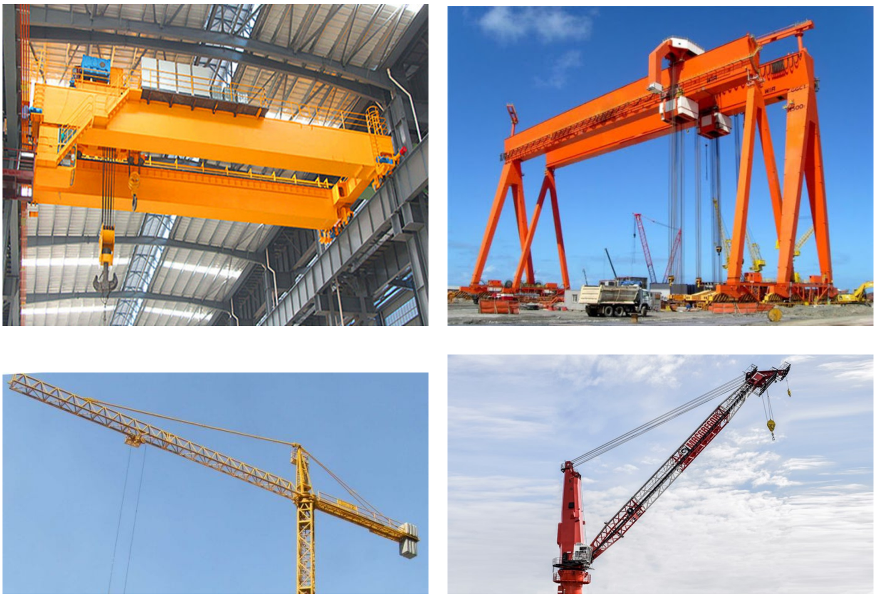


Figure 2.1: Industrial cranes: overhead crane (top left), portal crane (top right), rotary crane (bottom left) and boom crane (bottom right).

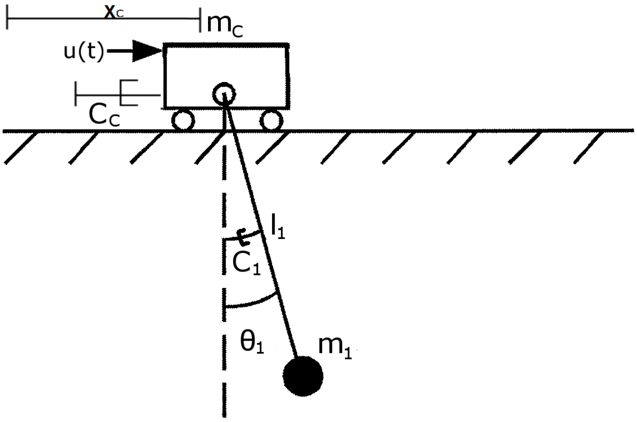


Figure 2.2: Overhead crane model.

m_p	Payload mass	[kg]
m_c	Cart mass	[kg]
l_1	Length of the hoisting line	[m]
C_c	Dynamic friction coefficient of the cart	[Ns/m]
C_1	Dynamic friction coefficient of the line	[Nms/rad]

Table 2.1: Parameters of the single pendulum model for overhead cranes.

2.1.1 Overhead and portal cranes as a single pendulum

Both overhead (or gantry) cranes and portal cranes can be modeled, for control purposes, as a single pendulum attached on a sliding cart, as shown in Figure 2.2. The assumption of motion on a bidimensional plane does not limit the extension to more general cases. The model is characterized by the parameters shown in Table 2.1. The system has a total of three degrees of freedom: the position of the cart x_c [m], the sway angle θ_1 [rad] and the length of the cable l [m]. Two motors can control the position of the cart and the length of the cable. As no actuator controls the sway angle,

the crane is indeed an underactuated system. The actuator controls the position of the cart by acting on the cart with a force u [N]. In this model the hook and the payload are considered as a single mass point.
Some approximations must be made in order to ensure a simple mathematical description of the model:

- the line is considered as an inflexible rod;
- compared to the mass of the payload, the mass of the cable can be neglected;
- the system moves only on a $x - y$ plan.

For the following dynamical model, a last assumption of static length of l have been made. This simplification allows for an easy linearization of the model. The effects of this assumption will be discussed in depth on the section dedicated to the control techniques.

2.1.1.1 Differential equations

The Lagrangian method can be used to find the differential equations of the system in Figure 2.2. The potential energy of the system is a function of the position of the payload, that is

$$P = m_p g l_1 (1 - \cos\theta_1) \tag{2.1}$$

where g [m/s^2] is the gravitational constant.
The kinetic energy of the system is given by

$$K = \frac{1}{2} m_c \dot{x}_c^2 + \frac{1}{2} m_p v_1^2 \tag{2.2}$$

where $\dot{x}_c$ is the velocity of the cart and v_1 is the velocity of the payload, that is

$$v_1^2 = v_{1x}^2 + v_{1y}^2 \tag{2.3}$$

where

$$v_{1x} = \dot{x}_c + l_1\dot{\theta}_1\cos(\theta_1) \quad \text{and} \quad v_{1y} = -l_1\dot{\theta}_1\sin(\theta_1) \tag{2.4}$$

Given (2.1) and (2.2), the Lagrangian of the system is

$$L = K - P = \frac{1}{2}m_c\dot{x}_c^2 + \frac{1}{2}m_p l_1 v^2 - m_p g l_1(1 - cos(\theta_1)) \tag{2.5}$$

The equations of the system dynamics can be calculated using the Lagrangian method, that states that

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{2.6}$$

where q_i, $i = [1...N]$, are the variables that represent the free ways of moving of the system and Q_i is the vector of the generalized external forces, which includes the dissipative forces through Rayleigh dissipation function [29].

The two variables that describe the free ways of moving of the system are x_c and θ_1. Differentiating L with respect to $\dot{x}_c$ in (2.5) we find

$$\frac{\partial L}{\partial \dot{x}_c} = m_c\dot{x}_c + m_p(\dot{x}_c + l_1\dot{\theta}_1\cos\theta_1) \tag{2.7}$$

Differentiating (2.7) with respect to time t yields

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_c}\right) = m_c\ddot{x}_c + m_1(\ddot{x}_c + l_1\ddot{\theta}_1\cos\theta_1 - l_1\dot{\theta}_1^2\sin\theta_1) \tag{2.8}$$

The potential energy of the system does not depend on x_c, therefore we have

$$\frac{\partial L}{\partial x_c} = 0 \tag{2.9}$$

The generalized external forces for the variable x_c depend on the external force $u(t)$ and the frictional force, thus

$$Q_x = u(t) - C_C\dot{x}_c \tag{2.10}$$

In the same way, differentiating (2.5) with respect to $\dot{\theta}_1$ yields

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_1} &= m_1[(\dot{x}_c + l_1\cos\theta_1)l_1\cos\theta_1 + (-l_1\dot{\theta}_1\sin\theta_1)(-l_1\dot{\theta}_1\sin\theta_1)] \\ &= m_1 l_1 \dot{x}_c\cos\theta_1 + m_1 l_1^2\dot{\theta}_1\end{aligned} \tag{2.11}$$

and differentiating (2.11) with respect to time we obtain

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = m_1 l_1 \ddot{x}_c\cos\theta_1 - m_1 l_1 \dot{x}_c\dot{\theta}_1\sin\theta_1 + m_1 l_1^2\ddot{\theta}_1 \tag{2.12}$$

Further, differentiating (2.5) with respect to θ_1 can have

$$\begin{aligned}\frac{\partial L}{\partial \theta_1} &= m_1[(\dot{x}_c + l_1\dot{\theta}_1\cos\theta_1)(-l_1\sin\theta_1) + (l_1\dot{\theta}_1\sin\theta_1)(l_1\dot{\theta}_1\cos\theta_1)] \\ &- m_1 g l_1\sin\theta_1 = -m_1 l_1\dot{x}_c\dot{\theta}_1\sin\theta_1 - m_1 g l_1\sin\theta_1\end{aligned} \tag{2.13}$$

Finally, generalized external forces for the variable θ_1 depends on the frictional force, thus

$$Q_{\theta_1} = -\frac{C_1}{l_1}\dot{\theta}_1 \tag{2.14}$$

Considering (2.8), (2.9) and (2.10) and (2.12), (2.13) and(2.14), from (2.6) the two equations that describe the dynamics of the system are

$$\begin{cases}(m_1 + m_C)\ddot{x}_c + m_1 l_1(\ddot{\theta}_1\cos\theta_1 - \dot{\theta}_1^2\sin\theta_1) = u(t) - C_C\dot{x}_c \\ m_1\ddot{x}_c\cos\theta_1 + m_1 l_1\ddot{\theta}_1 + m_1 g\sin\theta_1 = -\dfrac{C_1}{l_1}\dot{\theta}_1\end{cases} \tag{2.15}$$

2.1.1.2 Linearized dynamic model

The system of differential equations (2.15) can be linearized for control purposes, so that it can be expressed in a linear state space form. Under the hypothesis of small angles, the following approximations can be done:

$$\begin{aligned}\cos\theta_1 &\approxeq 1 \\ \sin\theta_1 &\approxeq \theta_1 \\ \dot{\theta}_1^2 &\approxeq 0,\end{aligned} \tag{2.16}$$

so that the system in (2.15) becomes

$$\begin{cases} (m_1 + m_C)\ddot{x}_c + m_1 l_1 \ddot{\theta}_1 + C_C \dot{x}_c = u(t) \\ m_1 \ddot{x}_c + m_1 l_1 \ddot{\theta}_1 + \dfrac{C_1}{l_1}\dot{\theta}_1 + m_1 g \theta_1 = 0. \end{cases} \tag{2.17}$$

The system (2.17) can be expressed in a well-known form for mechanical compliant systems as

$$\mathcal{M}\ddot{\boldsymbol{x}} + \mathcal{C}\dot{\boldsymbol{x}} + \mathcal{K}\boldsymbol{x} = \boldsymbol{F} \tag{2.18}$$

where

$$\mathcal{M} = \begin{pmatrix} m_1 + m_C & m_1 l_1 \\ m_1 & m_1 l_1 \end{pmatrix}, \mathcal{C} = \begin{pmatrix} C_C & 0 \\ 0 & \dfrac{C_1}{l_1} \end{pmatrix}, \mathcal{K} = \begin{pmatrix} 0 & 0 \\ 0 & m_1 g \end{pmatrix} \tag{2.19}$$

and where

$$\boldsymbol{x} = \begin{pmatrix} x_c \\ \theta_1 \end{pmatrix}, \boldsymbol{F} = \begin{pmatrix} u(t) \\ 0 \end{pmatrix}. \tag{2.20}$$

The linear representation in (2.18) can be rearranged in order to express the dynamics of the system in the form of the state space description, which is a standard description in the control field. Defining

$$\boldsymbol{x_{ss}} = \begin{pmatrix} x_c \\ \dot{x}_c \\ \theta_1 \\ \dot{\theta}_1 \end{pmatrix} \tag{2.21}$$

the state-space representation of the system (2.18) is in the form

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{ss}}(t) = \mathcal{A}\boldsymbol{x_{ss}}(t) + \mathcal{B}u(t) \\ \boldsymbol{y}(t) = \mathcal{C}\boldsymbol{x_{ss}}(t), \end{cases} \tag{2.22}$$

where

$$\mathcal{A}=\begin{pmatrix}0 & 1 & 0 & 0\\ 0 & -\dfrac{C_C}{m_C} & \dfrac{gm_1}{m_C} & \dfrac{C_1}{l_1 m_C}\\ 0 & 0 & 0 & 1\\ 0 & \dfrac{C_C}{l_1 m_C} & -\dfrac{g(m_1+m_C)}{l_1 m_C} & -\dfrac{C_1(m_1+m_C)}{l_1^2 m_1 m_C}\end{pmatrix},\mathcal{B}=\begin{pmatrix}0\\ \dfrac{1}{m_C}\\ 0\\ -\dfrac{1}{l_1 m_C}\end{pmatrix} \tag{2.23}$$

Defining the output of the system as the position of the load with respect to a fixed frame, it is given by

$$x_p = x_c + l_1 \sin\theta \tag{2.24}$$

Under the assumption of small angles, the equation (2.24) can be approximated as

$$x_p = x_c + l_1\theta \tag{2.25}$$

as $\sin\theta \approxeq \theta$.

The matrix of the outputs $\mathcal{C}$ in (2.22), considering the velocity of the payload as the output of the system, is therefore

$$\mathcal{C}=\begin{pmatrix}0 & 1 & 0 & l_1\end{pmatrix} \tag{2.26}$$

2.1.1.3 Kinematic model

Another way of modeling the overhead crane is to describe the dynamics between the kinematic of the cart and the kinematic of the load. In fact, by taking the acceleration of the cart as the input, the system is characterized by the following system of nonlinear differential equations:

$$\dot{\boldsymbol{x}}_{\boldsymbol{nl}}=\begin{bmatrix}\dot{x}_c\\ 0\\ 0\\ \dot{\theta}\\ -\frac{g\sin(\theta)}{l_1}\end{bmatrix}+\begin{bmatrix}0\\ 1\\ 0\\ 0\\ -\frac{\cos(\theta)}{l_1}\end{bmatrix}\ddot{x}_c+\begin{bmatrix}0\\ 0\\ 1\\ 0\\ -\frac{2\dot{\theta}}{l_1}\end{bmatrix}\dot{l_1} \tag{2.27}$$

where $x_{nl} = [x_c, \dot{x}_c, l_1, \theta, \dot{\theta}]^T$ is the state vector and the velocity of the payload is given by $y_{nl} = \dot{x}_c + \dot{\theta} l_1 \cos\theta + \dot{l}_1 \sin\theta$.

This nonlinear model can effectively take into account also the contribution of the lengthening and shortening of the hoisting cable. If the length cable is, again, considered as constant, the model can be linearized with the assumption of small oscillation angles, so that the system can be described again in a state space form like

$$\begin{aligned} \dot{x}_l &= \mathcal{A} x_l + \mathcal{B} \ddot{x}_c \\ y_l &= \mathcal{C} x_l \end{aligned} \tag{2.28}$$

where the state is $x_l = [x_c, \dot{x}_c, \theta, \dot{\theta}]^T$ and the state space matrices are

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{g}{l_1} & 0 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{l_1} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ l_1 \end{bmatrix}^T. \tag{2.29}$$

2.1.1.4 Useful transfer functions

In the following part, in particular when introducing input-output inversion control, the transfer function describing the dynamics of overhead cranes will assume significant importance.

The most important transfer functions regarding overhead cranes are the one that link the force acting on the cart and the velocity/position of the payload, and the one linking the velocity of the cart to the velocity of the payload.

The transfer function which takes as input the force acting on the cart and has as output the payload position is easily obtained from (2.22), remembering that the transfer function can be obtained from the state space model as

$$G(s) = \mathcal{C}(s\mathcal{I} - \mathcal{A})^{-1}\mathcal{B} + \mathcal{D} \tag{2.30}$$

obtaining

$$G(s) = \frac{X_p(s)}{U(s)} = \frac{N_G(s)}{D_G(s)} \tag{2.31}$$

where

$$N_G(s) = C_1 s + g m_1 l_1^2$$

and

$$\begin{aligned} D_G(s) =&(l_1^2 m_1 m_C)s^4 + (C_c m_1 l_1^2 + C_1 m_1 + C_1 m_C)s^3 + \\ &+ (g l_1^2 m_1^2 + g m_C l_1^2 m_1 + C_1 C_c)s^2 + (C_c g l_1^2 m_1)s. \end{aligned}$$

The transfer function with the force on the cart as the input and the velocity of the payload as output is easily obtained by differentiating (2.31), that is,

$$\frac{\dot{X}_p(s)}{U(s)} = \frac{X_p(s)}{U(s)} s = sG(s). \tag{2.32}$$

The transfer function between the velocity of the cart and the velocity of the payload can be obtained, with the same procedure, from (2.28), obtaining

$$\frac{\dot{X}_p(s)}{\dot{X}_c(s)} = \frac{g m_1 l_1^2 + C_1 s}{l_1^2 m_1 s^2 + C_1 s + g l_1^2 m_1}. \tag{2.33}$$

It should be noted that (2.33) is valid also to describe the dynamics between the position of the cart and the position of the payload, that is

$$\frac{X_p(s)}{X_c(s)} = \frac{\dot{X}_p(s)}{\dot{X}_c(s)}. \tag{2.34}$$

2.1.2 Overhead crane as a double pendulum

In many cases, approximating the dynamics of the overhead crane with a simple pendulum on a sliding cart is not sufficiently accurate. In the case of distributed inertia payloads or when the hook mass cannot be neglected, the dynamics of the overhead crane can be better described using a double

m_C	Cart mass	[kg]
m_1	First load mass	[kg]
m_2	Second load mass	[kg]
l_1	Length of the hoisting line	[m]
l_2	Length of the cable between loads	[m]
C_c	Dynamic friction coefficient of the cart	[Ns/m]
C_1	Dynamic friction coefficient of the cable l_1	[Nms/rad]
C_2	Dynamic friction coefficient of the cable l_2	[Nms/rad]

Table 2.2: Parameters of the double pendulum model for overhead cranes.

pendulum on a sliding cart, with two different masspoint bodies. This allows for a more accurate description of the dynamics in such cases where multiple oscillations are present in the dynamics of the crane.
The model can be schematized as in Figure 2.3, where x_c is the position of the cart, m_C, m_1 and m_2 the masses of the cart, the first and the second loads respectively, C_C, C_1 and C_2 the viscous friction coefficients of the cart, of the first and of the second cable, θ_1 and θ_2 are the inclinations of the cables with respect to the vertical and x_P is the position of the second load. The parameters of the model are resumed in Table 2.2.

2.1.2.1 Differential equations

Under the assumptions of lines considered as inflexible rods, concentrated axes and movement constrained to the $x - y$ plan, through the Lagrangian method used in Subsection 2.1.1.1, the system in Figure 2.3 can be described

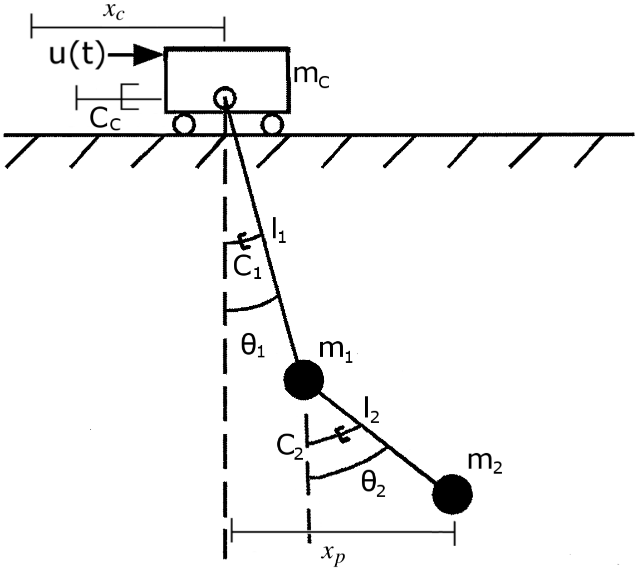


Figure 2.3: Scheme of an overhead crane modeled as a double pendulum on a sliding cart.

by the system of equations

$$\begin{cases} (m_C + m_1 + m_2)\ddot{x}_c + (m_1 + m_2)l_1\ddot{\theta}_1 \cos\theta_1 + m_2 l_2 \ddot{\theta}_2 \cos\theta_2 \\ -\,(m_1 + m_2)l_1\dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 = u(t) - C_C\dot{x}_c \\ (m_1 + m_2)l_1\ddot{x}_c \cos\theta_1 + (m_1 + m_2)l_1^2\ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 cos(\theta_1 - \theta_2) \\ +\, m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2) g l_1 \sin\theta_1 = -\dfrac{C_1}{l_1}\dot{\theta}_1 \\ m_2 l_2 \ddot{x}_c \cos\theta_2 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) \\ -\, m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 g l_2 \sin\theta_2 = -\dfrac{C_2}{l_2}\dot{\theta}_2. \end{cases} \tag{2.35}$$

2.1.2.2 Dinearized dynamic model

The system of differential equations (2.35) can be linearized around its stable equilibrium point, that is for $\theta_1 = 0$, $\theta_2 = 0$, $\dot{\theta}_1 = 0$ and $\dot{\theta}_2 = 0$. The obtained linear system can be written in the form

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{ss}}(t) = \mathcal{A}\boldsymbol{x}_{\boldsymbol{ss}}(t) + \mathcal{B}u(t) \\ \boldsymbol{x}_{\boldsymbol{p}}(t) = \mathcal{C}\boldsymbol{x}_{\boldsymbol{ss}}(t), \end{cases} \tag{2.36}$$

where the states vector $\boldsymbol{x}_{\boldsymbol{ss}}$ is composed by the cart position and velocity and the angular positions and velocities, that is

$$\boldsymbol{x}_{\boldsymbol{ss}} = \begin{pmatrix} x_c & \dot{x}_c & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 \end{pmatrix}^T \tag{2.37}$$

and

$$\mathcal{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{C_C}{m_C} & \frac{(m_1+m_2)g}{m_C} & \frac{C_1}{l_1^2 m_C} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{C_C}{l_1 m_C} & -\frac{g(m_1+m_2)(m_1+m_c)}{l_1 m_1 m_C} & -\frac{C_1(m_1+m_C)}{l_1^3 m_1 m_C} & \frac{g m_2}{l_1 m_1} & \frac{C_2}{l_1 l_2^2 m_1} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(m_1+m_2)}{l_2 m_1} & \frac{C_1}{l_1^2 l_2 m_1} & -\frac{g(m_1+m_2)}{l_2 m_1} & -\frac{C_2(m_1+m_2)}{l_2^3 m_1 m_2} \end{pmatrix}, \tag{2.38}$$

$$\mathcal{B} = \begin{pmatrix} 0 & \frac{1}{m_C} & 0 & -\frac{1}{l_1 m_C} & 0 & 0 \end{pmatrix}^T, \tag{2.39}$$

$$\mathcal{C} = \begin{pmatrix} 1 & 0 & l_1 & 0 & l_2 & 0 \end{pmatrix}. \tag{2.40}$$

2.1.2.3 Useful transfer functions

In what follows, in particular when introducing input-output inversion control, the transfer function describing the dynamics of overhead cranes will assume a significant importance.
From system (2.36) the transfer function of the system is determined as

$$F(s) := \frac{X_P(s)}{U(s)} = \frac{N_F(s)}{D_F(s)}, \tag{2.41}$$

where

$$\begin{aligned} N_F(s) =& (C_1 C_2)s^2 + (C_1 g m_2 l_2^2 + C_2(g l_1^2 m_1 + g l_1^2 m_2))s \\ &+ g^2 l_2^2 l_1^2 m_2 (m_1 + m_2) \end{aligned}$$

and

$$D_F(s) = p_6 s^6 + p_5 s^5 + p_4 s^4 + p_3 s^3 + p_2 s^2 + p_1 s,$$

where

$$\begin{aligned}
p_6 =& l_1^3 l_2^3 m_2 m_1 m_C, \\
p_5 =& C_1 l_2^3 m_1 m_2 + C_2 l_1^3 m_1 m_C + C_1 l_2^3 m_2 m_C + C_2 l_1^3 m_2 m_C \\
& + C_C l_1^3 l_2^3 m_1 m_2, \\
p_4 =& g m_C l_1^3 l_2^2 m_1 m_2 + g m_C l_1^3 l_2^2 m_2^2 + C_2 C_C l_1^3 m_1 + C_2 C_C l_1^3 m_2 \\
& + g l_1^2 l_2^3 m_1^2 m_2 + g l_1^2 l_2^3 m_1 m_2^2 + g m_C l_1^2 l_2^3 m_1 m_2 + g m_C l_1^2 l_2^3 m_2^2 \\
& + C_1 C_C l_2^3 m_2 + C_1 C_2 m_1 + C_1 C_2 m_2 + C_1 C_2 m_C, \\
p_3 =& + C_C g l_1^3 l_2^2 m_1 m_2 + C_C g l_1^3 l_2^2 m_2^2 + C_C g l_1^2 l_2^3 m_1 m_2 + C_C g l_1^2 l_2^3 m_2^2 \\
& + C_2 g l_1^2 m_1^2 + 2 C_2 g l_1^2 m_1 m_2 + C_2 g m_C l_1^2 m_1 + C_2 g l_1^2 m_2^2 \\
& + C_2 g m_C l_1^2 m_2 + C_1 g l_2^2 m_1 m_2 + C_1 g l_2^2 m_2^2 + C_1 g m_C l_2^2 m_2 \\
& + C_1 C_2 C_C, \\
p_2 =& g^2 l_1^2 l_2^2 m_1^2 m_2 + 2 g^2 l_1^2 l_2^2 m_1 m_2^2 + m_C g^2 l_1^2 l_2^2 m_1 m_2 + g^2 l_1^2 l_2^2 m_2^3 \\
& + m_C g^2 l_1^2 l_2^2 m_2^2 + C_2 C_C g l_1^2 m_1 + C_2 C_C g l_1^2 m_2 + C_1 C_C g l_2^2 m_2, \\
p_1 =& C_C g^2 l_1^2 l_2^2 m_2^2 + C_C m_1 g^2 l_1^2 l_2^2 m_2.
\end{aligned}$$

It is easy to demonstrate that the transfer function between the force on the cart and the payload velocity can be obtained by differentiating (2.41), that is

$$\frac{\dot{X}_P(s)}{U(s)} = s\frac{X_P(s)}{U(s)} = sF(s). \tag{2.42}$$

Since in typical industrial cranes only the position/speed of the motor (that is, of the cart) is measured, it is convenient to represent (2.41) as the product of two different transfer functions, namely

$$F(s) = \frac{X_P(s)}{U(s)} = P(s)\frac{1}{s}G(s), \tag{2.43}$$

where

$$P(s) := \frac{V_C(s)}{U(s)} = \frac{N_P(s)}{D_P(s)} \tag{2.44}$$

is the transfer function between the force applied to the cart and the velocity of the cart, where

$$N_P(s) = \lambda_4 s^4 + \lambda_3 s^3 + \lambda_2 s^2 + \lambda_1 s + \lambda_0,$$

where

$$\begin{aligned}
\lambda_4 &= \frac{l_1\, l_2\, m_1}{g^2\, (m_1 + m_2)},\\
\lambda_3 &= \frac{C_2\, l_1^3\, m_1 + C_1\, l_2^3\, m_2 + C_2\, l_1^3\, m_2}{g^2\, l_1^2\, l_2^2\, m_2\, (m_1 + m_2)},\\
\lambda_2 &= \frac{g\, l_1^3\, l_2^2\, m_2^2 + g\, m_1\, l_1^3\, l_2^2\, m_2 + g\, l_1^2\, l_2^3\, m_2^2 + g\, m_1\, l_1^2\, l_2^3\, m_2 + C_1\, C_2}{g^2\, l_1^2\, l_2^2\, m_2\, (m_1 + m_2)},\\
\lambda_1 &= \frac{C_2\, l_1^2\, m_1 + C_1\, l_2^2\, m_2 + C_2\, l_1^2\, m_2}{g\, l_1^2\, l_2^2\, m_2\, (m_1 + m_2)},\\
\lambda_0 &= 1,
\end{aligned}$$

and

$$D_P(s) = D_F(s), \tag{2.45}$$

and

$$G(s) := \frac{X_P(s)}{X_c(s)} = \frac{V_P(s)}{V_C(s)} = \frac{N_F(s)}{D_P(s)} \tag{2.46}$$

is the transfer function between the position (velocity) of the cart and the position (velocity) of the payload.

2.1.3 Boom crane model

In this work, antisway techniques have also been applied to boom cranes. Due to the rotational degree of freedom of boom cranes, their dynamics and kinematic are better described using the polar coordinates shown in Figure 2.4. In general, boom cranes have five degrees-of-freedom (DOF) which, referring to Figure 2.4, are the rotation angle α, the pitching angle β, the

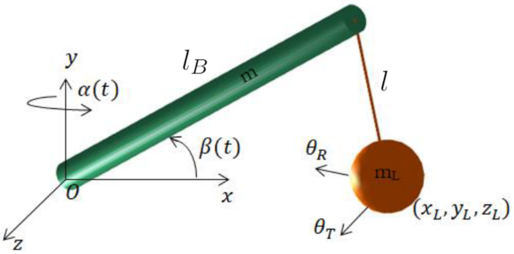


Figure 2.4: Scheme of a boom crane.

length of the cable l, and the tangential and radial sway angles, θ_T and θ_R respectively. While rotation, pitching and hoisting are actuated DOF, the sway angles cannot be directly controlled, so that the boom crane can be classified as an underactuated system. The dynamics of the system requires also the following parameters:

- l_B, the length of the boom arm
- m, the mass of the boom arm
- m_L, the payload mass.

For most industrial applications, for constructive reasons, the length of the hoisting cable is directly related to the pitching angle β, so that if the hoisting actuator is not activated, a change in the pitching does not affect the height of the payload, thus affecting l.

2.1.3.1 Differential equations

In order to obtain manageable dynamics equations, the following assumptions on the boom crane have to be made:

- the mass of the rope and of the hook supporting the payload have been neglected;

- the inertia of the payload has been neglected, and the payload has been considered as a point mass;
- the rope is considered as an inextensible cable;
- the frame of the boom crane is considered as infinitely rigid;
- frictions and dampenings are not considered, as they have very small influence on the crane dynamics.

Under these assumptions, the differential equations describing the dynamics of the boom crane can be obtained following the same procedure used for the single and double pendulum overhead cranes, that is by defining the Euler–Lagrange equations. The position of the payload is given by

$$\begin{cases} x_L(t) = & l_B \cos\beta(t) \cos\alpha(t) + l(t)(\cos\theta_R(t) \sin\theta_T(t) \sin\alpha(t) \\ & - \sin\theta_R(t) \cos\alpha(t)) \\ y_L(t) = & l_B \sin\beta(t) - l(t) \cos\theta_R(t) \cos\theta_T(t) \\ z_L(t) = & -l_B \cos\beta(t) \sin\alpha(t) - l(t)(\cos\theta_R(t) \sin\theta_T(t) \cos\alpha(t) \\ & - \sin\theta_R(t) \sin\alpha(t)). \end{cases} \tag{2.47}$$

By applying the Lagrangian method, using the sway angles θ_R and θ_T as variables, and by applying the small angles approximations

$$\sin(\theta_T) = \theta_T \ , \ \cos(\theta_T) = 1 \ , \ \sin(\theta_R) = \theta_R \ , \ \cos(\theta_R) = 1 \tag{2.48}$$

the dynamics of the boom crane can be described by the equations

$$\begin{cases} \ddot{\theta}_T + 2\frac{\dot{l}}{l}\dot{\theta}_T + (\frac{g}{l} - \dot{\alpha}^2 - \frac{l_B}{l} sin(\beta)\dot{\beta}^2 + \frac{l_B}{l} cos(\beta)\ddot{\beta})\theta_T \\ + (\ddot{\alpha} + 2\dot{\alpha}\frac{\dot{l}}{l})\theta_R = \frac{l_B}{l}\left(2\sin(\beta)\dot{\alpha}\dot{\beta} - \cos(\beta)\ddot{\alpha}\right) \\ \\ \ddot{\theta_R} - 2\dot{\alpha}\dot{\theta}_T + 2\frac{\dot{l}}{l}\dot{\theta}_R + \left(\frac{g}{l} - \dot{\alpha}^2 - \frac{l_B}{l} sin(\beta)\dot{\beta}^2 + \frac{l_B}{l} cos(\beta)\ddot{\beta}\right)\theta_R \\ - \left(\ddot{\alpha} + 2\dot{\alpha}\frac{\dot{l}}{l}\right)\theta_T = \frac{l_B}{l}\left(\cos(\beta)\dot{\alpha}^2 + \cos(\beta)\dot{\beta}^2 + \sin(\beta)\ddot{\beta}\right) \end{cases} \tag{2.49}$$

2.1.3.2 Dynamic model

The system in (2.49) can be rewritten in matrix form as

$$\mathcal{M}\ddot{\boldsymbol{\Theta}} + \mathcal{C}\dot{\boldsymbol{\Theta}} + \mathcal{K}\boldsymbol{\Theta} = \boldsymbol{U} \tag{2.50}$$

where

$$\boldsymbol{\Theta} = \begin{pmatrix} \theta_T \\ \theta_R \end{pmatrix}$$

$$\boldsymbol{U} = \begin{pmatrix} \frac{l_B}{l}\left(2\sin(\beta)\dot{\alpha}\dot{\beta} - \cos(\beta)\ddot{\alpha}\right) \\ \frac{l_B}{l}\left(2\cos(\beta)\dot{\alpha}^2 + \cos(\beta)\dot{\beta}^2 + \sin(\beta)\ddot{\beta}\right) \end{pmatrix}$$

$$\mathcal{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathcal{C} = 2\begin{pmatrix} \frac{\dot{l}}{l} & \dot{\alpha} \\ \dot{\alpha} & \frac{\dot{l}}{l} \end{pmatrix}$$

$$\mathcal{K} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

with

$$\begin{aligned}
K_{11} =& \frac{g}{l} - \dot{\alpha}^2 - \frac{l_B}{l}\sin(\beta)\dot{\beta}^2 + \frac{l_B}{l}\cos(\beta)be\ddot{t}a \\
K_{12} =& \left(\ddot{\alpha} + 2\frac{\dot{l}}{l}\dot{\alpha}\right) \\
K_{21} =& -\left(\ddot{\alpha} + 2\frac{\dot{l}}{l}\dot{\alpha}\right) \\
K_{22} =& \frac{g}{l} - \dot{\alpha}^2 - \frac{l_B}{l}\sin(\beta)\dot{\beta}^2 + \frac{l_B}{l}\cos(\beta)\ddot{\beta}.
\end{aligned}$$

The system in (2.50) can be linearized under the following assumptions:

- small values of $\dot{\alpha}$, $\dot{\beta}$ and $\dot{l}$;
- small values of $\ddot{\alpha}$, $\ddot{\beta}$ w.r.t. $\frac{g}{l}$.

Under these assumptions, (2.50) can be simplified as

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \ddot{\mathbf{\Theta}} + \begin{pmatrix} \frac{g}{l} & 0 \\ 0 & \frac{g}{l} \end{pmatrix} \mathbf{\Theta} = \begin{pmatrix} -\frac{l_B}{l}\cos(\beta)\ddot{\alpha} \\ \frac{l_B}{l}\sin(\beta)\ddot{\beta} \end{pmatrix} \tag{2.51}$$

2.2 Cranes control: state-of-the-art

As reported in [1, 57], a first differentiation between the proposed control techniques can be firstly done by dividing the techniques on open-loop and closed-loop approaches.

Open-loop approaches are by far the most diffused for the control of industrial cranes. Their approach is to devise a motion trajectory for the cart in order to obtain an efficient motion of the payload, i.e. with low (optimally null) residual oscillations. This approach is also the most appreciated in industry for a number of reasons:

- it avoids the need for sensors for the measure of the oscillation angles, which are hardly available in industrial off-the-shelf-cranes;
- the motion planning of the cart is easily implementable on standard industrial hardware as, being only based on a predefined model of the system, all the cart trajectories can be easily computed offline, avoiding the computational burden of online optimizations;
- their implementation is fully compatible with the standard velocity control loops, which are already present in industrial cranes.

The most widespread open-loop solution for the control of oscillating systems is the input shaping [20, 56, 63, 64, 65, 66, 67, 74], which is discussed in detail in Section 2.3.1. This technique consists in modifying the trajectory of the cart by convolving it with a filter composed by impulses, which amplitudes and times are computed so that the sum of the responses generated by each of them on the systems result in a perfect cancellation of the oscillatory dynamic on the total response, thanks to the superposition principle valid for linear systems. The research on input shaping is still active, and in the last 10 years various modifications to the approach have been proposed [6, 7, 45, 59].

Another open-loop technique proposes to solve one of the drawbacks of input shaping. Input shaping filters introduce a delay between input and output which is directly linked to the oscillating periods of the system. While, depending on the desired robustness, the operator can hypothetically choose between a wide set of input shaping filters, the choice is not supported by parameters with physical meaning, so that non-expert operators might prefer to not modify the default filter, resulting in sub-optimal performance in terms of positioning time when the filter is too conservative, and in excessive residual oscillations when the robustness is lacking. Input-output inversion [51, 54, 55], as presented in Section 2.3.2, is an open-loop technique based on the inversion of the model of the system. By defining a desired trajectory of the system, the corresponding input can indeed be found by inverting the transfer function of the system. Input-output inversion presents as its only tuning parameter the transition time τ required to reach the steady state. Having an immediate physical meaning, the transition time can be easily tuned also by non-expert operators.

A further possible solution for the open-loop control of industrial cranes is the use of a band-stop filters on the input of the system in order to cancel the frequency components of the input that would excite the vibration modes of the system. Various publications have proposed different approaches, which have been demonstrated to be effective in particular with rotary cranes [42, 52, 53].

While open-loop techniques are simple to implement and do not require additional sensors for being used in industrial cranes, they typically lack in terms of robustness. In fact, cranes are used to move a variety of loads which differs in terms of mass, inertia and anchoring mechanism, so that a precise model of the system is hard to obtain, and open-loop techniques are particularly subject to errors in the model parameters.
For this reason, closed-loop techniques have been proposed for the control of industrial cranes. With the lowering of the price of industrial sensors and the development of new and more performing communication standards, the sensorization of cranes for the measure of the sway angle is an issue which is more easily overcome than some years ago. Moreover, even if most closed-loop approaches require a knowledge of all the states of the system,

it is not necessary to have them all measured, as state observer can be easily used for their reconstruction.

An effective approach for the closed-loop control of overhead cranes is full state feedback control. With this approach, it is possible to define the dynamics of the states of the closed-loop system so that the generated oscillations are rapidly damped by simply defining the eigenvalues of the closed-loop dynamic matrix. In order to use full state feedback approaches, observers are typically used for reconstructing unmeasured states [38]. The application of full state feedback to overhead cranes is discussed in Section 2.4.1

Another closed-loop control that is worth addressing is Model Predictive Control (MPC). Due to its high computational requirements, MPC has been initially limited to process control, where large time constants allowed the controller to solve complex optimization problems that took time to be solved. With the increasing in computational capability of modern control hardware, optimization problems can be solved online also when controlling mechatronic systems, for which the control cycle time is in the range of the milliseconds.
While most of the MPC techniques for the control of industrial cranes proposed in the literature are focused on the position control of industrial cranes, their use is limited to fully automated cranes, for which the end-position of a manoeuvre is known in advance. Section 2.4.2 presents innovative and practical MPC approaches for the control of operated manoeuvre ed overhead cranes.

The aforementioned academic publications propose very different approaches to the problem of antisway control of cranes, but few of them have been validated on industrial systems, and even fewer are actually available as commercial solutions.
ABB provides a set of drives and PLCs dedicated to the control of industrial cranes, equipped with antisway libraries based on the input shaping approach and mechanical brake control features. No specific solution is provided for the closed-loop control of cranes.

Danfoss commercial products provide open-loop antisway control features, based again on input shaping, a variety of solutions for the safety (mechanical brake, functional safety), energy saving (regenerative power), but standard antisway closed-loop control solutions are not available.
Siemens provides a dedicated product, SIMOCRANE, which is available in five different versions, each one with different functionalities. Antisway closed-loop control features can be unlocked by mounting a camera on the trolley and a reflector on the hook.
Schneider electric provides libraries for the implementation of basic input shaping zero vibration solutions, but closed-loop control is not addressed.
Gefran s.p.a. provides a dedicated firmware for its ADV200 drive for "Hoist and cranes" applications, with full management of the mechanical brake and an antisway solution based on Zero Vibration input shaping.
Vinati S.r.l. provides an inertial platform for the measure of the oscillation and tilting of the load, together with closed-loop approaches for their compensation.

From the comparison between academic and industrial state-of-the-art, it is clear that a gap exists between academic solutions and commercially available products. For example, the proposed advanced closed-loop approaches such as Model Predictive Control is not available on the market as a solution for the antisway control of cranes. Advanced solutions would provide, as presented in various scientific publications, a way to respect hard constraints while reducing energy consumption, thus bringing the market to the cutting edge when existent academical solutions will be brought to commercial products.

Consistently with the goal of this thesis, the following chapters will present the details of a number of anti-sway control techniques for cranes, both new and already present in the literature, and the implementability of the presented techniques on industrial systems will be addressed, with the aim of bridging the gap existing between academic and commercially available industrial solutions.

Table 2.3: Parameters of the model used in simulations.

Model data			
m_C	100 $[kg]$	C_C	0.1 $[\frac{Ns}{m}]$
m_1	10 $[kg]$	C_1	3 $[\frac{Nms}{rad}]$
m_2	10 $[kg]$	C_2	3 $[\frac{Nms}{rad}]$
l_1	3 $[m]$	l_2	3 $[m]$

2.3 Open-loop control

In this section, open-loop approaches for the control of cranes will be presented, while closed-loop approaches are treated in Section 2.4.

The control of a pendulum is often used during control academic courses as an example for the control of oscillating systems. In fact, being the pendulum a highly underdamped underactuated system, it is a perfect example for the control of a system with strong oscillatory behaviours, while allowing for a simple and straightforward description of its dynamics [21].
As industrial cranes can be modelled basically as pendulums on a moving cart, the literature regarding their control is vast and it focuses mainly on open-loop approaches. Among proposed techniques, the most common approaches can be resumed in three main categories: Input Shaping, Notch filtering and Input-Output Inversion.

The various methodologies have been validated on a double pendulum model of an overhead crane, whose parameters are shown in Table 2.3. For the case of MIL validation, the model has been built with Simulink Simscape Multibody. For the HIL validation, the SM of the hardware setup described in Chapter 1.4.4 simulates the dynamic response of the double pendulum overhead crane.

2.3.1 Input shaping

Input shaping is the most widespread control techniques for industrial cranes. Conceived in the late '50 [9, 69] and then formalized in [63], input shaping is a technique which is based on the superposition principle for linear systems. Basing on the evidence that, for linear systems, the total response to a sum of a series of inputs is equal to the sum of the single responses of the system corresponding to each input, input shaping filters the input signal that would generate an oscillatory response in order to obtain two (or more) oscillatory responses, which summed cancel each other.

The impulse response of an oscillatory system can be described in the time domain by

$$y(t) = \left[A \frac{\omega_n}{\sqrt{1-\xi^2}} \exp^{\omega_n (t-t_0)} \right] \sin(\omega_n \sqrt{1-\xi^2}(t-t_0) + \phi) \tag{2.52}$$

where A is the initial amplitude of the oscillation, ω_n the natural frequency of the oscillation, ξ is the damping coefficient and ϕ is the phase. An example of impulse response is shown in Figure 2.5. The sum of the responses to two different impulses, thanks to the superposition principle for linear systems, can be written as

$$\begin{aligned} y_t(t) &= B_1 \sin(\alpha(t-t_0) + \phi_1) + B_2 \sin(\alpha(t-t_0) + \phi_2) \\ &= A_t \sin(\alpha t + \phi_t) \end{aligned} \tag{2.53}$$

where

$$B_i = A_i \frac{\omega_n}{\sqrt{1-\xi^2}} \exp^{\omega_n (t-t_i)} \quad , \quad \alpha = \omega_n \sqrt{1-\xi^2}.$$

The amplitude and phase of the resulting oscillatory response are given by

$$\begin{aligned} A_t &= \sqrt{(B_1 \cos\phi_1 + B_2 \cos\phi_2)^2 + (B_1 \sin\phi_1 + B_2 \sin\phi_2)^2} \\ \phi_t &= tan^{-1} \left(\frac{B_1 \cos\phi_1 + B_2 \cos\phi_2}{B_1 \sin\phi_1 + B_2 \sin\phi_2} \right) \end{aligned} \tag{2.54}$$

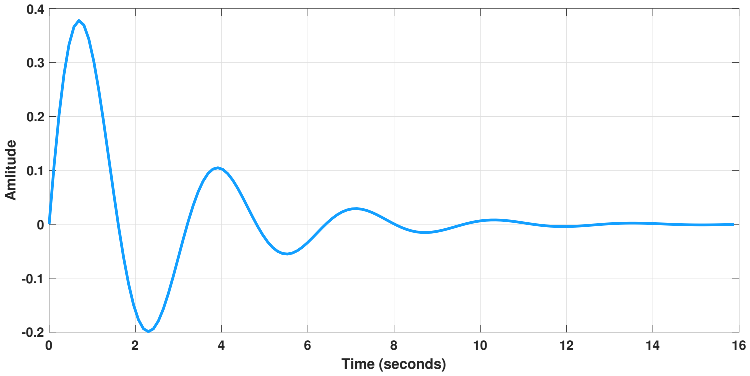


Figure 2.5: Impulse response of an oscillatory system, as defined by (2.52), with parameters $A = 1$, $\omega_n = 2$, $\xi = 0.2$, $\phi = 0$.

Generalizing for the N impulses, the amplitude and phase of the resulting response are given by

$$\begin{aligned} A_t &= \sqrt{(\sum_{i=1}^{N} B_i \cos\phi_i)^2 + (\sum_{i=1}^{N} B_i \sin\phi_i)^2} \\ \phi_t &= tan^{-1}\left(\frac{\sum_{i=1}^{N} B_i \cos\phi_i}{\sum_{i=1}^{N} B_i \sin\phi_i}\right). \end{aligned} \tag{2.55}$$

It must be noted that, if the impulses act on the system at different times (i.e. some impulses do not intervene at t_0), the result in (2.55) is still valid only after the last impulse has acted on the system.

In order to have a non-oscillatory final response, the resulting amplitude A_t should be equal to zero. This yields, for $t > \max(t_i)$, to

$$\begin{cases} \sum_{i=1}^{N} B_i e^{-\xi\omega_n(t_N - t_i)} \sin(t_i\omega_n\sqrt{1-\xi^2}) = 0 \\ \sum_{i=1}^{N} B_i e^{-\xi\omega_n(t_N - t_i)} \cos(t_i\omega_n\sqrt{1-\xi^2}) = 0. \end{cases} \tag{2.56}$$

The solution of system (2.56) gives the amplitudes and times of the impulses that would result in zero oscillation for the system.

Two-impulse (ZV) input shaping

The simplest and most diffused input shaping filter derives from solving system (2.56) with $N = 2$ and imposing a strictly positive value for the impulse amplitudes, together with the condition

$$\sum_{i=1}^{N} A_i = 1.$$

The resulting solution, which is named Zero Vibration (ZV) Input Shaping, in terms of amplitudes and times of the two impulses, can be written as

$$\mathbf{ZV} = \begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+K} & \dfrac{K}{1+K} \\ 0 & \dfrac{\pi}{\omega_d} \end{bmatrix} \tag{2.57}$$

with

$$K = e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}}$$

and w_d the damped natural frequency of the system

$$w_d = w_0\sqrt{1-\xi^2}.$$

A scheme of the ZV input shaping technique is shown in Figure 2.6.

ZVD and ZVDD input shaping

By increasing the number of impulses, other conditions must be added in order to analytically solve system (2.56). One solution is to set as equal to zero the derivatives of the resulting amplitudes, that is, the first equation in system (2.55). With $N = 3$, and by setting equal to zero the first derivatives of the resulting amplitude, the series of impulses found have the name of Zero Vibration and Derivatives (ZVD) input shaping, and can be written as

$$\mathbf{ZVD} = \begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+2K+K^2} & \dfrac{2K}{1+2K+K^2} & \dfrac{K^2}{1+2K+K^2} \\ 0 & \dfrac{\pi}{\omega_d} & \dfrac{2\pi}{\omega_d} \end{bmatrix} \tag{2.58}$$

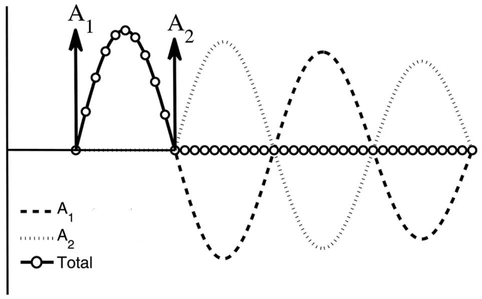


Figure 2.6: Two-impulse (ZV) IS method.

By further increasing the number of impulses to $N = 4$, the second derivatives of the total amplitude must be set equal to zero (giving name to the technique as ZVDD input shaping), and the resulting impulses series is defined by

$$\mathbf{ZVDD} = \begin{bmatrix} A_i \\ \\ t_i \end{bmatrix} = \begin{bmatrix} \dfrac{1}{D_{ZVDD}} & \dfrac{3K}{D_{ZVDD}} & \dfrac{3K^2}{D_{ZVDD}} & \dfrac{K^3}{D_{ZVDD}} \\ \\ 0 & \dfrac{\pi}{\omega_d} & \dfrac{2\pi}{\omega_d} & \dfrac{3\pi}{\omega_d} \end{bmatrix} \tag{2.59}$$

where

$$D_{ZVDD} = 1 + 3K + 3K^2 + K^3 \tag{2.60}$$

The reason for increasing the number of impulses is that, by setting equal to zero high order derivatives of A_t, the robustness of the method with respect to errors in the model parameters increases, as shown in Figure 2.7.

While increasing the number of the impulses could be further increased, it must be noted that N is strictly linked to the time of the last impulse. In particular, t_N is equal to $N - 1$ times the half-period of the system oscillation. For this reason, as will be shown later, it is unpractical to increase the total number of impulses, as it results in an increase of the delay introduced by the input shaping filter.

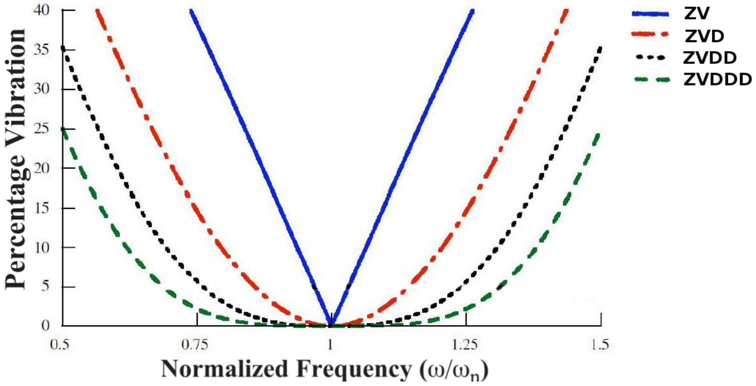


Figure 2.7: Robustness of the IS methods [74].

Extra Insensitive input shaping

Being input shaping an open-loop technique, its robustness is a key issue when applied to industrial applications, as it is rarely possible to have an accurate model of the system to be controlled, and errors in the model parameter can be relevant.
One way to further increase the robustness of the technique without increasing the introduced delay is to allow a certain level of residual oscillation A_t in the nominal case, resulting in the so called Extra Insensitive (EI) input shapers. The rationale behind this is shown in Figure 2.8

For example, the resulting filter obtained by setting the admissible oscillation at the nominal case equal to V_{tol} and $N = 3$ is given by

$$\mathbf{EI} = \begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} A_1 & 1-(A_1+A_3) & A_3 \\ 0 & t_2 & \dfrac{2\pi}{\omega_d} \end{bmatrix} \tag{2.61}$$

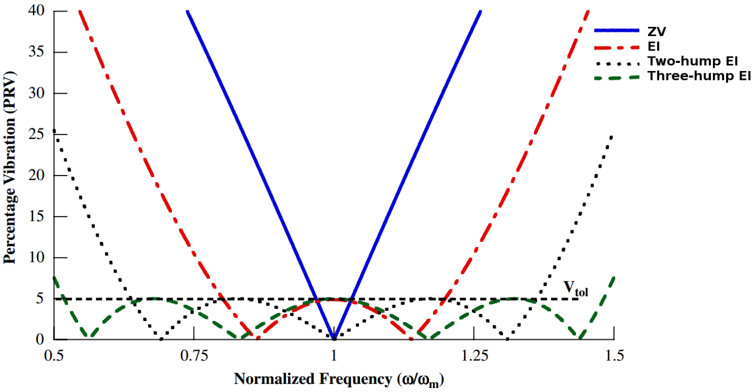


Figure 2.8: Robustness of Extra Intensive shapers [74].

with

$$
\begin{aligned}
A_1 =& 0.24968 + 0.24962V_{tol} + 0.80008\xi + 1.23328V_{tol}\xi + 0.49599\xi^2 \\
& + 3.17316V_{tol}\xi^2 \\
\\
A_3 =& 0.25149 + 0.21474V_{tol} - 0.83249\xi + 1.41498V_{tol}\xi + 0.85181\xi^2 \\
& - 4.90094V_{tol}\xi^2 \\
\\
t_2 =& \frac{2\pi}{\omega_d}(0.49990 + 0.46159V_{tol}\xi + 4.26169V_{tol}\xi^2 + 1.75601V_{tol}\xi^3 \\
& + 8.57843V_{tol}^2\xi - 108.644V_{tol}^2\xi^2 + 336.989V_{tol}^2\xi^3)
\end{aligned}
\qquad (2.62)
$$

The parameters of other EI filters (for $N = 4$ and $N = 5$) can be found in [74].

Negative impulse input shapers

One limitation of input shaping is that the delay introduced in the input is forced to be a multiple of half of the oscillation period of the system. For some applications, for which a fast manoeuvre time is more relevant than

a precise and robust oscillation compensation, and for which the interested oscillation period can last for seconds, half of a period can be an unacceptable delay.
By allowing the impulses that define input shaping filters to have negative values, it is possible to overcome this limitation [68]. By limiting the amplitude of the impulses to unitary values results in Unitary Module (UM) input shaping

$$\mathbf{UM} = \begin{bmatrix} A_i \\ \\ \\ t_i \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ \\ 0 & t_2 & t_3 \end{bmatrix} \tag{2.63}$$

where

$$\begin{aligned} t_2 &= \frac{2\pi}{\omega_n}(0.16724 + \xi 0.27242 + \xi^2 0.20345) \\ t_3 &= \frac{2\pi}{\omega_n}(0.33323 + \xi 0.00533 + \xi^2 0.17914 + \xi^3 0.20125) \end{aligned}$$

An input shaping with further reduced delay is the Partial Sum (PS) input shaping, for which the cumulative sum of the impulses is limited to the range $[-1, 1]$:

$$\mathbf{PS} = \begin{bmatrix} A_i \\ \\ t_i \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 \\ \\ 0 & t_2 & t_3 \end{bmatrix} \tag{2.64}$$

where

$$\begin{aligned} t_2 &= \frac{2\pi}{\omega_n}(0.20970 + \xi 0.22441 + \xi^2 0.08028 + \xi^3 0.23124) \\ t_3 &= \frac{2\pi}{\omega_n}(0.29013 + \xi 0.09557 + \xi^2 0.10346 + \xi^3 0.24624) \end{aligned}$$

Input shaping as a filter

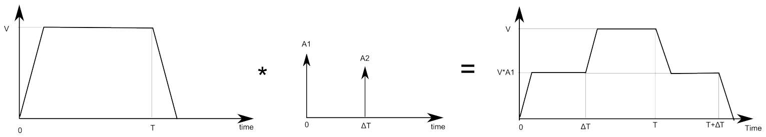


Figure 2.9: Input shaping used as a filter for a general input signal

Until now, the provided results only consider the system response to a train of impulses. The convolution principle extends the use of the Input Shaping approach to general inputs, as shown in Figure 2.9.

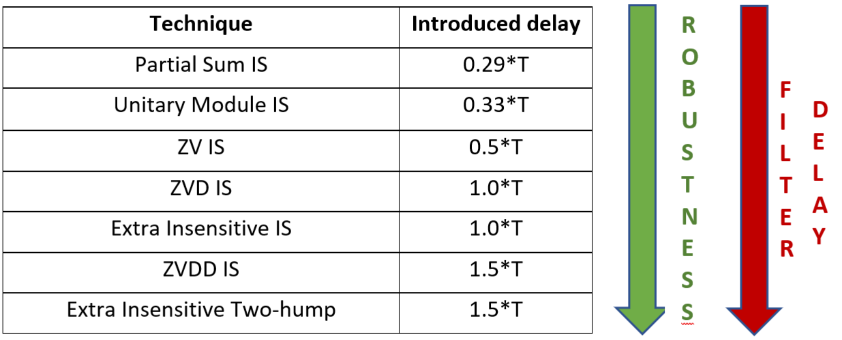
In this sense, input shaping can handily be implemented as a filter on the input of the system.

Input shaping for multiple oscillations

The extension of input shaping for the control of multimode system is straightforward. Thanks again to the superposition principle, multiple input shaping filters can be designed, one for each oscillatory frequency, and the input must be filtered by the series of all the build filters.
One drawback of this approach is that the delays introduced by the filters sum each other, resulting in a total delay which could slow down the performance of the system to an unacceptable level.

The choice of the filter

Industrial cranes are used for lifting a number of different loads, with very different inertias which influence the natural oscillation frequency of the system. As already mentioned, it is therefore of the uttermost importance to address the issue of robustness of the technique with respect to errors in the model parameters.
Table 2.10 resumes the rule of thumb for the choice of the correct filter for the application. When robustness is a critical factor, performance in terms of fast acceleration times should be sacrificed, and more robust techniques such as Extra Insensitive Two-Hump input shaping should be adopted. On the contrary, in applications where the key goal is the reduction of the



Technique	Introduced delay
Partial Sum IS	0.29*T
Unitary Module IS	0.33*T
ZV IS	0.5*T
ZVD IS	1.0*T
Extra Insensitive IS	1.0*T
ZVDD IS	1.5*T
Extra Insensitive Two-hump	1.5*T

Figure 2.10: Trade-off between introduced delay and robustness for input shaping filters. T is the period of the oscillation to be deleted.

manoeuvre time, and (reduced) residual oscillation are acceptable, more aggressive approaches such as PS and UM input shaping can be considered. In Section 2.3.5 the robustness of input shaping methods will be compared with the robustness of the other open-loop control techniques presented in this chapter.

2.3.1.1 MIL validation

In order to validate IS methodologies for the control of industrial cranes, a Simulink model of a double pendulum overhead crane have been developed using Simscape Multibody. The parameters describing the model are shown in Figure 2.3.

A series of two input shaping filters have been used to filter the cart reference velocity. Given the parameters in Table 2.3, it can be demonstrated that the two natural oscillation frequencies are given by

$$\omega_{n_1} = 1.49 \text{ [rad/s]}, \quad \omega_{n_2} = 3.39 \text{ [rad/s]}, \tag{2.65}$$

while the damping of the oscillation can be neglected, as done in practice for the control of industrial cranes.

From (2.65), the oscillation periods necessary for the tuning of the two filters are respectively

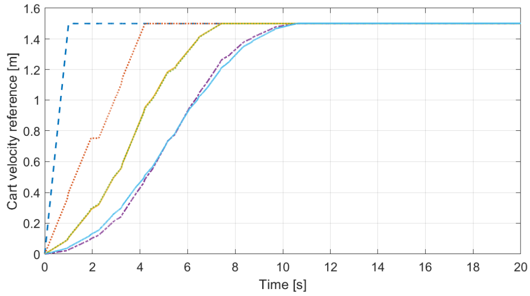
$$T_1 = 4.217 \text{ [s]}, \quad T_2 = 1.853 \text{ [s]}. \tag{2.66}$$

Figure 2.11(a) shows the original ramped cart trajectory reference together with the filtered reference for all the simulated input shaping approaches. The delay introduced by the filters depends on the choice of the filter, as shown by Figure 2.10.

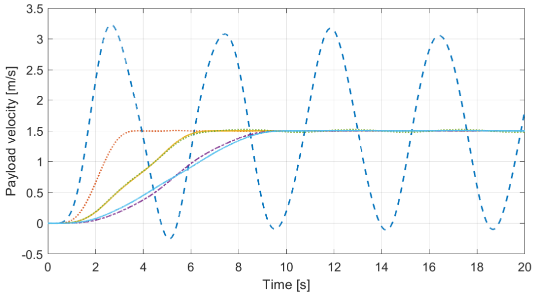
The resulting payload velocity is shown in Figure 2.11(b). For all the techniques the resulting trajectories reach the setpoint value with small residual oscillations with respect to the unfiltered case. This is also evident by analyzing Figure 2.11(c), which shows the displacement of the payload with respect to the position of the cart. After an initial transient, the trajectories resulting from the input shaping filtered inputs are affected by minor residual oscillations.

2.3.1.2 HIL validation

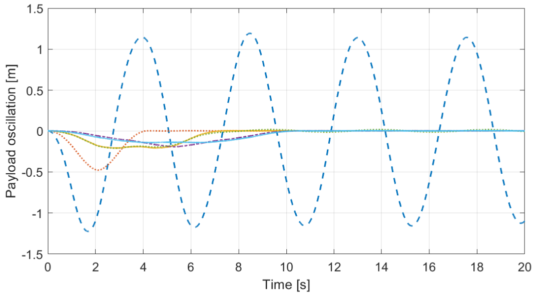
Input shaping techniques have been implemented as function blocks for the ADV200 industrial drive of the industrial setup in Figure 1.6, while the SM simulated the dynamics of the double pendulum described by the parameters in Table 2.3. The results, shown in Figure 2.12, are coherent with the MIL test results of Figure 2.11, so that input shaping techniques effectively reduce residual oscillations when applied to the antisway control of overhead cranes.



(a) Cart velocity.

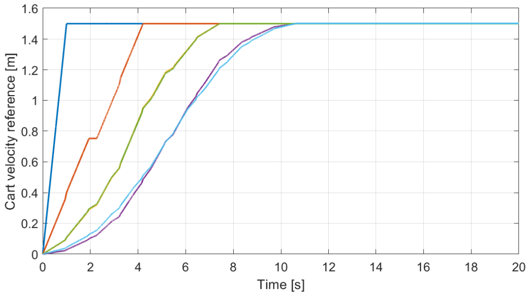


(b) Payload velocity.

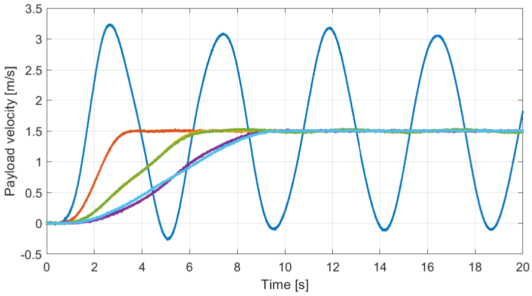


(c) Displacement of the payload with respect to cart position

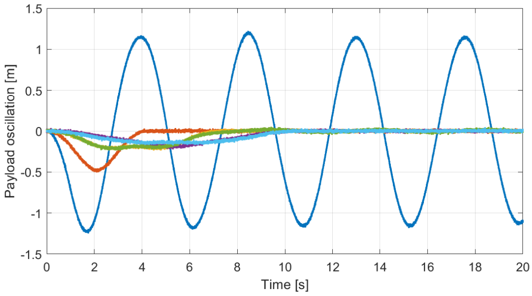
Figure 2.11: MIL validation of input shaping techniques. In dashed blue line, the unshaped command. The shaped commands are obtained with ZV (orange dotted line), ZVD (green dotted line), ZVDD (light-blue solid line), EI (yellow solid line) and EI Two-Hump (dash-dot purple line) input shaping



(a) Cart velocity.



(b) Payload velocity.



(c) Displacement of the payload with respect to cart position

Figure 2.12: HIL validation of input shaping techniques. In blue line, the unshaped command. The shaped commands are obtained with ZV (orange line), ZVD (green line), ZVDD (light-blue line), EI (yellow line) and EI Two-Hump (purple line) input shaping

2.3.2 Input-output inversion

Input-output inversion is an open-loop technique which consists in computing an adequate input for the system in order to have a predefined non-oscillating response by inverting the transfer function of the system. Thus, once a suitable desired payload trajectory has been defined, the corresponding input can be found analytically. Unlike input shaping, the transient time necessary for reaching a situation with zero oscillation can be varied, as it is indeed the tuning variable of the approach. The operator, depending on the application, can choose between a fast manoeuvre or a more robust approach.

Input-output inversion has been demonstrated to be a suitable choice for the control of industrial overhead cranes [51], also when double pendulum dynamics are present [25].

The input-output inversion approach for the control of industrial cranes is presented in [51] and is here summarized.

Considering a generic strictly proper linear system in the form

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\beta_m s^m + \beta_{m-q} s^{m-1} + \cdots + \beta_1 s + \beta_0}{\alpha_n s^n + \alpha_{n-1} s^{n-1} + \cdots + \alpha_1 s + 1} \tag{2.67}$$

with a relative order

$$\rho = n - m > 0.$$

The linear system in (2.67) can be inverted, resulting in

$$H^{-1}(s) = \frac{U(s)}{Y(s)} = \gamma_\rho s^\rho + \gamma_{\rho-1} s^{\rho-1} + \cdots + \gamma_1 s + \gamma_0 + H_0(s), \tag{2.68}$$

where

$$H_0(s) = \frac{\sigma_{m-1} s^{m-1} + \sigma_{m-2} s^{m-2} + \cdots + \sigma_0}{\beta_m s^m + \beta_{m-q} s^{m-1} + \cdots + \beta_1 s + \beta_0} \tag{2.69}$$

is the zero order dynamics of the system $H(s)$.

It can be demonstrated [25] that, for highly underdamped systems, the contribution of $H_0(s)$ to the dynamic of the inverted system can be ignored, so that in this case the inverted transfer function in (2.68) can be written as

$$H^{-1}(s) = \frac{U(s)}{Y(s)} = \gamma_\rho s^\rho + \gamma_{\rho-1} s^{\rho-1} + \cdots + \gamma_1 s + \gamma_0. \qquad (2.70)$$

This means that the input $U(s)$ of the system can be computed as a weighted sum of the derivatives of the output $Y(s)$ up to order ρ, where the weighting coefficients are $[\gamma_\rho, \gamma_{\rho-1}, \ldots, \gamma_1, \gamma_0]$, that is,

$$U(s) = H^{-1}(s)Y(s). \qquad (2.71)$$

Consequently, by defining an adequate output trajectory, the corresponding input can be easily computed. For the output trajectory to be adequate, it has to be ρ times differentiable. For this reason, a polynomial form for the output trajectory seems like a good choice, as its differentiability order is easily defined.

The polynomial normalized output trajectory $\bar{y}$ can be written in the form

$$\bar{y}(t,\tau) = \begin{cases} 0 & t \leq 0 \\ \frac{(2k+1)!}{k!\tau^{2k+1}} \sum_{i=0}^{k} \frac{(-1)^{k-i} t^{2k-i+1}}{i!(k-i)!(2k-i+1)} & 0 < t < \tau \\ 1 & t \geq \tau \end{cases} \qquad (2.72)$$

where τ is the transient time of the polynomial trajectory and $2k+1$ is the order of the polynomial. In order for (2.72) to be a feasible output trajectory for system (2.67), k must be chosen so that

$$2k + 1 \geq \rho.$$

The derivative of order α of (2.72) D^α can be written in closed form as

$$D^\alpha \bar{y}(t,\tau) =$$

$$\begin{cases} 0 & t \leq 0 \\ \frac{(2k+1)!}{k!\tau^{2k+1}} \sum_{i=0}^{k} \frac{(-1)^{k-i} \tau^i (2k-i+1)}{i!(k-i)!(2k-i+1)(2k-i+1-\alpha)!} \tau^i t^{2k-i+1-\alpha} & 0 < t < \tau \\ 0 & t \geq \tau \end{cases} \qquad (2.73)$$

The corresponding input can be therefore computed in closed form as

$$u(t,\tau) = \gamma_\rho D^\rho y(t,\tau) + \gamma_{\rho-1} D^{\rho-1} y(t,\tau) + ... + \gamma_0 y(t,\tau) \tag{2.74}$$

2.3.3 Input-output inversion for double-pendulum overhead cranes

The proposed simplified input-output inversion approach is here extended to the control of a overhead crane which exhibits double pendulum dynamics. The approach and the results have been published in [25].

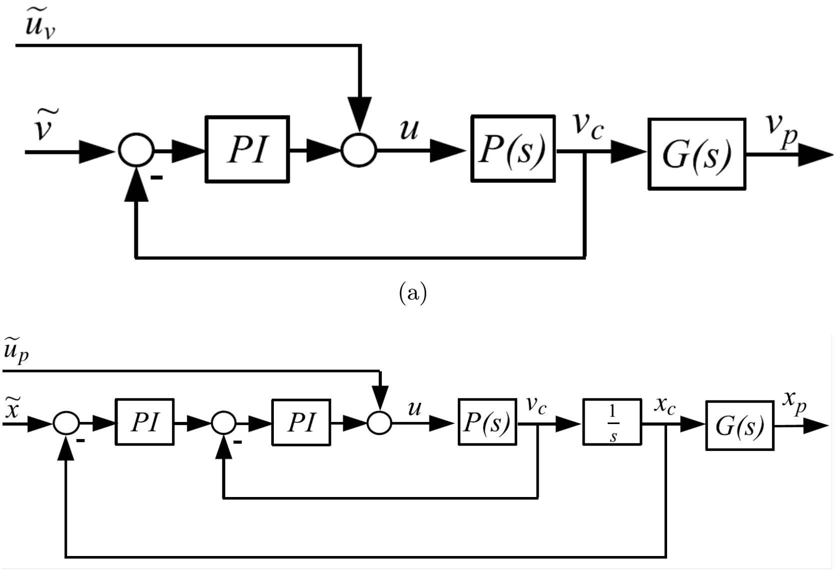
Input-output inversion can be applied for both the case of position and velocity control of overhead cranes. Depending on the variable to be controlled, the schemes in Figure 2.13(a) or Figure 2.13(b) can be chosen.

While the cart position or velocity trajectories can be computed by computing the inverse of the transfer function that links the cart kinematics with the payload velocity and position (2.46), the feedforward force signal can be computed by inverting the transfer function that describes the dynamic of the crane when the considered input is the force acting on the cart (that is, equation (2.41)). The feedforward force signal increases the tracking performance of the control system.

By inverting equation (2.46), the resulting inverted transfer function can be written in the form

$$G^{-1}(s) = \gamma_2 s^2 + \gamma_1 s + \gamma_0 + \frac{\rho_1 s + \rho_0}{N_F(s)}. \tag{2.75}$$

If the damping coefficients C_1 and C_2 in equation (2.46) are small enough [25], the zero dynamics of the system can be neglected when computing the input of the system. This brings a major advantage to the approach. In fact, the zero dynamics would result in a convolution integral when computing



(a)

(b)

Figure 2.13: (a) Velocity control scheme and (b) position control scheme.

the input, resulting in a non-analytical form for the input. If the zero dynamics can be neglected, the input results in a parametric closed form, which is constant when $t > \tau$.

Under the assumption of small damping coefficients, such is verified in industrial overhead cranes, (2.75) can be simplified in

$$G^{-1}(s) = \gamma_2 s^2 + \gamma_1 s + \gamma_0 \tag{2.76}$$

.

As transfer function (2.46) describes both the relation between cart velocity and payload velocity, and between cart position and payload position, the result in (2.76) is valid for both the schemes in Figure 2.13.

The desired trajectory, for both velocity ($\dot{\bar{x}}_p$) and position ($\bar{x}_p$) control, can be computed by multiplying the trajectory in (2.72) for the desired steady state payload velocity or position q, resulting in (for the case of position control)

$$\bar{x}_p(t,\tau,q) = \begin{cases} 0 & t \le 0 \\ q\frac{(2k+1)!}{k!\tau^{2k+1}} \sum_{i=0}^{k} \frac{(-1)^{k-i} t^{2k-i+1}}{i!(k-i)!(2k-i+1)} & 0 \le t < \tau \\ q & t \ge \tau \end{cases} \tag{2.77}$$

so that the derivatives are easily computed as

$$D^{\alpha}\bar{x}_p(t,\tau,q) = \\ \begin{cases} q\frac{(2k+1)!}{k!\tau^{2k+1}} \sum_{i=0}^{k} \frac{(-1)^{k-i}\tau^{i}(2k-i+1)}{i!(k-i)!(2k-i+1)(2k-i+1-\alpha)!} \tau^{i} t^{2k-i+1-\alpha} & 0 < t < \tau \\ 0 & t \ge \tau \end{cases} \tag{2.78}$$

The cart velocity or position trajectory, from (2.76), is therefore given by

$$\bar{x}_c(t,\tau,q) = q\left[\gamma_2 D^2 \bar{x}_p(t,\tau) + \gamma_1 D^1 \bar{x}_p(t,\tau) + \gamma_0 \bar{x}_p(t,\tau)\right]. \tag{2.79}$$

The force feedforward for the velocity control can be computed by inverting transfer function (2.42), which results, under the assumption of negligible values of C_1 and C_2, in

$$\begin{aligned}\bar{u}(t,\tau,q) =& \lambda_4 D^4 \bar{x}_p(t,\tau) + \lambda_3 D^3 \bar{x}_p(t,\tau) + \lambda_2 D^2 \bar{x}_p(t,\tau) \\ & + \lambda_1 D^1 \bar{x}_p(t,\tau) + \lambda_0 \bar{x}_p(t,\tau)\end{aligned} \tag{2.80}$$

Similarly, the force feedforward for the position control can be computing by inverting (2.41), resulting in

$$\begin{aligned}\bar{u}(t,\tau,q) =& \psi_6 D^6 \bar{x}_p(t,\tau) + \psi_5 D^5 \bar{x}_p(t,\tau) + \psi_4 D^4 \bar{x}_p(t,\tau) \\ & + \psi_3 D^3 \bar{x}_p(t,\tau) + \psi_2 D^2 \bar{x}_p(t,\tau) + \psi_1 D^1 \bar{x}_p(t,\tau) \\ & + \psi_0 \bar{x}_p(t,\tau)\end{aligned} \tag{2.81}$$

2.3.3.1 MIL validation

The technique has been validate through MIL testing, using a model of the system with the parameters shown in Figure 2.3. As a first example we consider the velocity control with final velocity $q = 5$ [m/s] and transition time $\tau = 4.21$ [s].
The application of the input-output inversion methodology yields the command signal $\dot{\bar{x}}_c$ and the feedforward signal $\bar{u}$, which are shown in Figure 2.14, where the desired output (i.e., the transition polynomial (2.72)) is also shown, along with the response of the nonlinear model. As a second example the position control task is considered, with $q = 10$ [m] and $\tau = 4.21$ [s]. The signals $\bar{x}_c$ and $\bar{u}$ obtained with the inversion based methodology are shown in Figure 2.15 together with the obtained results.

2.3.3.2 HIL validation

The same test have been executed on the HIL setup described in Chapter 1.4.4. A function block with the parametric function for the position, velocity and feedforward force have been coded directly on the ADV200 industrial drive. The only needed parameters are q, which can be either the steady state velocity or position reference depending on the desired variable to be controlled, and the transition time τ.

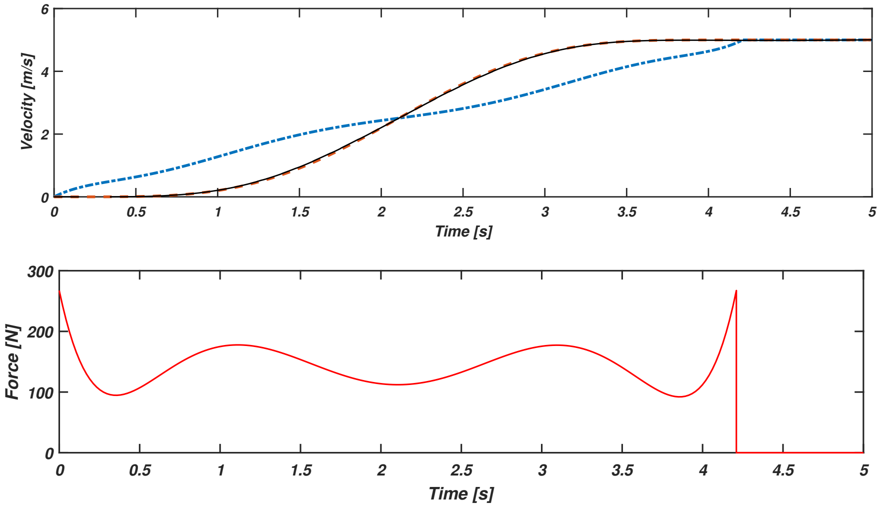


Figure 2.14: Velocity command signal $\dot{x}_c$ (blue dash-dot line), desired payload trajectory (solid black) and actual payload trajectory obtained with the nonlinear model (orange dashed line) (top). Force feedforward signal $\widetilde{u}_v$ for velocity control obtained with input-output inversion of the model (bottom).

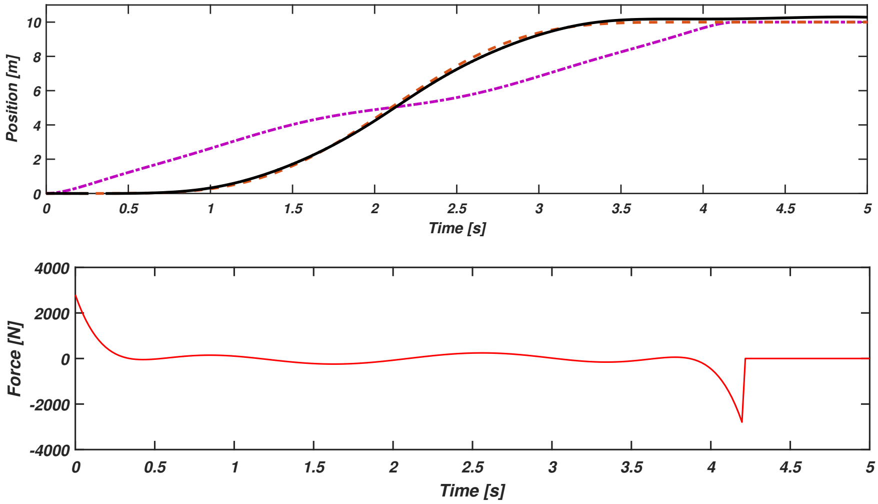


Figure 2.15: Position command signal $\bar{x}_c$ (purple dash-dot line), desired payload trajectory (black solid line) and actual payload trajectory (dashed orange line) (top). Force feedforward signal $\bar{u}$ for position control obtained with input-output inversion of the model (bottom).

For the case of velocity control, the same parameters of the MIL validation phase have been used, that is, a desired velocity of $q = 5$ [m/s] and transition time $\tau = 4.21$ [s].
The application of the input-output inversion methodology yields the command signal $\dot{\bar{x}}_c$ and the feedforward signal $\bar{u}$, which are shown in Figure 2.16, where the desired output (i.e., the transition polynomial (2.72)) is also shown, along with the velocity of the payload, simulated by the SM on the HIL setup. Then, also input-output inversion position control have been validated on the HIL setup, with the same parameters of the MIL validation, that is, a desired final position of $q = 10$ [m] and a transient time $\tau = 4.21$ [s]. The signals $\bar{x}$ and $\bar{u}$ obtained with the inversion based methodology are shown in Figure 2.17 together with the obtained results.

By comparing the results obtained with HIL testing with the ones ob-

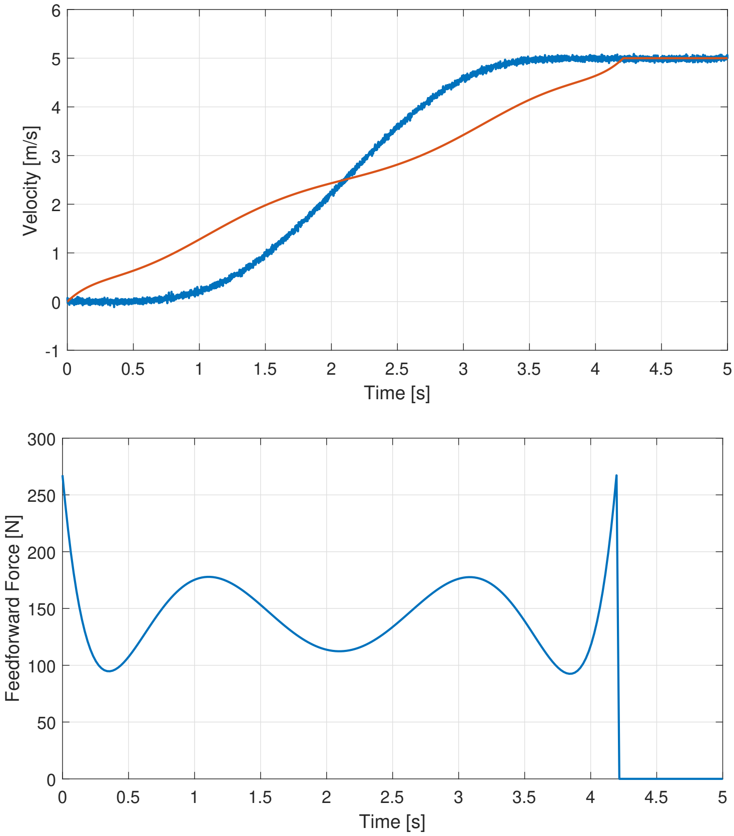


Figure 2.16: Cart velocity command signal $\dot{\tilde{x}}_c$ (orange line), desired payload trajectory (solid black) and actual payload trajectory simulated by the SM on the HIL setup (blue line)(top). Force feedforward signal $\widetilde{u}_v$ for velocity control obtained with input-output inversion of the model (bottom).

tained by MIL testing (Figure 2.14 and Figure 2.15), no significant differences can be seen.

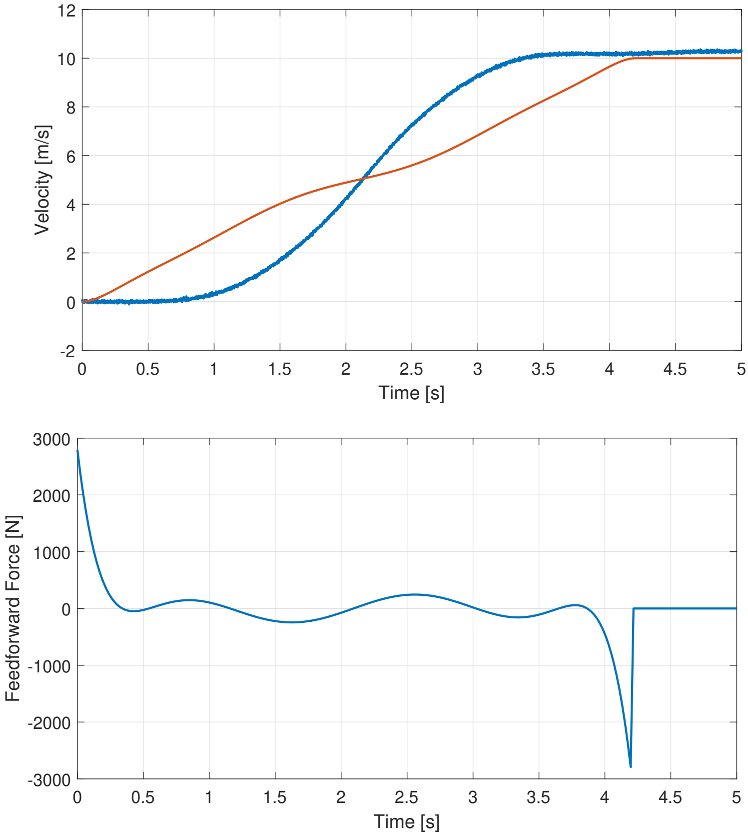


Figure 2.17: Cart position command signal $\bar{x}_c$ (orange line), desired payload trajectory (black line) and actual payload trajectory simulated by the SM on the HIL setup (blue line) (top). Force feedforward signal $\bar{u}$ for position control obtained with input-output inversion of the model (bottom).

2.3.4 Input Notch filtering

Oscillations in dynamic systems are represented by couples of poles in the transfer function with non-zero imaginary part. Typically, if this poles couple is known, it can easily be cancelled by filtering the input with a notch filter [60].

In industrial cranes, the oscillation frequency, and thus the poles position, can be easily expressed as a function of the cable length, while the damping is approximately equal to zero (generated oscillations can last for minutes in industrial cranes). Thus, it is straightforward to apply notch filtering for the anti-sway control of industrial cranes [43].

For the simplest of single pendulum crane dynamics, the oscillation frequency is given by

$$\omega_n(l) = \sqrt{\frac{g}{l}} \tag{2.82}$$

Thus, for highly underdamped cranes, the notch filter can be defined as

$$F_N(s) = \frac{K(s^2 + \omega_n^2)}{(s+a)^3} \tag{2.83}$$

where

$$a = \hat{a}\omega_n$$

and $\hat{a}$ is a tuning parameter that is approximately equal to one. A reduction of this parameters brings increased robustness in spite of reduced performance in terms of rapidity in the response.

Then K is set in order to obtain a unitary gain filter, thus

$$K = \frac{a^3}{\omega_n^2}.$$

2.3.4.1 MIL validation

Notch filter approach has first been validated on the MIL testing bed, with the nonlinear Simscape Multibody model, whose parameters are described in Table 2.3. In order to test different filter robustness, three different values (0.7, 1.5 and 2.5) have been chosen for the tuning parameter $\hat{a}$. This value have been chosen as they approximately correspond to the input shaping filters ZV, ZVD and ZVDD in terms of introduced delay on the input, that is, $0.5(T_1 + T_2)$, $(T_1 + T_2)$ and $1.5(T_1 + T_2)$, where $(T_1 + T_2)$ are the natural periods of the system.

Figure 2.18(a) shows the velocity reference for the cart, in the filtered and unfiltered cases. By comparing the velocity trajectories with Figure 2.11(a), the filter with $\hat{a}$ result in a slightly more aggressive cart velocity trajectory with respect to its analogue ZV input shaping filter.

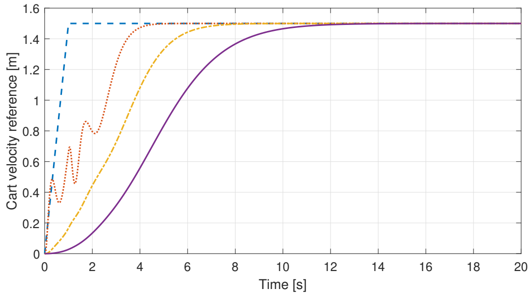
Payload velocities, shown in Figure 2.18(b), reach without oscillating the setpoint value. In fact, the oscillation (shown in terms of relative displacement of the payload with respect to the cart position), after an initial transient, is contained and stabilizes around zero.

2.3.4.2 HIL validation

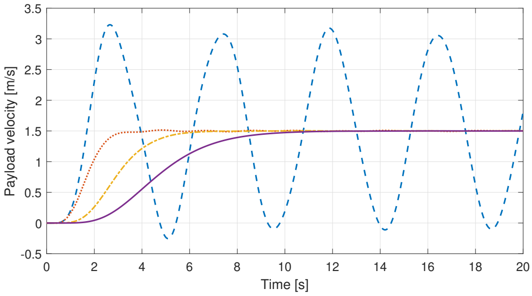
A notch filter function block has been built on the ADV200 industrial drive controlling the MUT of the HIL setup described in 1.4.4, while the SM motor simulated the behaviour of the double pendulum crane already described. By setting the same parameters as in the MIL validation, that is, $\hat{a} = [0.7, 1.5, 2.5]$, the resulting cart trajectories are shown in Figure 2.19(a). The payload velocity trajectories are shown in Figure 2.19(b). In all the cases, except for the unfiltered case, payload velocity reaches the setpoint value with only small residual oscillations.

This can also be seen by comparing the payload displacement with respect to the position of the cart, as shown in Figure 2.19(c).

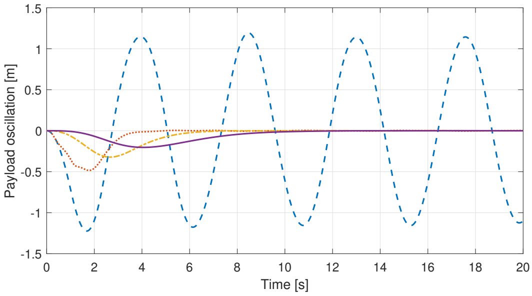
The results obtained during the HIL validation are consistent with the results obtained with the MIL, as the use of notch filtering technique successfully reduces the residual oscillation of the payload.



(a) Cart velocity.

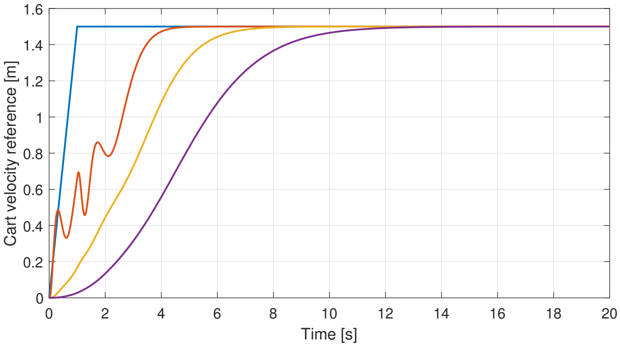


(b) Payload velocity.

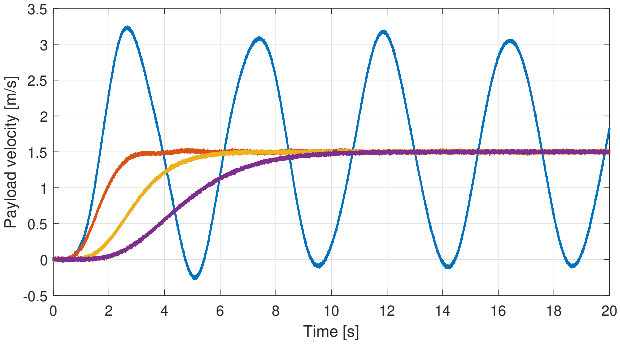


(c) Displacement of the payload with respect to cart position.

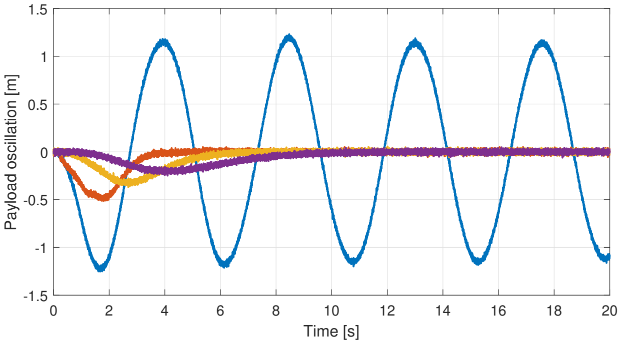
Figure 2.18: HIL validation of notch filtering. Unfiltered cart velocity command reference (blue dashed line), and notch filtered signals obtained by setting a delay time in the filter corresponding to $0.5(T_1 + T_2)$ (orange dotted line), $(T_1 + T_2)$ (yellow dash-dot line) and $1.5(T_1 + T_2)$ (solid purple line), where $(T_1 + T_2)$ are the natural oscillation periods of the system.



(a) Cart velocity.



(b) Payload velocity.



(c) Displacement of the payload with respect to cart position.

Figure 2.19: HIL validation of notch filtering. Unfiltered cart velocity command reference (blue line), and notch filtered signals obtained by setting a delay time in the filter corresponding to $0.5(T_1+T_2)$ (orange line), (T_1+T_2) (yellow line) and $1.5(T_1+T_2)$ (solid line), where (T_1+T_2) are the natural oscillation periods of the system.

2.3.5 A comparison between open-loop techniques

In order to compare the presented open-loop techniques for the control of industrial cranes, intensive simulations have been executed in order to test their robustness with respect to errors in the model parameters.

The measure of robustness adopted in this thesis is the sensitivity curves surface, which correlates the maximum residual oscillation of the payload with the parameters uncertainty. The reason of the success of the maximum residual oscillation for the evaluation of the robustness lies in its simplicity and at-a-glance comprehensibility, which renders the adopted metric very suitable for non highly-specialized engineers and industrial practitioners. This is in line with the objective of devising industrially feasible solutions, easily implementable by using off-the-shelf automation components.

In order to obtain a fair comparison of the techniques, various input shaping filters have been tested, and both input-output inversion and input notch filtering control parameters have been adjusted in order to set the introduced delay of each technique comparable with the various input shaping techniques.

The system on which the robustness is tested is a double-pendulum overhead crane, whose simulated model is described by the parameters in Table 2.3. The input shaping techniques tested are shown in Table 2.4, along with the delay they introduce in the nominal case.

Table 2.4: Delays introduced by the input shaping techniques. T is the sum of the periods of the system to be controlled.

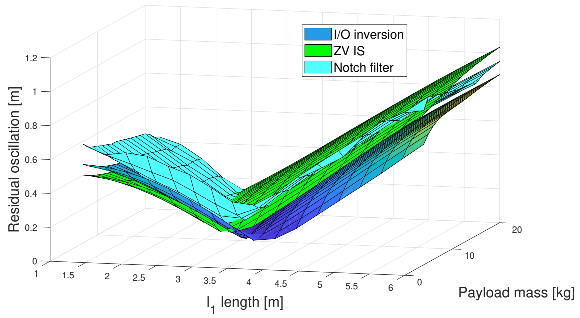
IS technique	Delay	Total transient time [s]
ZV	$0.5T$	4.21
ZVD	T	7.42
EI	T	7.42
ZVDD	$1.5T$	10.63
Two-hump EI	$1.5T$	10.63

As input-output inversion has τ as its tuning parameter, which is exactly the transition time, it is easy to directly compare input shaping and input-output inversion by setting τ equal to the delay introduced by the input shaping filter.
The notch filter parameters have been tuned manually to have the same delay as input shaping.

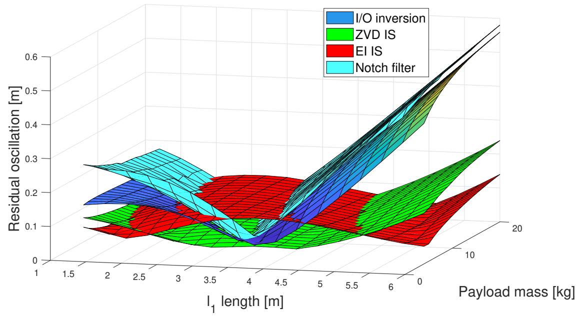
In order to test the robustness, both the payload mass m_2 and the hoisting cable length l_1 have been varied.
Residual oscillations are measured at the end of the acceleration phase, with a reference steady state velocity of 1.5 [m/s], simulating an Operator-In-the-Loop overhead crane.

The resulting sensitivity curves surfaces are shown in Figure 2.20.
It can be seen that the resulting robustness is comparable when the total delay time introduced by the techniques is the same. In particular in Figure 2.20(a) all the techniques have comparable robustness, but the notch filter approach is slightly less robust when the estimated length of l_1 is less than the nominal value of 3 [m].
In Figure 2.20(b) input-output inversion and notch filter have comparable robustness. ZVD input shaping results in a flat sensitivity curve around the nominal value of l_1, while Extra Insensitive input shaping have non-zero residual oscillation at the nominal case, but contrarily to the other techniques it shows a decreasing residual oscillation when the length l_1 moves away from its nominal value.
In Figure 2.20(c), all the techniques show comparable robustness around the nominal values, except for the notch filter, which result in a less robust sensitivity curve.

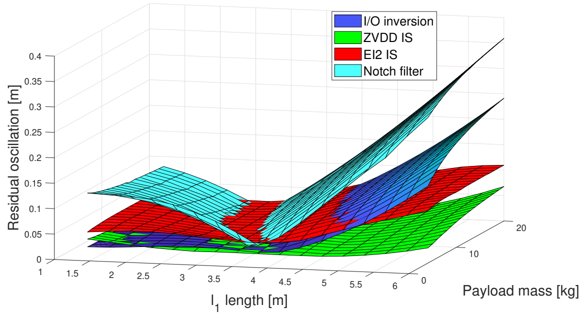
Moreover, as shown by Figure 2.21 and Figure 2.22, the results in terms of residual oscillations when comparing input shaping to input-output inversion are very similar, both in the case of velocity and position control, meaning that the robustness does not depend on the controlled variable.



(a) Total time of $1 + \frac{T_1 + T_2}{2}$ seconds.



(b) Total time of $1 + T_1 + T_2$ seconds.



(c) Total time of $1 + \frac{3(T_1 + T_2)}{2}$ seconds.

Figure 2.20: Comparison of the robustness to parameter errors of input shaping, input-output inversion and notch filtering techniques for the manoeuvre of an OIL double-pendulum crane.

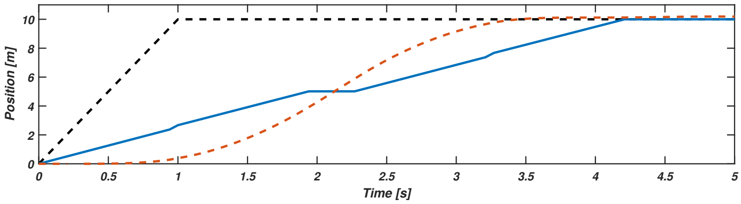


Figure 2.21: Position trajectory reference of the cart (black dashed line) and shaped cart position signal (blue solid line), using the ZV shaper, and resulting payload trajectory (dashed orange line).

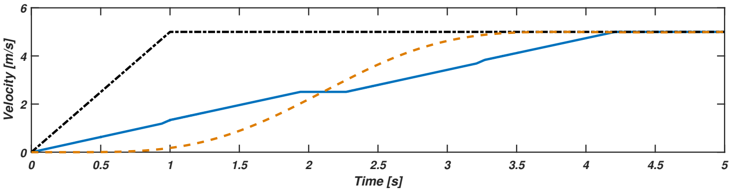


Figure 2.22: Velocity ramp trajectory reference of the cart (black dash-dot line), shaped cart velocity signal (blue solid line) using the ZV shaper, and actual payload trajectory (dashed orange line).

2.3.5.1 Experimental results

In order to test the effectiveness of the proposed techniques on a real system, having evaluated the seemingly greater robustness of input shaping and input-output inversion techniques with respect to notch filtering, these two techniques have been developed as function blocks for the ADV200 Gefran inverter and tested on the overhead crane inside Gefran industrial warehouse. Gefran s.p.a. already optionally provides the ADV200 inverter with a built-in software specially dedicated to hoist and crane solutions. Nonetheless, the previous available solution was based on a predefined set of antisway trajectories for the cart to follow, mainly based on a ZV input shaping approach. Hence, the already available solution was not capable

of coping with user-defined trajectory and its robustness was not tunable, resulting in a non flexible solution.
The introduction of input shaping and input-output inversion function blocks can increase the flexibility of antisway solutions. Input shaping function blocks can work as filters, being independent from the input signal which can be generated by the operator depending on the particular application. input-output inversion provide for a very practical tuning of the technique robustness, which can again be defined depending on the application. As already mentioned, the reason for not having experimentally tested the notch filter approach is the lack in robustness when compared to robust input shaping approaches when applied to the antisway control of overhead cranes, as shown in Figure 2.20(c). With already being able to chose between input shaping and input-output approaches, which have been demonstrated through intensive simulations to be more robust than notch filtering, this last technique has been excluded by the experimental testing procedure.
The techniques have therefore been tested on the industrial overhead crane at Gefran s.p.a., manoeuvreing a payload consisting in a 1000 [kg] high voltage transformer. The load was moved for a predefined displacement and then stopped. The residual oscillations where measured by recording the load after the cart has stopped and by identifying the displacement of a single point of the load during some oscillation periods. A frame of the resulting post-processing procedure is shown in Figure 2.23.

The results of the testing are shown in Figure 2.24 and Figure 2.25. Both the techniques effectively reduce significantly the residual oscillations when applied for the control of industrial overhead cranes. It must be noted that, as the parameters of the real system where roughly estimated, errors in the parametrization of the model must have been made. For this reason, it can be seen that more robust methods effectively tackle the problem of residual oscillations also when based on uncertain models.

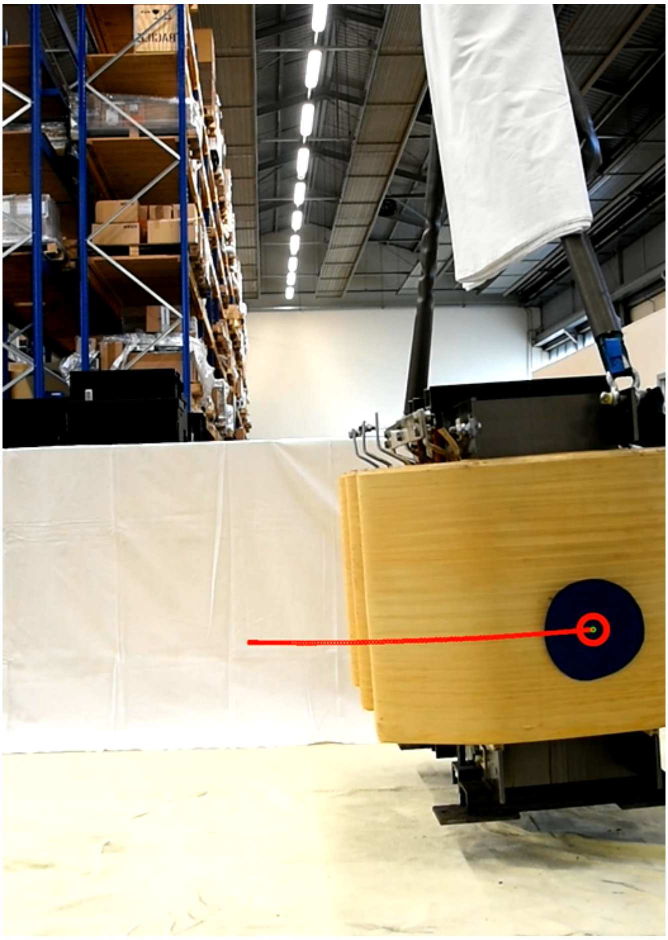


Figure 2.23: Measuring procedure of residual oscillations during the test of open-loop technique on the industrial crane in Gefran s.p.a.. A marker was placed in order to identify a single point of the load, and the video was analyzed frame per frame.

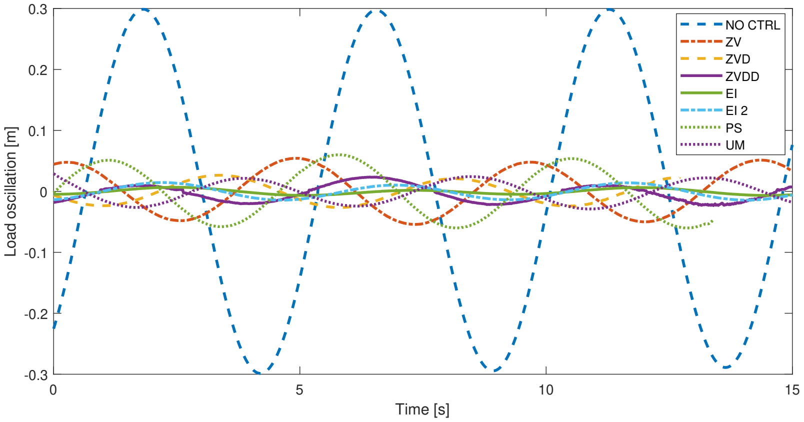


Figure 2.24: Residual oscillation of the overhead crane with input shaping control. Residual oscillations are greatly reduced with respect to the case when no antisway technique is used.

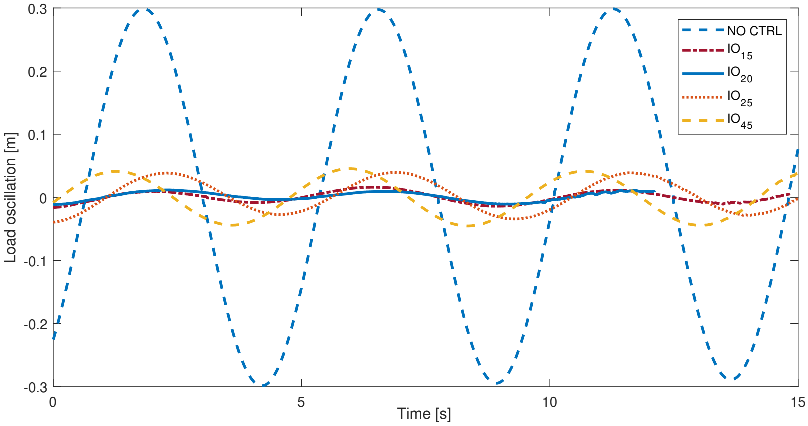


Figure 2.25: Residual oscillation of the overhead crane with input-output inversion control. Residual oscillations are greatly reduced with respect to the case when no antisway technique is used.

2.3.6 An example of application of open-loop techniques to boom cranes

All the open-loop techniques presented in this chapter strongly rely on the assumption of linearity for the system to be controlled. This assumption is acceptable for gantry, overhead and portal cranes, for which the sources of nonlinearities in their dynamics are limited to the sinus and cosinus functions, whose contribution is limited for small sway angles.
The dynamics of other cranes configurations, such as tower and boom cranes, can not be so easily described by means of linear models, as they are inherently nonlinear. While their kinematics can be easily described by means of polar coordinates, their dynamics cannot be reduced to linear models.

As the strength of the proposed open-loop approaches lies in their simplicity and easy application, tailoring complex open-loop approaches for this nonlinear cranes would disrupt the aim of this work, which lays in providing easy-to-use mudules for the control of industrial applications.

Focusing on the concept behind input shaping, input-output inversion and notch filtering approaches, we note that only input-output inversion approach strictly relies on the linear model of the system, while input shaping and notch filtering are based on the suppression of certain frequencies on the input signal. This approach can provide decreased residual oscillations also when applied to highly nonlinear systems, such as boom cranes.

Input shaping and notch filtering approaches will now be applied for the control of a boom crane, and simulations will show the effectiveness of these techniques in containing the residual oscillations affecting the payload. The model of the boom crane is described in Chapter 2.1.3. A nonlinear model of the crane has been built using Simscape Multibody in Simulink environment, with values for its parameters which are shown in Table 2.5 and which reflect the values of a real industrial boom crane.

For the sake of simplicity, only the rotation angle α is controlled during the simulation, while the pitching angle β has been kept constant to 0 [rad].

Table 2.5: Parameters of the boom crane model used in simulations.

Model data	
Parameter	Value
L	55 [m]
m_L	3000 [kg]
L_R	30 [m]

In this configuration, to all effects, the boom crane behaves as a tower crane.

In order to verify the effectiveness of the proposed open-loop methods, different end positions have been identified for the rotation angle α. The velocity reference signal for α has been built as a three traits law of motion, with acceleration $\ddot{\alpha} = 0.0279$ [rad/s^2] and maximum velocity $\dot{\alpha}_{max} = 0.0838$ [rad/s].

The velocity reference has then been filtered using different input shaping and notch filters. In order to tune the filters, the required parameters (oscillation period and natural frequency) can be easily obtained as

$$\begin{aligned} T =& 2\pi\sqrt{\frac{L_R}{g}} \\ \omega_n =& \frac{1}{T}2\pi \end{aligned} \tag{2.84}$$

ZV, ZVD and ZVDD input shaping filters have been tested. For the sake of a fair comparison, the notch filter tuning parameter $\hat{a}$ has been adjusted in order to introduce a delay in the filtered signal equal to the delay introduced by the input shaping filters, that is, equal to $\frac{T}{2}$ for the ZV filter, T for the ZVD filter and $\frac{3T}{2}$ for the ZVDD filter.

The payload trajectories resulting from the simulations are shown in Figure 2.26, Figure 2.27 and Figure 2.28. It is easy to see that the introduction of both either input shaping or notch filters greatly improve the

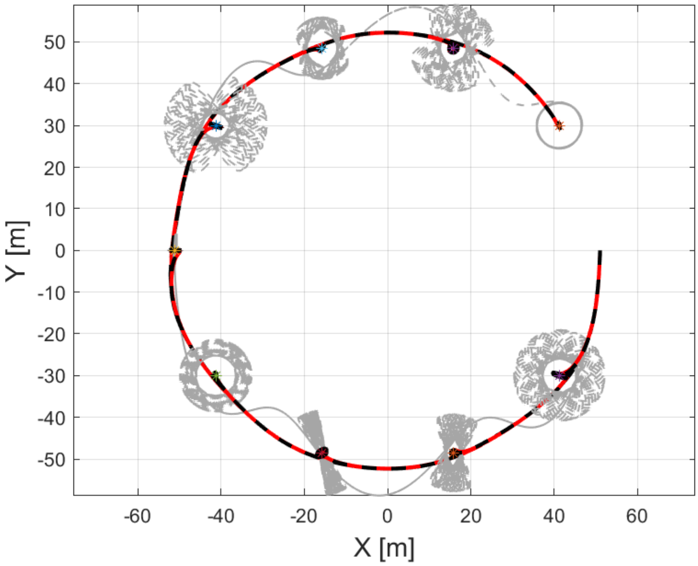


Figure 2.26: Payload trajectories for the case of unfiltered velocity reference (grey dashed line), ZV input shaping velocity filtered reference (black solid line) and notch filtered velocity reference, with $\hat{a} = 2.0$ (dashed red line).

trajectories of the payload, by both reducing the oscillations during and after the manoeuvres.

In order to better understand the contribution of the two approaches, the residual oscillation after the movement has been evaluated as the maximum distance, in the horizontal plane, between the projection of the point on which the cable is connected to the boom and the position of the payload.

The results of this evaluation are shown in Figure 2.29, 2.30 and 2.31. As already pointed out, the residual oscillations are greatly reduced when filtering the reference signal. The two approaches (input shaping and notch filtering) provide similar performance when the filter introduces a delay corresponding to the ZV approach.

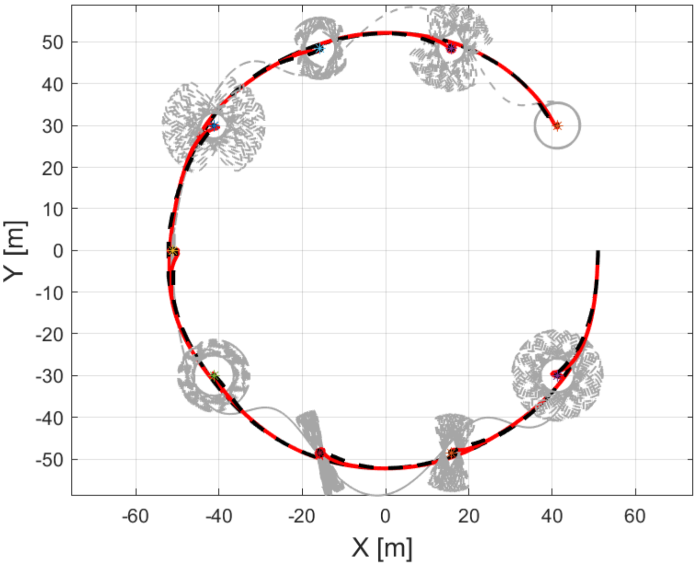


Figure 2.27: Payload trajectories for the case of unfiltered velocity reference (grey dashed line), ZVD input shaping velocity filtered reference (black solid line) and notch filtered velocity reference, with $\hat{a} = 1.4$ (dashed red line).

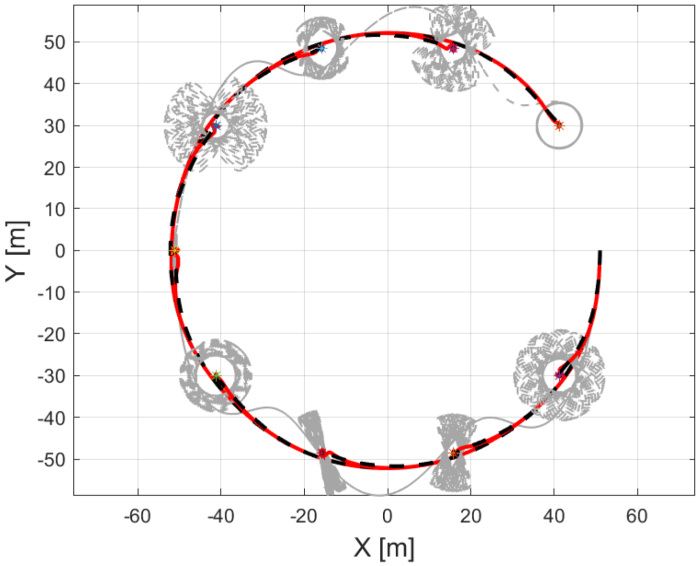


Figure 2.28: Payload trajectories for the case of unfiltered velocity reference (grey dashed line), ZVDD input shaping velocity filtered reference (black solid line) and notch filtered velocity reference, with $\hat{a} = 0.9$ (dashed red line).

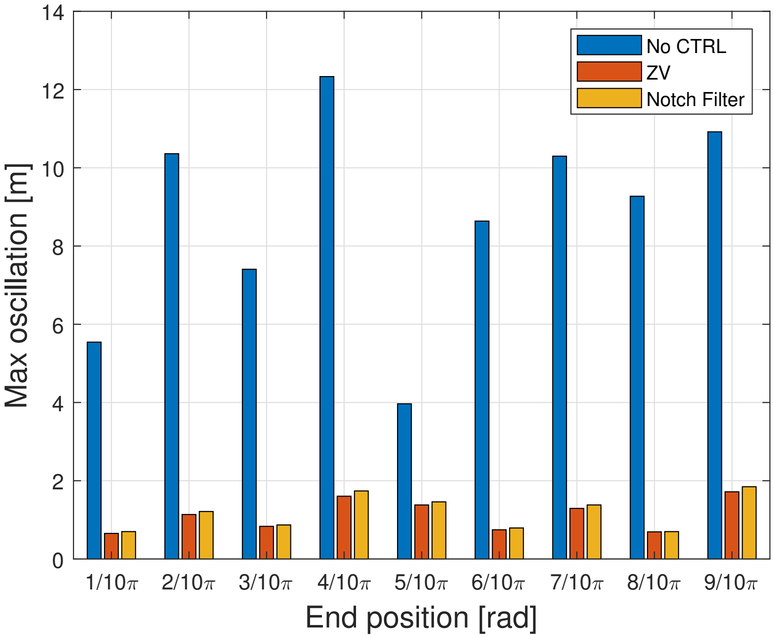


Figure 2.29: Maximum residual oscillation of the payload during different rotation manoeuvres, for the case of unfiltered velocity reference, ZV input shaping and notch filtering with $\hat{a} = 2.0$.

On the other hand, when more robust approaches are needed, by the comparison in Figure 2.30 and Figure 2.31 notch filtering guarantees less residual oscillations with respect to input shaping approaches. Moreover, changing the robustness of notch filtering is straightforward, as it only depends on the tuning parameter $\hat{a}$. For this reason, for this particular application notch filtering approach is to be preferred.

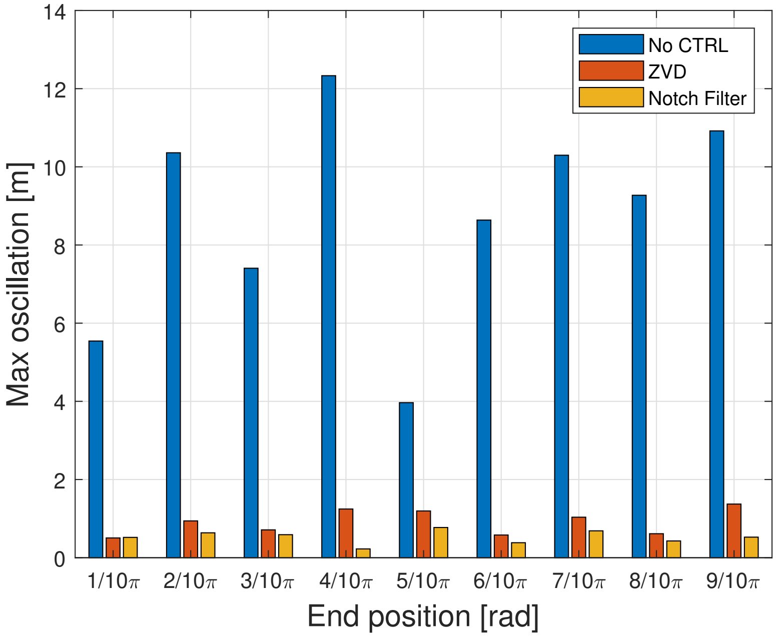


Figure 2.30: Maximum residual oscillation of the payload during different rotation manoeuvres, for the case of unfiltered velocity reference, ZVD input shaping and notch filtering with $\hat{a} = 1.4$.

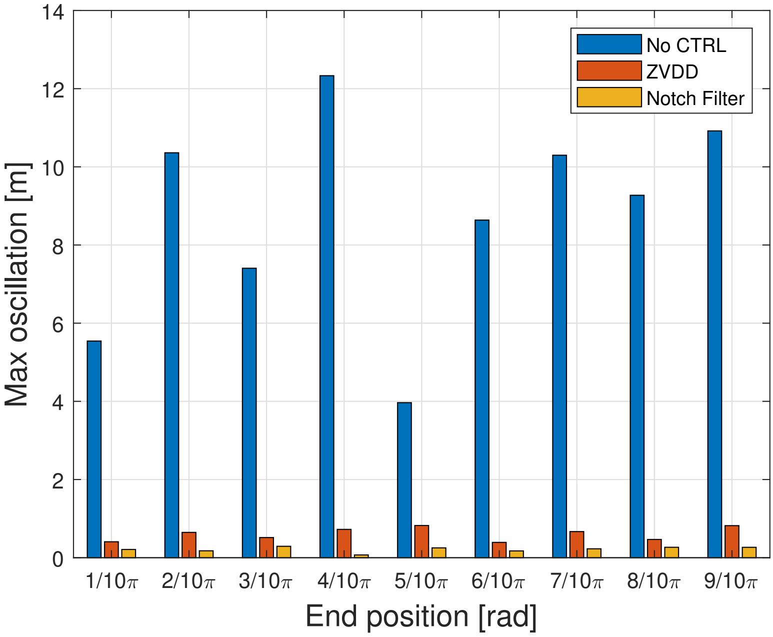


Figure 2.31: Maximum residual oscillation of the payload during different rotation manoeuvres, for the case of unfiltered velocity reference, ZVDD input shaping and notch filtering with $\hat{a} = 0.9$.

2.4 Closed-loop control

In the previous section, open-loop control approaches for the antisway control of industrial cranes have been evaluated. The evaluation of this approaches have been mainly focused on their robustness with respect to errors in the model parametrization in terms of residual oscillation. As open-loop approaches are not able to cope with the reduction of the residual oscillation, it is important to reduce it as much as possible, hence the robust input shaping filters. In some applications (offshore cranes, shipyard cranes etc.), external disturbances such as, for example, wind and sea waves, may introduce oscillations that directly affect the payload. These oscillations can not be reduced in any way with open-loop approaches that, on the contrary, by introducing delays between the operator command and the cart motion, may negatively affect the ability of expert operators in manually reducing payload oscillations.

For this reason, and with the lowering in costs of modern cameras and the capability of modern control hardware to cope with computationally requiring tasks such as image recognition, closed-loop control approaches for the control of industrial cranes are under study, and some of them are already available as commercial solutions.

Closed-loop control techniques are more robust to errors in the model parameters and to external disturbances with respect to open-loop approaches [49]. Nonetheless, the implementation of closed-loop techniques for such powerful and expensive machines arises issues related to stability and noise problems; moreover, the implementation of closed-loop approaches in industrial cranes, which have been typically controlled in open loop, may disrupt the control performance when used in conjunction with expert operators, who are not accustomed to the sometimes counterintuitive cart motions generated by the control [75].

In the literature, various closed-loop control technique have been proposed for the control of industrial cranes, and among them linear control, sliding mode control, state feedback control, Model Predictive Control and adaptive control are worth mentioning. A comprehensive overview on the closed-loop control techniques available in the literature has been presented in [57].

2.4.1 Full state feedback

Full state feedback is a practical approach to the control of overhead cranes. Since the problem when controlling these systems are the highly underdamped complex poles on their transfer functions, which result in highly underdamped oscillations of the payload, state feedback control aims to relocate the poles of the closed-loop transfer function, which will be positioned as opportunely damped ones.

Indeed, with full state feedback control, under some assumptions, one is able to arbitrarily place the poles of the controlled system. For this reason full state feedback is also referred to simply with the more generic name of pole placement. Such assumptions are satisfied by overhead cranes, as can be demonstrated by computing the rank of the controllability matrix for both the simple and double pendulum models in Chapter 2.1.

As already mentioned, one of the drawbacks of using closed-loop approaches for the control of cranes is that the states of the system are not measured on off-the-shelf industrial cranes, and adding ad-hoc sensors for the measure of the sway angle is usually unpractical. In these cases, full state feedback control can be applied by reconstructing the states of the system under control by using an observer. Again, by checking the rank of the observability matrix of the overhead crane models, overhead cranes result in fully observable systems, so that the use of an observer for computing the sway angle of the payload is possible.

The scheme for the state feedback control of a generic system is shown in Figure 2.32. The measured state vector $\boldsymbol{x}$ is multiplied by a vector of coefficients $\boldsymbol{x}$ and summed to the input of the system.

The coefficients of $\boldsymbol{K}$ are computed in order to obtain the desired closed-loop poles, knowing that the poles of the closed-loop systems are given by

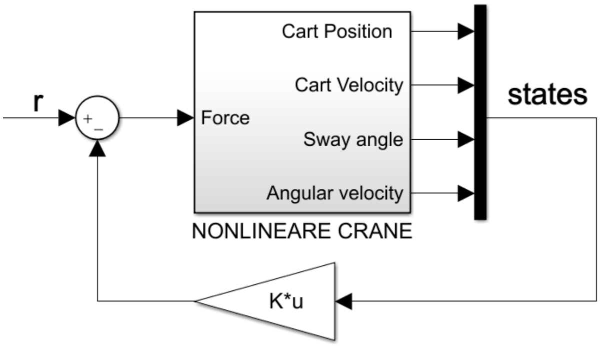


Figure 2.32: Full state feedback control scheme.

the eigenvalues of the matrix $\mathcal{A} - \mathcal{B}\boldsymbol{K}$.

As already mentioned, when it is not possible to directly measure the states of the system, an observer can be exploited for the reconstruction of the full state vector $\boldsymbol{K}$. The scheme representing the state feedback control of a system with an observer is shown in Figure 2.33.

In the scheme, an approximated model of the system is simulated, receiving the same input as the real system. The internal states evolution is corrected by means of the feedback between of the difference between the real measured output and the simulated output. It must be noted that, for the simulated system, the output of the system can be chosen by selecting an appropriate matrix $\mathcal{C}$, so that it is composed by a linear combination of only measurable states, as sometimes the desired output of the system is not measurable.

The scalar values of $\boldsymbol{L}$ are chosen to consider the dynamics with which the observer states follow the measured states depends on its value, as this dynamics is defined by the eigenvalues of the matrix $\mathcal{A} - \boldsymbol{L}\mathcal{C}$. For the choice of this dynamics, a trade-off is necessary between a too-slow dynamics which would introduce delays on the feedback line, and a too-fast dynamics, which would amplify the effects of measurement noise.

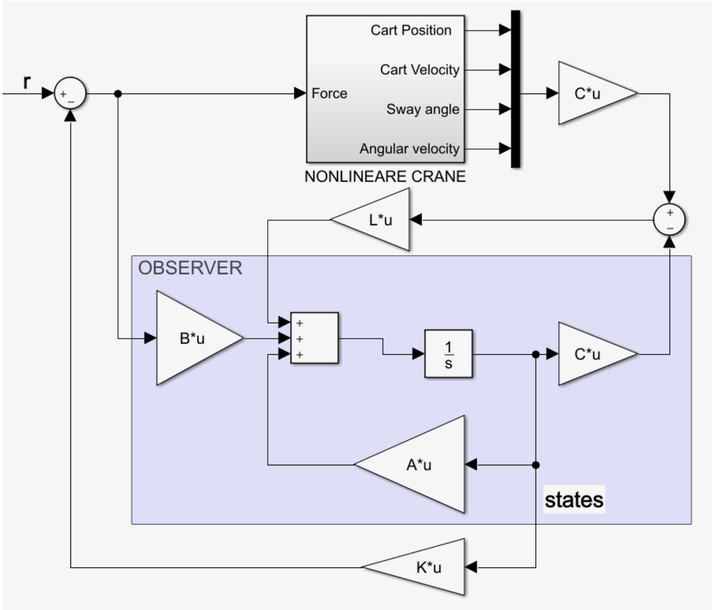


Figure 2.33: Full state feedback control scheme with state observer.

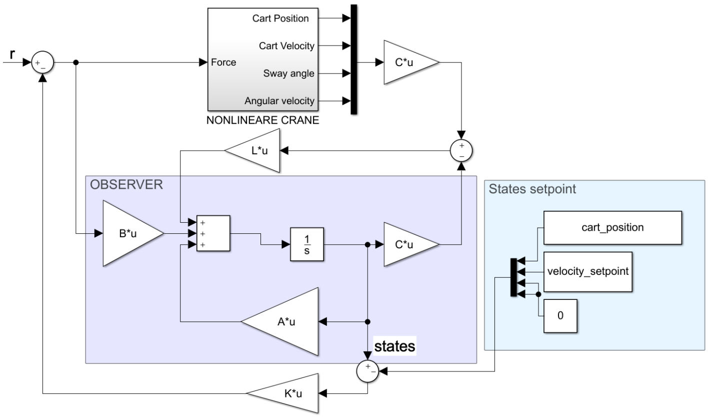


Figure 2.34: Full state feedback control scheme with state observer and states set point.

In order to adapt state feedback control to the control of overhead cranes, a setpoint is added on the scheme in Figure 2.33 for every state. While the cart position and velocity reference are easily obtained (for example by defining a three traits law of motion), the setpoints for the sway angle and the sway angular velocity are always set to zero, as the oscillations should be ideally always reduced during and after the manoeuvre. The resulting scheme is shown in Figure 2.34.

It must be stressed that the control scheme in Figure 2.34 can be applied for the control of OIL overhead cranes, as the cart velocity reference can be easily defined as

$$\boldsymbol{w_1} := \begin{cases} \alpha & \text{if forward jog button is pressed,} \\ -\alpha & \text{if backward jog button is pressed,} \\ 0 & \text{if no button is pressed,} \end{cases} \tag{2.85}$$

where α is a predefined velocity reference value, and the cart position reference can be set as the actual position, in order to have a prompt stop of the system when the button is released.

The application of state feedback control to overhead cranes will be demonstrated practically in the following MIL and HIL tests.

2.4.1.1 MIL validation

The simulated model of the overhead crane is obtained by substituting the parameters in Table 2.6, and simulating the nonlinear dynamics of the overhead crane using Simscape Multibody. The coefficients of the feedback vector $\boldsymbol{K}$ are computed, based on the model with correct parameters described by the values in the second column of Table 2.6, in order to obtain a closed-loop system with poles in

Table 2.6: Parameters of the overhead crane model used in simulations.

Model data		
Parameter	Exact parameters	Parameters with errors
m_C	38 [kg]	45 [kg]
C_C	20 [Ns/m]	25 [Ns/m]
C_p	0.1 [Nms/rad]	0.2 [Nms/rad]
l	1.5 [m]	1.3 [m]
m_p	10 [kg]	12 [kg]

$$[-10, \ \ -20, \ \ -1+0.1i, \ \ -1-0.1i]$$

and by considering the linear model in (2.22). The resulting feedback gain vector is

$$\boldsymbol{K} = [1174, \ \ 2480, \ \ -12632, \ \ 1926]\,.$$

For the purpose of simulating the control of an industrial overhead crane, only cart velocity and position are supposed to be measurable, and the values of the sway angle and sway angular velocity are reconstructed by means of an observer, whose model is based on the estimated parameters in

Table 2.6, where errors have been introduced. Following the rule of thumb, without having information regarding the noise affecting the measure of the states, the poles of $\mathcal{A} - \boldsymbol{L}\mathcal{C}$ have been placed one decade faster than the closed-loop dynamics given by $\mathcal{A} - \mathcal{B}K$, while avoiding imaginary or multiple poles, so that the poles have been arbitrarily placed in

$$[-100, \;\; -200, \;\; -10, \;\; -11] \, .$$

The resulting vector $\boldsymbol{L}$ is

$$\boldsymbol{L} = [-335240, \;\; 335560, \;\; -51640, \;\; 225250]^T \, .$$

The velocity reference is shown in Figure 2.35, along with the cart and payload velocities. The force acting on the cart is shown in Figure 2.36. While the velocity of the payload does not reach the reference, a steady state error in the velocity is acceptable when the crane is directly actuated by an operator. It should also be noted that, due to the impossibility of including constraints during the control, the initial and final force peaks are hardly acceptable for industrial actuators, which should be greatly oversized for the application in order to guarantee the required peaks.

2.4.1.2 HIL validation

State feedback control have been implemented on Gefran ADV200 industrial drive in order to test the methodology on the HIL setup described in Chapter 1.4.4. The control application runs with a cycle time of $T_s = 0.001$ [s]. The Simulation Motor simulates the response of the overhead crane described by the parameters in Table 2.6. The limits of the actuator have been imposed to be

$$F_{max} = 500 \text{ [N]}$$

The feedback vector $\boldsymbol{K}$ and the innovation vector $\boldsymbol{L}$ are the same ones as those used for the MIL validation.

The velocity reference is shown in Figure 2.37, along with the cart and payload velocities. The force acting on the cart is shown in Figure 2.38.

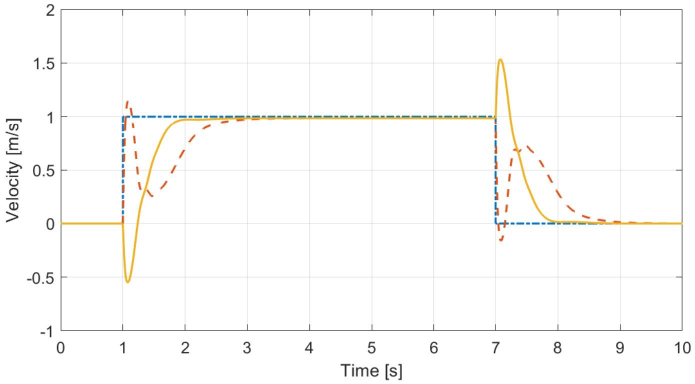


Figure 2.35: Full state feedback MIL validation. Velocity reference (dash-dot blue line, cart velocity (dashed orange line) and payload velocity (solid yellow line).

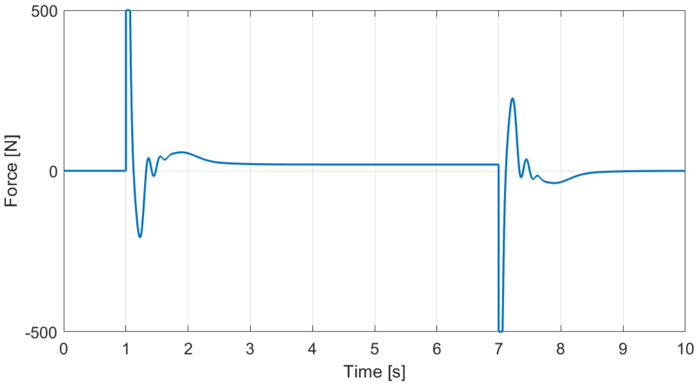


Figure 2.36: Full state feedback MIL validation. Force action on the cart.

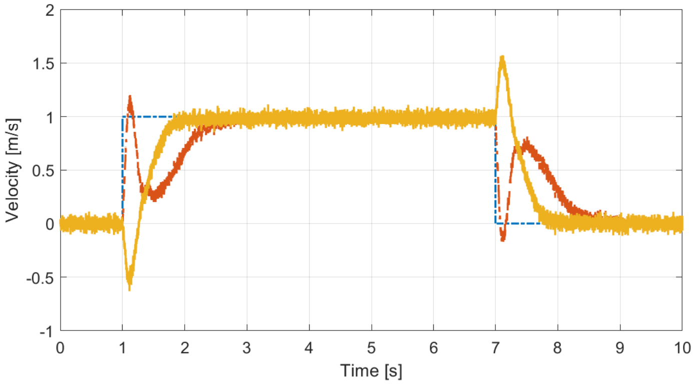


Figure 2.37: Full state feedback HIL validation. Velocity reference (dash-dot blue line, cart velocity (dashed orange line) and payload velocity (solid yellow line).

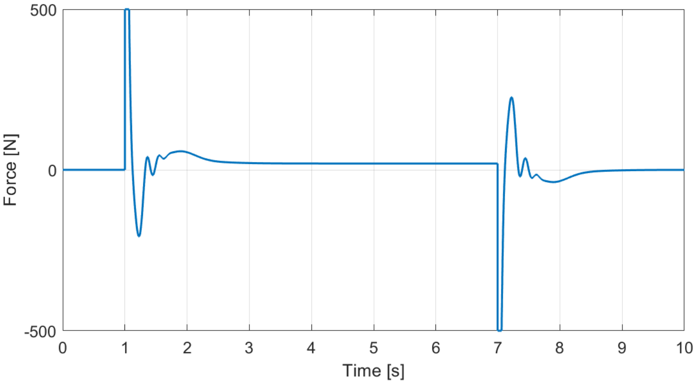


Figure 2.38: Full state feedback HIL validation. Force action on the cart.

It can be seen that the results are comparable with the tests on the MIL validation test. Force limit of 500 [N] does not limit significantly the overall performance of the control system.

2.4.2 Model Predictive Control

Although it has been shown in Chapter 2.4.1 that state feedback control is a valid choice for the control of OIL overhead cranes, it is not capable of coping with systems with hard constraints. When working with such systems and state feedback, the only option is to decrease the bandwidth of the controlled system until all the constraints are satisfied, thus obtaining sub-optimal performance. An advanced control approach that is focused on the optimality of the response given some constraints is Model Predictive Control (MPC).

MPC has been identified as one of the more promising approach for the closed-loop control of cranes. This is motivated by the increasing computational capability of modern off-the-shelf controllers. While MPC was in origin confined to process control applications, due to the heavy computational burden bound to the solution of constrained quadratic problems, modern controllers can now afford to run MPC at a cycle time which is suitable for the control of mechatronic systems.

MPC brings together the intrinsic robustness of closed-loop techniques and the ability to respect hard constraints typical of optimal control approaches. Although solutions for the closed-loop control of cranes are indeed present in the literature, few of them address the problem from an industrial point of view. In fact, most of them consider the problem as a positioning problem, defining an offline computed trajectory and relying on the closed-loop control in order to follow this trajectory. While this approach is appropriate to fully automated cranes, in typical industrial applications the end position is usually not known in advance with sufficient precision, and the operator acts directly on the cart in an incremental fashion until the desired position is reached, and then he eventually act on the cart in order to dampen pay-

load oscillations that have come into being.
In this sense, Operator-In-the-Loop (OIL) cranes are velocity-controlled systems. In fact, when the operator presses a button, the cart reaches a predefined velocity, and when the button is released the cart stops.

MPC has here been applied for the control of OIL overhead cranes. First, an MPC controller acting directly on the torque moving the cart has been developed. Then, by using the cart velocity control loop which is already present in industrial cranes, the model has been simplified and a MPC-PID control scheme has been proposed, which results in increased robustness w.r.t. nonlinearities like friction in the sliding cart. Finally, in order to cope with the varying cable length during hoisting manoeuvres, different linear and nonlinear approaches have been tested and compared in order to identify the best solution to the problem of simultaneous hoisting and travelling manoeuvres for overhead cranes.

MPC is a predictive technique that exploit a given model of the system in order to know in advance its response to a vector of future inputs $\boldsymbol{U}$, which is defined as the vector of c future inputs, where c is the control horizon measured in number of controller cycles., each of them kept constant along the controller cycle time.
The predicted output $\boldsymbol{Y}$ is a vector of p values, where p is the predictive horizon measured in number of controller cycles. For the case of Operator In the Loop cranes, the velocity reference $\boldsymbol{w_1}$ can considered as constant along the prediction horizon p, as it is not possible to predict the future behaviour of the operator. The velocity reference can therefore be defined as done for the case of state feedback, that is, as in (2.85).
The input vector $\boldsymbol{U}$ is computed at each control period by minimizing a cost function, with weights on the predicted input vector $\boldsymbol{U}$, on the predicted error between the velocity reference and the velocity of the payload, and on other linear combinations of the system states.

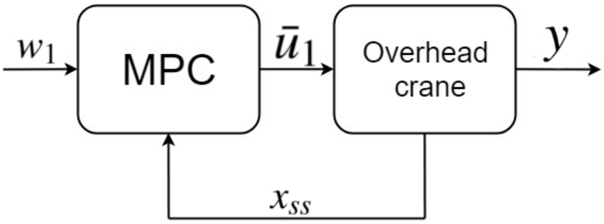


Figure 2.39: MPC control scheme.

2.4.2.1 MPC for overhead cranes

The first proposed controller is based on the model of the crane which has the force acting on the cart as input of the system and as output the velocity of the payload. The reason for considering the force acting on the cart as the input is motivated by the fact that the cart is actuated by a motor, for which the simpler and most immediate control mode is the torque control. The force acting on the cart is than related to the motor torque by a constant factor that is given by the transmission.
The model to be considered is the one defined by (2.22). The scheme of the proposed MPC approach is shown in Figure 2.39. The matrix $\mathcal{C}$ of system (2.22) can be modified, in order to obtain and weight the states of the system, as

$$\mathcal{C} = \begin{pmatrix} \boldsymbol{C_0}^T & \boldsymbol{C_1}^T & \boldsymbol{C_2}^T & \boldsymbol{C_3}^T \end{pmatrix}^T \tag{2.86}$$

with

$$\begin{aligned} \boldsymbol{C_0} &= \begin{pmatrix} 1 & 0 & l & 0 \end{pmatrix} \\ \boldsymbol{C_1} &= \begin{pmatrix} 0 & 1 & 0 & l \end{pmatrix} \\ \boldsymbol{C_2} &= \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \\ \boldsymbol{C_3} &= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Doing so, the output y is now $\boldsymbol{y} = (y_0, y_1, y_2, y_3)^T$ where y_i are, respectively, payload position, payload velocity, sway angle angular velocity $\dot{\theta}_1$ and cart velocity $\dot{x}_c$.

The predictive equations of the outputs at instant k can be written in the form

$$\boldsymbol{Y_1} = \mathcal{F}_1\, \boldsymbol{x}(k) + \mathcal{G}_1\, \boldsymbol{U} \tag{2.87}$$

$$\boldsymbol{Y_2} = \mathcal{F}_2\, \boldsymbol{x}(k) + \mathcal{G}_2\, \boldsymbol{U} \tag{2.88}$$

$$\boldsymbol{Y_3} = \mathcal{F}_3\, \boldsymbol{x}(k) + \mathcal{G}_3\, \boldsymbol{U} \tag{2.89}$$

with

$$\boldsymbol{Y_1} := (y_1(k+1), \dots, y_1(k+p))^T \in \mathbb{R}^p,$$

and $\boldsymbol{Y_2}$, $\boldsymbol{Y_3}$ similarly, $\boldsymbol{U} := (\bar{u}_1, \dots, \bar{u}_c)^T \in \mathbb{R}^c$ and $\boldsymbol{x}(k) \in \mathbb{R}^4$ is the current state vector.

The matrices $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3 \in \mathbb{R}^{p\times 4}$ and $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \in \mathbb{R}^{p\times c}$ are the predictive matrices, calculated from the system model matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ as in [58]. At each control cycle, the MPC controller solves an Optimal Control Problem (OCP) that is defined as a Quadratic Program (QP) that minimizes the squared norm of the difference between the predicted outputs and their references. For the case of crane control, the payload velocity reference has been defined in (2.85). Moreover, in order to avoid excessive oscillations during the manoeuvre , also the angular velocity should be kept as small as possible. At last, in order to avoid chattering behaviours, the force input is also weighted.
The OCP is then defined as

$$\begin{aligned} \text{minimize} \quad & \left\| \boldsymbol{Y_1} - w_1 \right\|^2 + \lambda_2 \left\| \boldsymbol{Y_2} \right\|^2 + \lambda_u \left\| \boldsymbol{U} \right\|^2 \\ \text{subject to} \quad & \dot{\boldsymbol{x}}_{\min} \le \boldsymbol{Y_3} \le \dot{\boldsymbol{x}}_{\max} \\ & u_{\min} \le \boldsymbol{U} \le u_{\max} \end{aligned} \tag{2.90}$$

where λ_2 is the weighting factor on the angular velocity of the sway angle and λ_u is the weighting factor on the input force. As the primary goal of the controller is the tracking of the velocity reference, the introduced weighting factors should be small enough.
In the OCP (2.90), the velocity reference $\boldsymbol{w_1}$ of (2.85) is kept constant

along the predictive horizon. This choice derives from the impossibility of predicting in advance the future behaviour of the operator, that could indeed release the button on the very next control instant. For this unpredictability of the future velocity reference, the use of a reduced predictive horizon is also motivated.

Problem (2.90) can be written in standard quadratic form with respect to the control vector U as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\boldsymbol{U}^T \mathcal{M}\,\boldsymbol{U} + \boldsymbol{N}^T \boldsymbol{U} \\ \text{subject to} \quad & \mathcal{E}\,\boldsymbol{U} \leq \boldsymbol{D} \\ & u_{\min} \leq \boldsymbol{U} \leq u_{\max} \end{aligned} \tag{2.91}$$

where:

$$\mathcal{M} := \mathcal{G}_1^T \mathcal{G}_1 + \lambda_2 \mathcal{G}_2^T \mathcal{G}_2 + \lambda_u \mathcal{I}_c \tag{2.92}$$

$$\boldsymbol{N} := \mathcal{G}_1^T (\mathcal{F}_1\, \boldsymbol{x}(k) - w_1) + \lambda_2 \mathcal{G}_2^T \mathcal{F}_2\, \boldsymbol{x}(k) \tag{2.93}$$

$$\mathcal{E} := \left(\mathcal{G}_3^T,\, -\mathcal{G}_3^T \right)^T \tag{2.94}$$

$$\boldsymbol{D} := \left(\dot{\boldsymbol{x}}_{\max} - \mathcal{F}_3\, \boldsymbol{x}(k),\, -\dot{\boldsymbol{x}}_{\min} + \mathcal{F}_3\, \boldsymbol{x}(k) \right)^T \tag{2.95}$$

where $\mathcal{I}_c$ denotes the identity matrix.

While the steady state velocity reference α could be a parameter set by the operator, a method for automatically link α with the overshoot in the payload position when the jog button is released is here presented. The overshoot in the payload position is a relevant parameter, as it directly affects a precise positioning of the payload and the safety of the overall system.

Once the maximum acceptable overshoot value $\Delta x_{\max}$ has been defined, the maximum value of α so that the overshoot in position Δx does not overcome $\Delta x_{\max}$ must be found.

In order to do so, the following optimization problem must be solved:

$$\begin{aligned} \text{maximize} \quad & \alpha \\ \text{subject to} \quad & \Delta x(t) \leq \Delta x_{\max} \;\; \forall\, t \geq t_B \end{aligned} \tag{2.96}$$

where t_B is the time instant when the operator releases the jog button.
To solve (2.96) numerically, assume that: (i) the predictive horizon p is long enough to take into account the maximum position overshoot, (ii) when the operator releases the button, the system is at steady-state, that is, $\boldsymbol{x}(t_B) = (0, \alpha, 0, 0)' =: \boldsymbol{x}_0$.
The predicted overshoot can be therefore calculated as:

$$\Delta \boldsymbol{X} = \mathcal{F}_0\, \boldsymbol{x}_0 + \mathcal{G}_0\, \boldsymbol{U}_0 \tag{2.97}$$

where $\Delta X := (\Delta X(k+1), \dots, \Delta X(k+p))^T$, $\mathcal{F}_0 \in \mathbb{R}^{p\times 4}$ and $\mathcal{G}_0 \in \mathbb{R}^{p\times c}$ can be computed from $\mathcal{A}, \mathcal{B}, \mathcal{C}_0$, similarly to the predictive matrices in (2.87), and $\boldsymbol{U}_0$ is calculated as the solution of (2.91), with $\boldsymbol{x}(k) = \boldsymbol{x}_0$. Thus, problem (2.96) becomes:

$$\begin{aligned}
\text{maximize} \quad & \alpha \\
\text{subject to} \quad & \mathcal{F}_0\, \boldsymbol{x}_0 + \mathcal{G}_0\, \boldsymbol{U}_0 \leq \Delta x_{\max} \\
& \boldsymbol{x}_0 \ = (0, \alpha, 0, 0)^T \\
& \boldsymbol{U}_0 \ = \ \begin{array}{ll} \text{argmin} & \scriptstyle \boldsymbol{U}^T \mathcal{M} \boldsymbol{U} + \boldsymbol{N}^T \boldsymbol{U} \\ \text{s.t.} & \scriptstyle \mathcal{E} \boldsymbol{U} \leq \boldsymbol{D} \\ & \scriptstyle u_{\min} \leq \boldsymbol{U} \leq u_{\max} \end{array}
\end{aligned} \tag{2.98}$$

Problem (2.98) can now be solved by using an iterative method, such as the secant method, to find the root of the equation $f(\alpha) = 0$, where $f : [0, \dot{x}_{\max}] \subseteq \mathbb{R} \to \mathbb{R}$, $\alpha \mapsto \max(\Delta X(\alpha)) - \Delta x_{\max}$.

MIL validation

The proposed method has been tested by using the MIL approach described in Chapter 1.4.1. To validate the proposed approach, a model of the system with the parameters in Table 2.6 has been simulated using Amesim with the scheme shown in Figure 2.40, while running the control approach on Simulink in co-simulation. The validation follows the guidelines for MIL

validation, and follows the scheme in Figure 1.4.
For the purpose of comparison, both the proposed MPC approach and the ZV input shaping technique have been implemented, is it is the most widely diffused technique for the control of industrial overhead cranes.

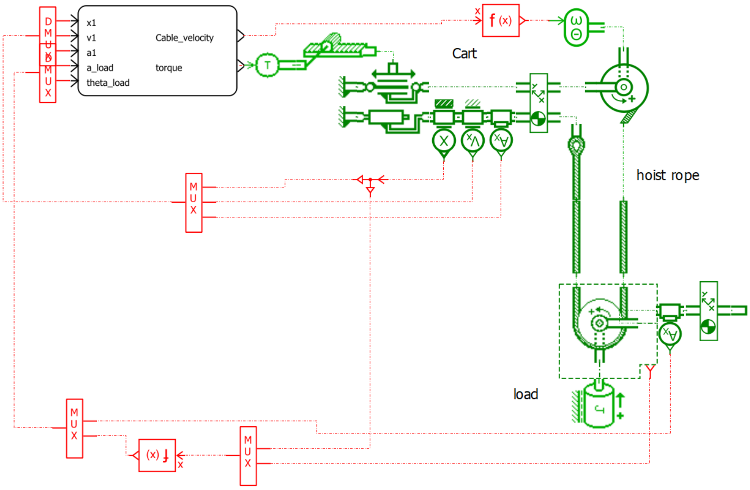


Figure 2.40: Simcenter Amesim model of the overhead crane

Two different situations have been tested. First, a perfect knowledge of the model parameters is assumed. Then, errors in the model parameters have been introduced. We expect the performance of the ZV input shaping control to strongly degrade in the second situation, while MPC, being a closed-loop technique, should be more robust with respect to changes in the model parameters.
The parameters of the simulated systems are shown in Table 2.6, along with the estimated (wrong) parameters used for the second trial.
The limits for the simulated overhead crane imposed in (2.90) are set to

$$\dot{x}_{\max} = -\dot{x}_{\min} = 1.2\ [\mathrm{m/s}] \qquad u_{\max} = -u_{\min} = 150\ [\mathrm{N}] \tag{2.99}$$

while the weighting parameters in (2.90) are set to

$$\lambda_2 = 5.0 \cdot 10^{-4} \qquad \lambda_u = 1.0 \cdot 10^{-6}. \tag{2.100}$$

The values in (2.100) have been tuned by trial and error method.

The maximum overshoot in position Δx_{max} has been set equal to 0.5 [m], and the resulting set point velocity value α results from (2.98) in $\alpha = 0.985$ [m/s]. The control sampling time is set to $T_s = 0.01$ [s]; the control horizon is set to $c = 50$ and the predictive horizon to $p = 100$.
The reference velocity is set to α at time instant $t = 1.0$ [s], and it is set back to zero at $t = 7.0$ [s].

Nominal case

In the nominal case, that is, when the model parameters are supposed to be perfectly known, the MPC approach is compared to the ZV input shaping. The force acting on the cart for both techniques is shown in Figure 2.41. As expected, the MPC approach takes into account explicitly the constraints on the maximum force to be applied to the cart.
Figure 2.42 shows that also the velocity of the cart is successfully contained within its limits.
In Figure 2.43 the velocity of the cart is shown for both MPC and input shaping approaches. Due to the aggressive tuning of the parameters in (2.100), the cart velocity has a more aggressive behaviour with respect to the velocity generated by the input shaping approach. Nonetheless, it should be noted that the velocity of the cart is always positive, meaning that the cart does not even stop during the manoeuvre .
In Figure 2.44 the sway angle θ is shown. Due to the perfectly known parameters of the system, the input shaping approach result in reduced oscillations during the acceleration and deceleration phase, and in a very small residual oscillation, which is generated by the non-modelled nonlinearities of the simulated overhead crane. On the contrary, the MPC approach yields to an increased sway angle peak, and an increased settling time of the sway angle. Nonetheless, the oscillation is contained and controlled.

In Figure 2.45, the position overshoots are compared. Position over-

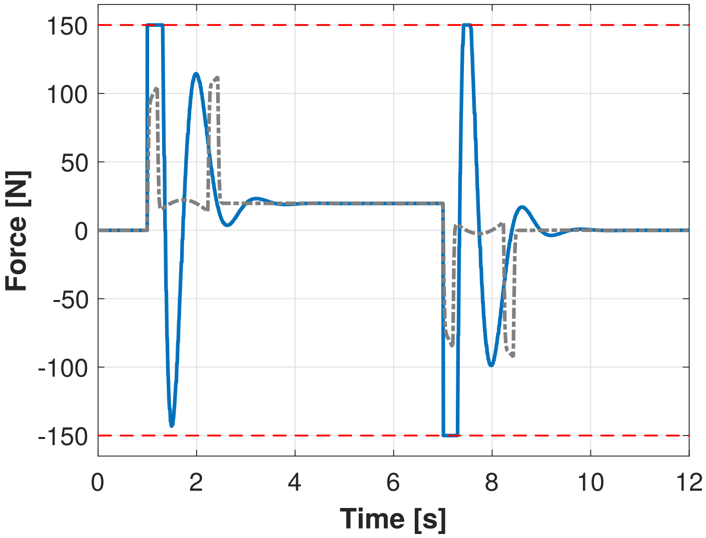


Figure 2.41: Force acting on the cart. Solid blue: proposed method. Dashed-dotted grey: ZV input shaping. Dashed red: cart force limits.

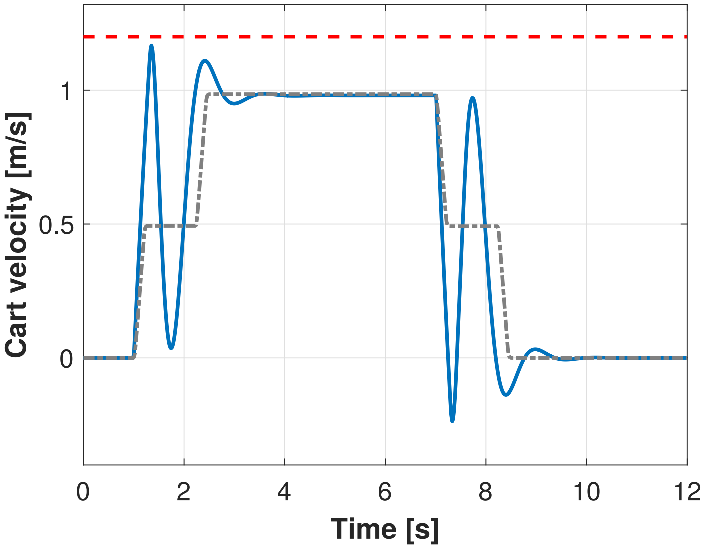


Figure 2.42: Velocity of the cart during the control, for both MPC (blue solid line) and input-shaping (grey dash-dot line) in the case of exact model parameters. The dashed red line represents the velocity limit imposed to the cart.

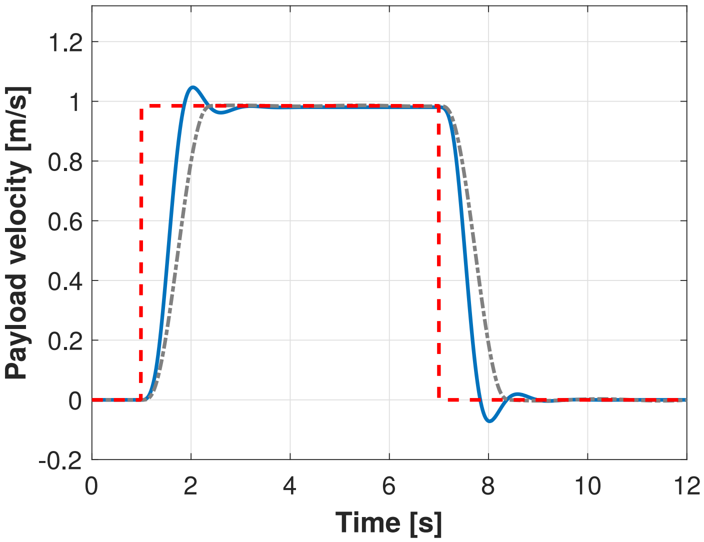


Figure 2.43: Velocity of the payload with MPC (blue solid line) and input-shaping (grey dash-dot line). In red dashed line the velocity reference.

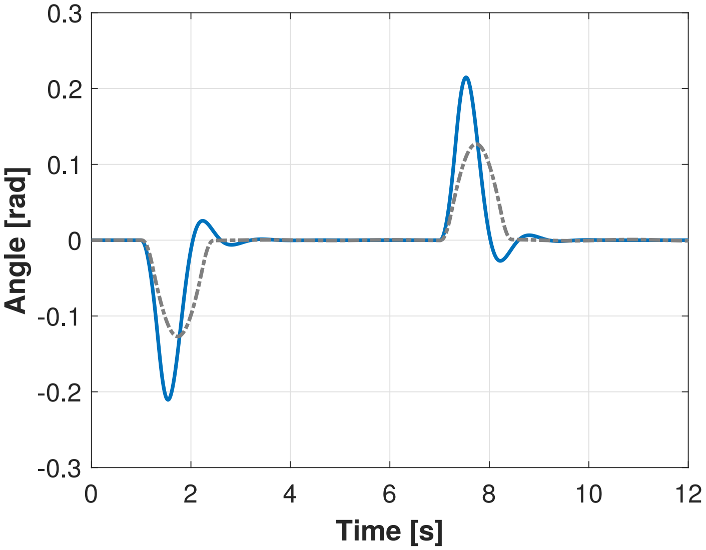


Figure 2.44: Swing angle of the payload in the case of exact model parameters, for both MPC (blue solid line) and input-shaping (grey dash-dot line).

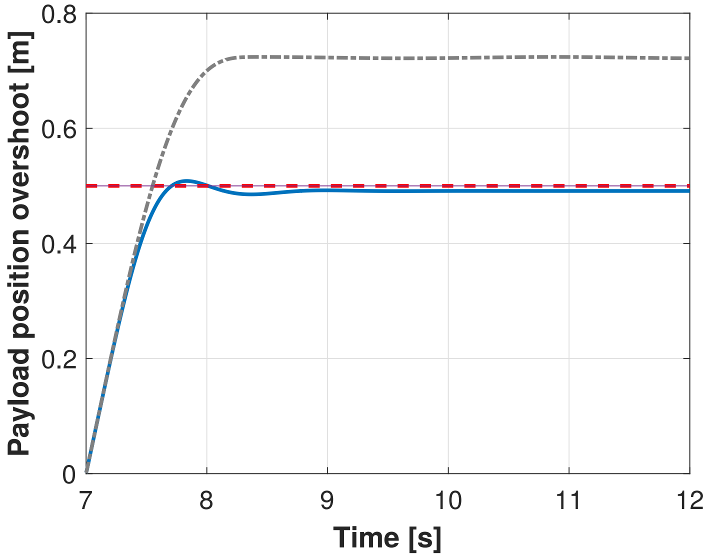


Figure 2.45: Position overshoot of the payload in the case exact model parameters. Solid blue: proposed method. Dashed-dotted grey: ZV input shaping. The red dashed line is the imposed limit.

shoots are measured as the position of the payload in excess to its position when the reference velocity $\boldsymbol{w_1}$ is set to 0, that is, when the operator releases the jog button.
As expected by having found the value of α from (2.98), the position overshoot of the MPC approach is below the set value of Δx_{max}. The IS approach shows an increased drifting in the payload position after the jog button is released, which is not only an issue in terms of safety, but has also be demonstrated to be counterintuitive for the operator.

Errors in the model parameters

When applying model based control strategies in industrial applications, modelling errors can affect the effectiveness of the methods. For this reason,

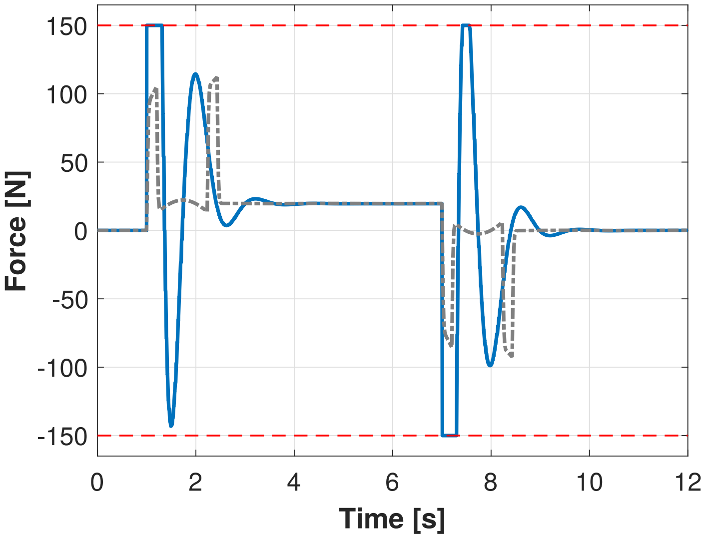


Figure 2.46: Force acting on the cart in the case of errors in the model parameters. Solid blue: proposed method. Dashed-dotted grey: ZV input shaping. Dashed red: cart force limits.

the simulation will consider a simulated system with the same parameters as before, while the control techniques will be based on a model with the parameters with errors in Table 2.6.

In this case, the force acting on the cart, shown in Figure 2.46 does not differ significantly from the previous case. It is worth stressing that the MPC approach exploits the force constraints of the actuator without exceeding its limits.

In Figure 2.47, due to the errors in the model parametrization, the cart velocity for the MPC case exhibit an oscillatory trajectory.

In Figure 2.48, the payload velocity using both control techniques is compared to the reference velocity. While both approaches generate oscillations, input shaping approach can not compensate for the residual oscillations gen-

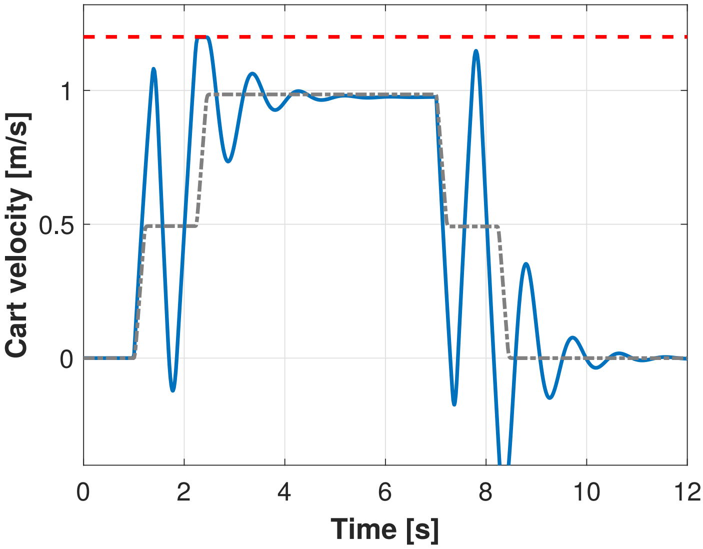


Figure 2.47: Velocity of the cart during the control, for both MPC (blue solid line) and input-shaping (grey dash-dot line) in the case of errors in the model parameters. The dashed red line represents the velocity limit imposed to the cart.

erated by the errors in the parametrization of the model. On the contrary, the residual oscillations are rapidly compensated with the MPC approach. This is underlined in Figure 2.49 where the sway angles are compared. The MPC is able to bring the oscillation angle under control despite the introduced errors in the model parameters.

The position overshoot is shown in Figure 2.50. As the optimization problem (2.98) is based on the model with errors on the parameters, when the approach is applied to the simulated system, the position overshoot is slightly more than the imposed value of $\Delta x_{max} = 0.5$ [m].

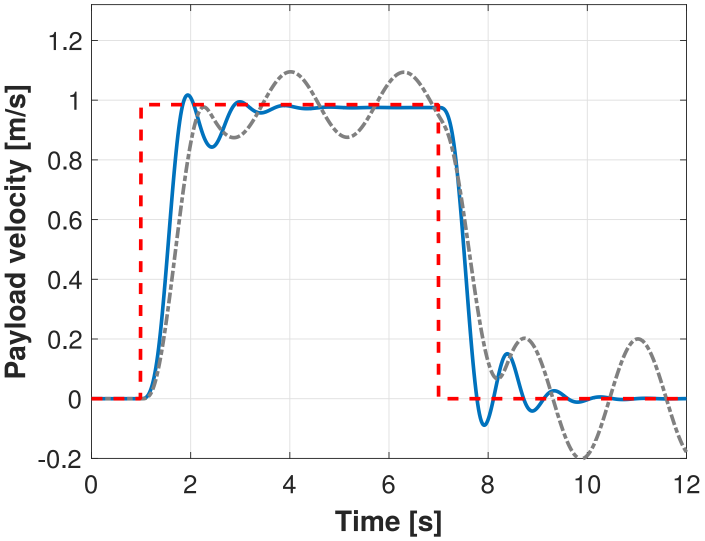


Figure 2.48: Velocity of the payload with MPC (blue solid line) and input-shaping (grey dash-dot line) control in the case of errors in the model parameters. In red dashed line the velocity reference.

With the proposed MPC approach, it has been shown that a closed-loop control of an overhead can fully exploit the limits of the actuator; moreover, the MPC can cope with errors in the model parameters, showing increased

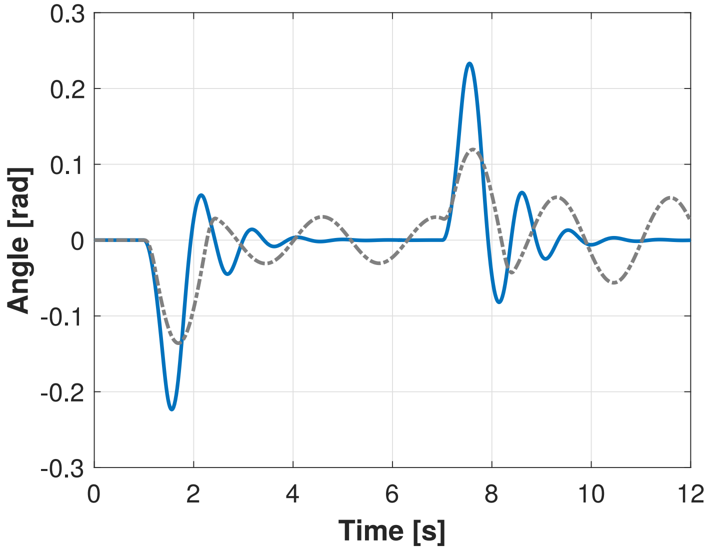


Figure 2.49: Swing angle of the payload in the case of errors in the model parameters, for both MPC (blue solid line) and input-shaping (grey dash-dot line).

robustness with respect to standard open-loop approaches. The operator can select a significant parameter for the definition of the velocity reference value α, which is the position overshoot after the jog button is released. The proposed method for the MPC control of overhead crane has been published as [23].

2.4.2.2 MPC-PID control of overhead cranes

The method proposed in the previous chapter focuses on the MPC control of overhead cranes acting directly on the torque generated by the actuator that moves the crane. The proposed MPC approach is based on a linear model of the system, thus ignoring the nonlinearities (e.g. the friction act-

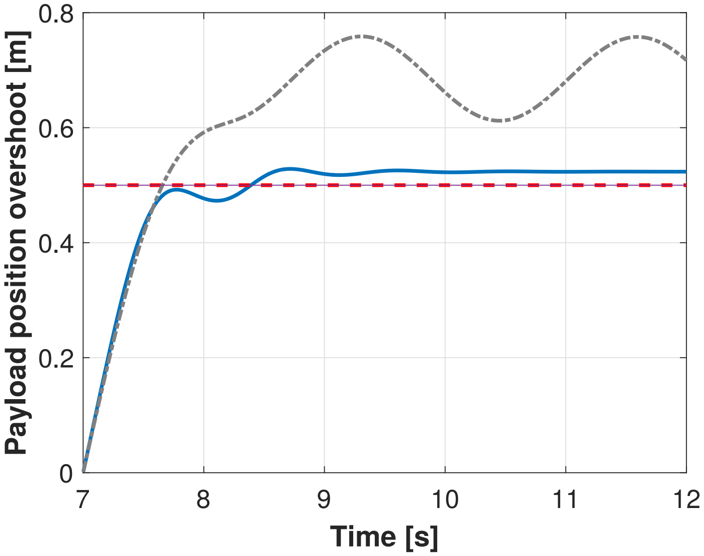


Figure 2.50: Position overshoot of the payload in the case of errors in the model parameters. The red dashed line is the imposed limit.

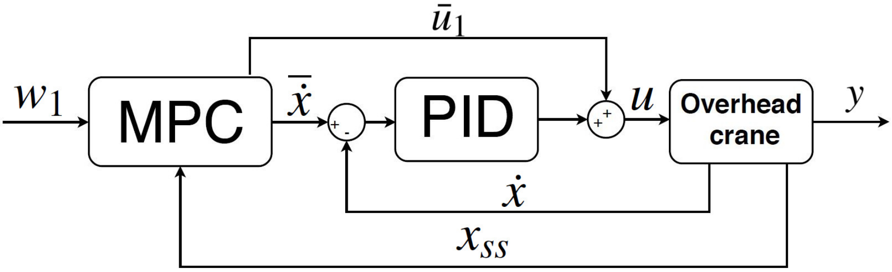


Figure 2.51: Cascade control of the crane, with an internal PID for the velocity control of the sliding cart, and an external MPC controller for the velocity control of the payload.

ing on the sliding cart) that profoundly influence the dynamics of a real overhead crane.

For this reason, the approach has been modified in order to take advantage of the control loops already present on industrial overhead cranes to compensate for some of the nonlinearities that affects the system. In particular, the control of industrial overhead cranes relies on PID controllers for the control of the cart velocity. As the cart itself is not affected by significant oscillating dynamics, its control is easy and the reachable bandwidth can effectively compensate for modelling errors such as errors in the model parameters and unmodelled nonlinearities. A PID controller can therefore be used for the velocity control of the sliding cart, while an external MPC controller acts on the velocity reference of the cart in order to control the payload, thus creating a cascade control of the overhead crane as shown in Figure 2.51.

The outer MPC controller solves, at every control cycle, the same OCP as before, that is, (2.90). The velocity reference to be given to the internal velocity control loop can be computed by applying the first value of the optimal control action array $\boldsymbol{U}(k) = \bar{u}_1$ to the model of the crane described by (2.22). In fact, the future cart velocity can be easily computed as, by

setting the output matrix $\mathcal{C}$ as in (2.86), it corresponds to $\boldsymbol{y}_3(k+1)$.
In order to enhance the performance of the internal loop, the computed optimal force $\bar{u}_1$ is given as a feedforward signal to the actuator. Thus, thanks to the force feedforward that relies on the linear model of the system, the internal PID controller can be tuned for disturbances rejection, as its task is to compensate for nonlinearities, model uncertainties and external disturbances.
It is worth stressing that industrial controllers have built-in torque and position loops, which are typically based on PID structures. Most of them also provide the possibility to assign torque feedforward signals. Therefore, the implementation of the proposed technique on industrial hardware is facilitated by the presence of an underlying well-known control structure.

MIL validation

In order to test the increased performance of the MPC-PID approach w.r.t. the MPC approach proposed in the previous section, both of the approaches have been implemented in Simulink. To validate the proposed approach, a model of the system with the parameters in Table 2.6 has been simulated using Amesim, as done for the previous MIL validation tests. While the model used for the MPC approaches is the linear one described by (2.22) and the exact parameters of Table 2.6, some variations have been inserted in the simulated overhead crane, in order to test the robustness of the techniques. In particular, the mass of the cart has been changed to $m_C = 49.4$ [kg], the friction coefficient has been changed to $C_C = 10$ [Ns/m], and a Coulomb friction has been added on the sliding cart, with a value of $C_0 = 150$ [N]. Moreover, a disturbance in the form of a $F_d = 500$ [N] acts on the cart at time $t = 9.0$ [s], as shown in Figure 2.52.

During the simulation, the jog button is pressed at time $t = 1.0$ [s] and released at time $t = 6.0$ [s]. The reference velocity trajectory is shown in Figure 2.53.

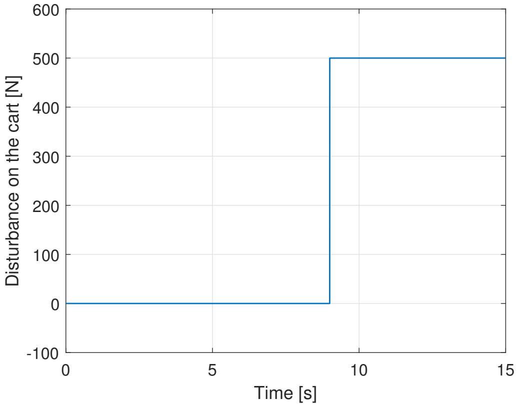


Figure 2.52: Disturbance acting on the force acting on the cart at time $t = 9$ $[s]$.

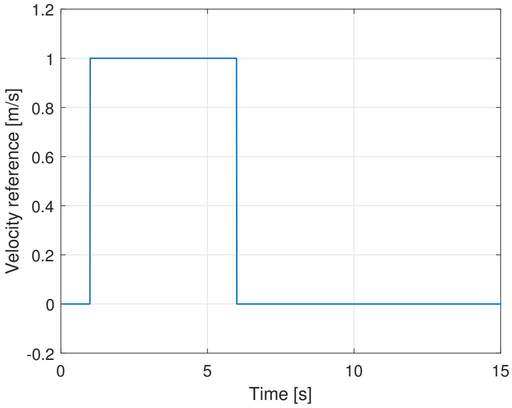


Figure 2.53: Payload velocity reference.

As MPC controllers are computational consuming, the control cycle for the MPC controllers has been set to $T_{MPC} = 0.024$ [s]. The control cycle for the PID controller has been set to $T_{PID} = 0.001$ [s]. The weighting parameters are set to $\lambda_2 = 5.0 \cdot 10^{-4}$, $\lambda_u = 3.0 \cdot 10^{-6}$, with a control horizon $c = 5$ and a predictive horizon $p = 30$. The PID has been tuned, through trial and error, as

$$PID(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right)$$

where

$$\begin{aligned} K_p &= 1800 \\ T_i &= 0.2778 \ [\mathrm{s}] \\ T_d &= 0.0056 \ [\mathrm{s}] \\ N &= 20 \end{aligned}$$

The system constraints have been set to

$$\dot{x}_{\max} = -\dot{x}_{\min} = 1.2 \ \ [\mathrm{m/s}] \tag{2.101}$$

$$u_{\max} = -u_{\min} = 1000 \ \ [\mathrm{N}]. \tag{2.102}$$

In Figure 2.54 the payload velocities obtained with the MPC and MPC-PID approaches are shown. While the acceleration phase is mostly comparable, the settling time is reduced for the case of the MPC-PID control scheme. Moreover, after the stop phase, the payload velocity continues oscillating around zero, with a more significant oscillation for the case of the MPC approach. This effect can be observed also when analyzing the sway angle, shown in Figure 2.55.

The reason for this residual oscillation is to be identified as the Coulomb friction, which introduces a strong nonlinearity in the system. The effect of the Coulomb friction can be observed also in Figure 2.56, where the cart velocity is shown.

Figure 2.57 compares the actuator efforts of the two approaches, which do not differ much from each others.

The proposed method for the MPC-PID control of overhead crane has been published in [24].

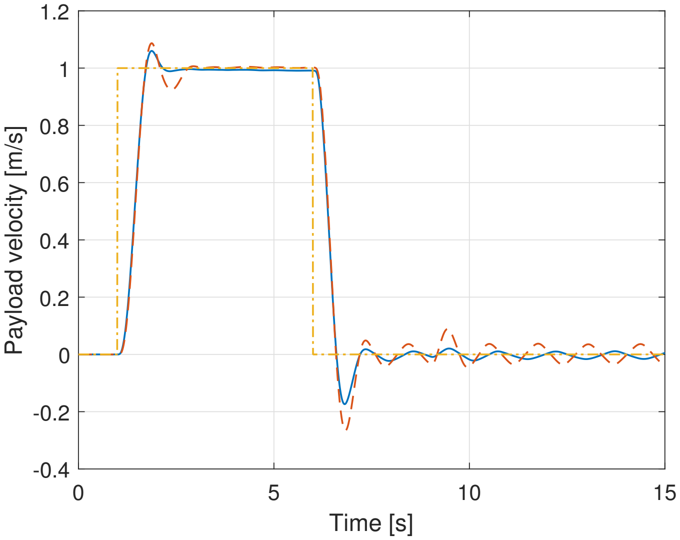


Figure 2.54: Load velocity response of the MPC standard scheme (red dashed line) and the MPC-PID scheme (blue solid line). The velocity reference $\boldsymbol{w_1}$ is the yellow dash-dot line.

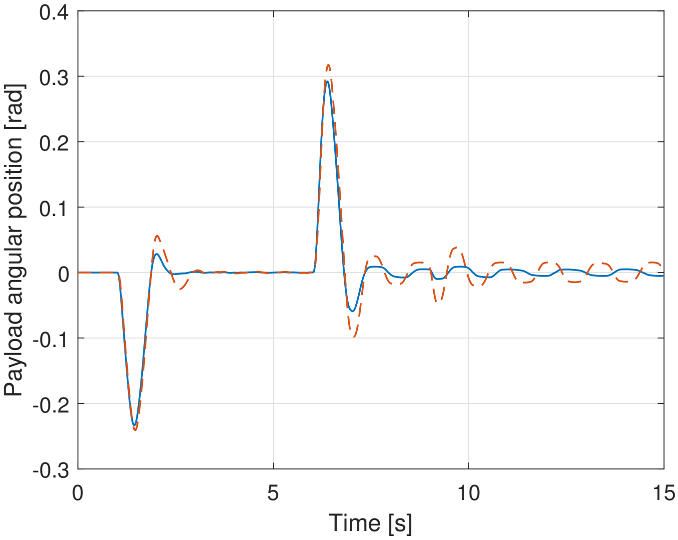


Figure 2.55: Angular position θ: standard MPC control scheme (red dashed line) and proposed MPC-PID approach (blue solid line).

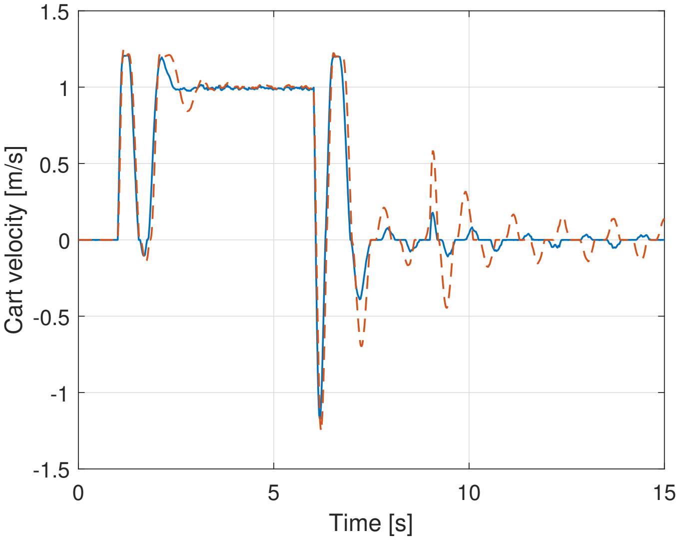


Figure 2.56: Cart velocity $\dot{x}$ for both standard MPC (red dashed line) and MPC-PID control schemes (blue solid line).

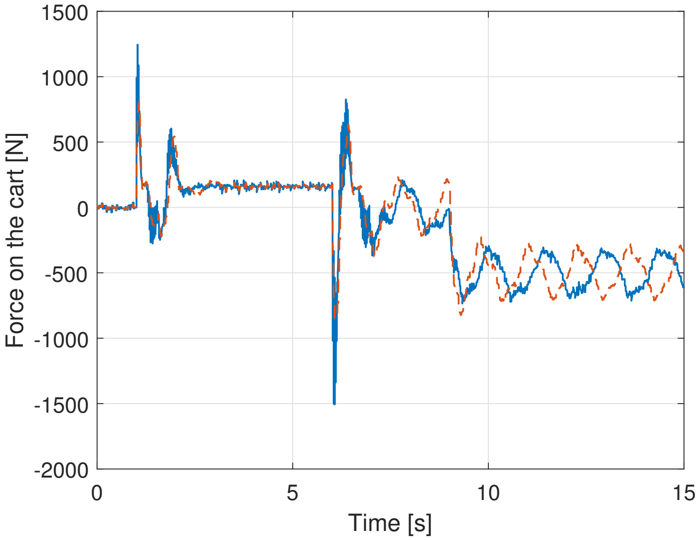


Figure 2.57: Control force acting on the cart. Standard MPC control (dashed red line), MPC-PID scheme (blue solid line).

2.4.2.3 Adaptive and Nonlinear MPC

The closed-loop approaches proposed up until now relies on model of the cranes with constant length of the cable. This implies not only that the cable length is considered as constant along the predictive horizon, but that it is also kept constant along the various control periods.
In industrial applications, overhead cranes usually perform manoeuvres that include simultaneous payload lifting and horizontal displacement, as the material to be handled must move in a cluttered environment, so that the assumption of constant cable length is a substantial limitation to the application of the already presented MPC approaches on the industrial field.
In order to address and solve this limitation by including the variation of the length of the cable in the model, two different approaches have been investigated.

The first solution consists in adopting predictive approaches based on a nonlinear model of the system, that is, by using Nonlinear Model Predictive Control (NMPC). For the case study, a nonlinear model that takes into account the dynamics related to the variation of the cable length is the on in (2.27).

The second solution consists in approaching the problem with an adaptive technique, thus using Adaptive Model Predictive Control (or Gain-scheduling MPC, [15]). This approach relies on the linear model of the system in (2.28), which is used to solve the OCP at each control cycle, but that is updated with the current value of the cable length at every control cycle. In this sense, the system can be seen as a linear parameter-varying system.

Both models describe the dynamics of the system by considering the acceleration of the cart as the input of the system. The velocity of the cart is easily obtained by integration of the input, so that the PID velocity control of the cart can again be exploited, and the control scheme is the same as in Figure 2.51.
With respect to the previously proposed MPC-PID approach, where the

linear dynamic model (2.22) was used, the use of the kinematic model allows for a simplification of the matrices and a reduction of the needed parameters, as the cart and payload masses do not affect model (2.28).
The approach to the definition of the OCP is similar to the methods already proposed. While the velocity reference of the payload can be defined as in (2.85), the lengthening velocity of the cable can be defined similarly as

$$\dot{l}_1 := \begin{cases} \beta & \text{if downward button is pressed,} \\ -\beta & \text{if upward button is pressed,} \\ 0 & \text{if no buttons are pressed,} \end{cases} \tag{2.103}$$

where $\beta \in \mathbb{R}^+$ is the desired absolute value of the lengthening velocity at steady state.

The control of the length of the cable is not considered as being an issue, as it is generally easy to control, and it is thus assumed as perfectly controlled.

The OCP can be defined, for both approaches, as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{p} \big(\boldsymbol{y}(k+i) - \boldsymbol{w}_{\mathbf{1}}(k)\big)^2 + \\ & + \lambda_2 \sum_{i=1}^{p} \big(\boldsymbol{\theta}(k+i) - \boldsymbol{w}_{\mathbf{1}}(k)\big)^2 + \lambda_u \sum_{i=1}^{c} \big(\ddot{\boldsymbol{x}}(k+i-1)\big)^2 \\ \text{subject to} \quad & \dot{\boldsymbol{x}}_{\min} \le \dot{x}(k+i) \le \dot{x}_{\max} \quad \forall i \in \{1, \dots, p\} \\ & \ddot{x}_{\min} \le \ddot{\boldsymbol{x}}(k+i) \le \ddot{x}_{\max} \quad \forall i \in \{1, \dots, p\} \end{aligned} \tag{2.104}$$

In the most relevant case for industrial applications, that is, with operators manoeuvred cranes (OIL cranes), the evolution of $\boldsymbol{w}_{\mathbf{1}}$ and $\dot{l}_1$ along the predictive horizon can not be known in advance, as it strictly depends on the operator real time decisions which, on the other hand, depend on a multitude of factors such, for examples, the obstacles in the cluttered environment. For this reason, assumptions must be made regarding their evolution. As already done in the methods presented in the previous section, the velocity reverence $\boldsymbol{w_1}$ is kept constant and equal to the current velocity reference along the predictive horizon, that is,

$$\boldsymbol{w_1}(k+i) = \boldsymbol{w_1}(k), \quad \forall i \in \{1, \dots, p\}.$$

Different assumptions can instead be made regarding the evolution of the cable length. Depending on the choice made, different MPC approaches can be applied.

ADAPTIVE MPC

One option is to consider the length of the cable as constant along the predictive horizon, while updating its value at every control cycle, that is,

$$\boldsymbol{l}(k+i) = \boldsymbol{l}(k), \quad \forall i \in \{1, \dots, p\}.$$

In this case, the MPC approach can exploit the linear model at every control time, while updating the model parameters at every control time. For this reason, this approach can be referred to as an Adaptive MPC approach.

NMPC with constant cable length

If the length of the cable is kept constant along the along the predictive horizon, in the same way as done for the case of the Adaptive MPC, the nonlinear model (2.27) can be exploited on the optimization process by setting

$$\dot{\boldsymbol{l}}(k+i) = 0, \quad \forall i \in \{1, \dots, p\}.$$

As the optimization process is based on a nonlinear model, this is by all means a NMPC approach.

NMPC with constant cable lengthening velocity

The last option is to consider, at every control cycle, the lengthening velocity of the cable as constant and equal to its current value, that is, by setting

$$\dot{\boldsymbol{l}}(k+i) = \dot{l}(k), \quad \forall i \in \{1, \dots, p\}.$$

In this case, a Nonlinear MPC approach based on model (2.27) can be exploited.

MIL validation

For comparison purpose, the approaches of Adaptive MPC and Nonlinear MPC with both constant and varying cable length along the predictive horizon have been validated using the scheme in Figure 1.4, with the controllers running on Simulink and the nonlinear model of the crane running on Simcenter Amesim, as shown in Figure 2.40. Moreover, as a term of comparison, the proposed approaches are compared with the technique proposed in [24], which does not take into account the variation of the cable length.
The mass of the cart has been set to $m_C = 600$ [kg], the mass of the payload to $m_1 = 100$ [kg], the viscous friction of the cart and of the cable to $C_c = 0.1$ Ns/m and $C_1 = 0.01$ [Nms/rad] respectively. The cable has been characterized with stiffness and viscous friction $K_l = 1 \cdot 10^6$ [N/m] and $C_l = 1000$ [Ns/m] per meter of cable.

The control part has been developed using Simulink, with a cycle time $T_s = 0.02$ [s], chosen to consider the slow dynamics of the system and the presence of already velocity and position control loops for the relatively fast dynamics of the cart. The weights of the MPC controller have been obtained through trial and error and have been set to $\lambda_1 = 1$ and $\lambda_u = 0.2$, and control and prediction horizons $N_c = 10$ and $N_p = 100$ respectively. The Simulink controller and the Simcenter Amesim model run in cosimulation.
The manoeuvre that the crane has to follow can be divided into horizontal and vertical reference. At time t_1 the travelling button is pressed, setting the horizontal velocity reference to $\alpha = 1$ [m/s]. At time t_2, the payload is lifted at a constant velocity $\beta = 1$ [m/s] for 4 seconds. The travelling button is released at time t_3 and the lowering button is pressed at time t_4 and is released after 4 seconds.
Ten different simulations have been analyzed. In order to take into account

Table 2.7: Time instants for the ten simulations analyzed.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
t_1	1.83	2.20	1.50	1.92	4.37	1.77	1.90	1.68	1.91	2.74
t_2	4.21	3.60	3.75	2.24	2.28	1.72	2.53	2.54	3.77	3.67
t_3	12.24	14.69	14.65	11.73	14.61	14.91	12.75	11.44	12.03	12.63
t_4	12.57	11.51	13.62	12.13	11.40	13.75	13.50	12.20	12.48	12.34

the operator unpredictability in the manoeuvre, the aforementioned time instants have been chosen in a random fashion, and their value are shown in Table 2.7.

For the sake of brevity, only Manoeuvre 1 is shown in the following plots, while the quantitative analysis is performed for all the manoeuvres in Table 2.7.

The trajectories for Manoeuvre 1 are shown in Figure 2.58, and the time instants $t_{1,\dots 4}$ are highlighted. In Figure 2.59 the tracking between reference and payload velocity is shown.

First, Figure 2.59 clearly shows that the linear MPC approach is not able to cope with varying cable lengths; when the cable is shortened, oscillations around the reference value are generated due to the difference between the overhead crane and the predictive model.

Second, it has to be noted that, when the length of the cable is considered constant along the prediction horizon and equal to the one measured at each control instant, adaptive and nonlinear MPC provide almost exactly the same performance. In fact, in this case, the only difference between the two approaches is the approximation for small sway angles introduced by the linearization. As the sway angles during industrial cranes manoeuvres are indeed small, the results do not differ significantly.

On the other hand, when the length of the cable is considered to vary during the prediction horizon with the velocity measured at the present control

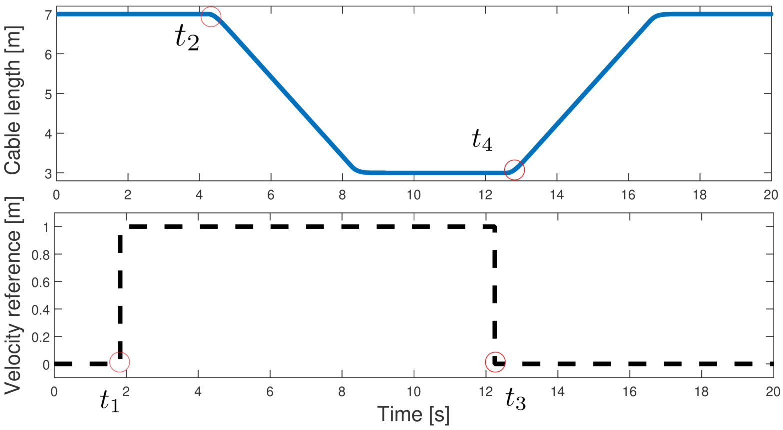


Figure 2.58: Length of the cable and payload horizontal velocity reference over time.

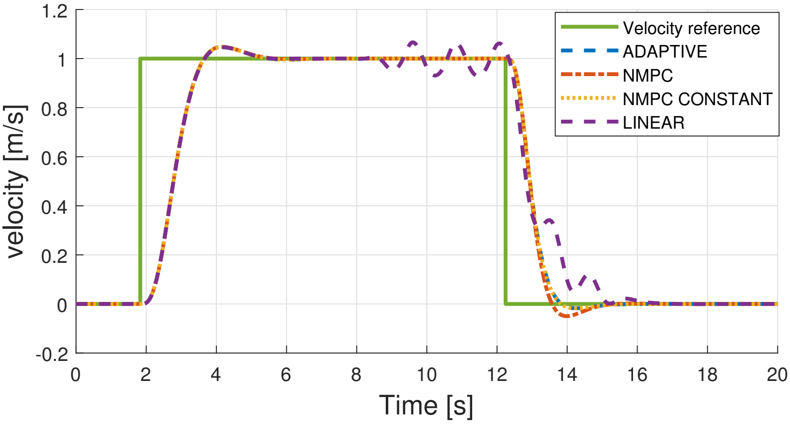


Figure 2.59: velocity reference and payload velocity tracking performance with the simulated MPC approaches.

instant, only NMPC can be exploited. Nonetheless, as the cable length does not vary significantly along the prediction horizon of 1 second ($\Delta l = \pm 1$ m),

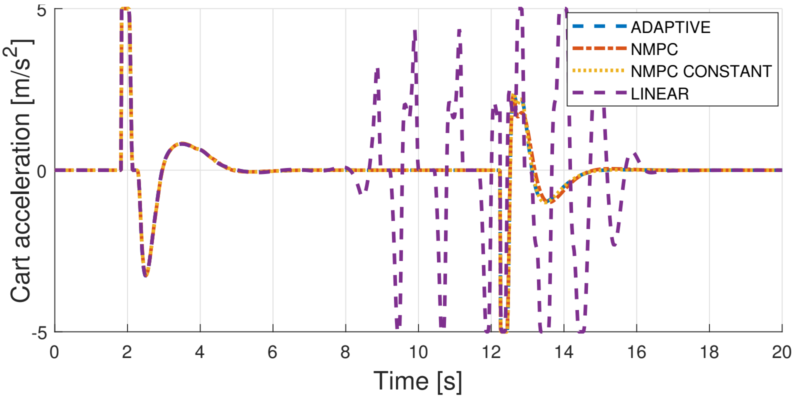


Figure 2.60: Acceleration of the cart resulting with the simulated MPC approaches.

the effect of this variation does not influence the performance of the control in a significant way, so that the NMPC with constant cable velocity along the prediction horizon results to be comparable with the adaptive MPC. On the other hand, the adaptive MPC can rely on a simpler model of the crane, as the nonlinearities introduced by the varying length do not have to be addressed.

In Figure 2.60 the manipulated variable (acceleration of the cart) is shown for the simulated techniques. The constraint of $\ddot{s}_{max} = 5$ [m/s^2] is always satisfied. In order to quantify the performance of the simulated techniques, two indices have been computed: the total Integral Absolute Error (IAE) and the position overshoot after the travelling button is released. Results show that the technique presented in Section 2.4.2.1 yields an unsatisfactory performance when dealing with simultaneous travelling and hoisting manoeuvres, as its IAE always exceeds the ones of the other techniques. Moreover, it is also not effective in containing the position drift of the payload after the operator releases the travelling button.

NMPC effectively reduces the IAE and the position drift with respect to the

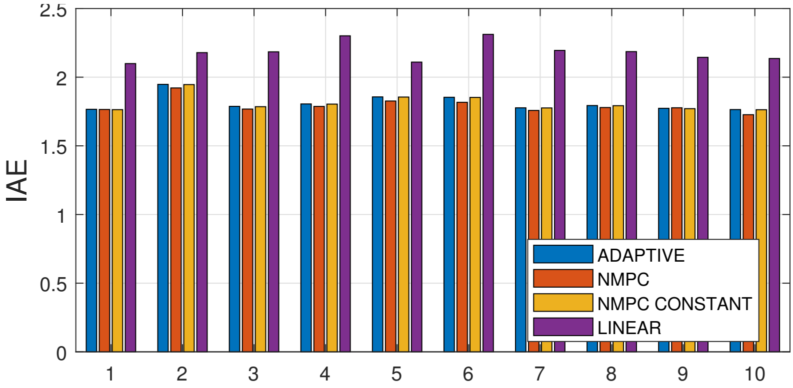


Figure 2.61: Integral Absolute Error during the ten manoeuvres with the compared techniques.

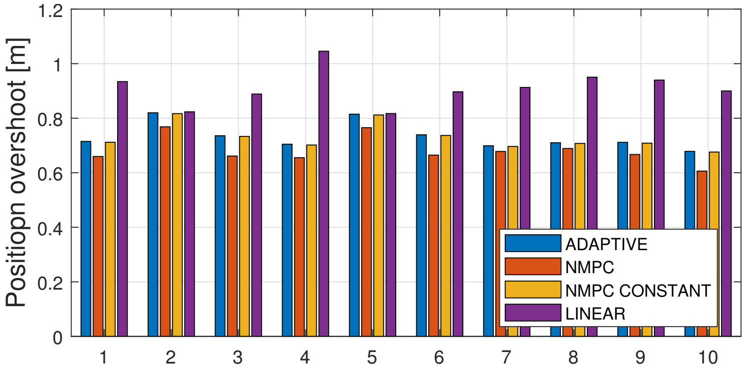


Figure 2.62: Position overshoot after the release of the travelling button for the ten simulations analyzed.

other two techniques which consider the cable length as invariant along the prediction horizon. Nonetheless, the performance of adaptive, NMPC and NPMC with constant cable length is very similar in all the ten simulations.

On the implementation of MPC in industrial hardware

With the aim of validating the presented MPC approaches on the HIL setup of Figure 1.6, the necessary code has been written using IEC 61131-3 Structured Text directly on the hardware. In order to do so, some complications had to be overcome.

Gefran ADV200 industrial drive does not allow neither programming languages external to IEC61131-3 standard, nor the import of external libraries. For this reason, the multiplications between matrices had to be managed by ad-hoc parts of the program. This constitutes a significant computational burden for the CPU. Moreover, although it is possible to obtain the MPC matrices in a closed, parametric form, these matrices can not be updated at every control cycle with the values of the current parameters (nominally, the length of the hoisting cable) as it would require an excessive burden for the CPU. The testable approaches were therefore reduced to the ones relying on linear time-invariant systems. The fastest reachable cycle time for the control on the ADV200 drive was limited to 0.024 [s] due to the high computational burden required by the solution of the OCP.

While testing the approaches on the HIL setup, when comparing the obtained MPC matrices with the ones computed using Matlab, minor numerical errors where present, due to the computational accuracy of the ADV200 CPU, which was limited by single-precision floating-point computations. These numerical errors where propagated along the predictive horizon, and the resulting optimal input vector was distorted by the propagated error. For this reason, while good results were obtained during the MIL validation of the proposed MPC approaches, under the same conditions HIL tests resulted in divergent and unstable results.

This attempt of implementing MPC approaches directly in an industrial drive shows that it is necessary to improve further the computational capability of modern industrial drives if advanced control techniques have to be implemented without the use of external hardware such as PLCs.

On the other hand, the obtained implementation of the MPC approaches have been demonstrated, by means of extensive simulations, to work in practice. As the only obstacle for the application of this approach have been identified as the presence of numeric errors that occurs when using single-precision computation, it can be easily overcome by implementing double-precision computations on the drive. On the contrary, while different fast nonlinear MPC solutions are present in literature (e.g. [35]), their implementation with IEC61131-3 industrial programming languages is not feasible, as they strongly rely, for the online optimization, on complex mathematical libraries which are not compatible with industrial standard solutions. For this reason, the solutions presented in this section represent a step in the direction of a practical industrial implementation of Model Predictive Control for industrial mechatronic systems.

2.5 Conclusions and contributions

In this chapter, different techniques for the open-loop and closed-loop control of cranes have been presented and developed.

All the techniques have been described in details, with a particular focus on the aspects that are relevant in industrial applications, such as robustness, performance in terms of manoeuvre time consumption, implementability in standard industrial hardware and immediacy in the tuning of the required parameters. For all the techniques, MIL validation tests have established the correctness of the methods when applied to a nonlinear simulated model of a generic overhead crane. All the techniques have then been implemented in standard industrial drives, except for MPC techniques, for which the industrial drive has been demonstrated to have inadequate performance for this advanced approach.

Therefore, the contribution of this chapter can be summarized as follows:

- A new methodology for the open-loop control of double-pendulum cranes, based on the input-output inversion approach, have been proposed and tested. The main advantage of this new approach is the possibility to dynamically tune the robustness and, thus, the total delay introduced in the manoeuvre, with a single tuning parameter τ. With respect to input shaping, input-output inversion can vary its robustness in a continuous way, while input shaping presents a set of available filters of fixed robustness. Moreover, it is theoretically possible to decrease at will the introduced delay, which is τ, assuming that the model of the system is perfectly known. With respect to notch filtering, input-output inversion only tuning parameter τ has a clear physical meaning (the amount of delay introduced), while notch filtering tuning parameter $\hat{a}$ could result as being hard to tune by an inexperienced operator.

- A library containing all input shaping techniques as easy-to-use function blocks have been implemented in IEC 61131-3 Structured Text. This solution makes the techniques available over a wide range of standard industrial drives.

- A library for the input-output inversion technique have been implemented in IEC 61131-3 Structured Text. This technique has never been used in industrial applications, and the simplicity of its tuning motivate the need for its implementation with an industrial standard coding language.

- A library containing the notch filter function block has been implemented in IEC 61131-3 Structured Text.

- Full state feedback has been demonstrated to be an effective and easy-to-implement option for the closed-loop control of overhead cranes.

- Different MPC approaches to the control of overhead cranes have been proposed. Nonetheless, the implementation on the ADV200 Gefran drive demonstrated the need for a more powerful computational capability and a more flexible control environment if such advanced techniques have to be implemented in industrial drives.

Chapter 3

Autotuning techniques for mechatronic systems

In this chapter a set of techniques for the automatic tuning of motion control systems is presented.
The techniques presented in this chapter address the problem of automatic tuning of the motion control loops in mechatronic systems by considering different case studies, such as the presence of rigid or elastic shafts and/or transmissions, the availability of pre-existent control loops or their absence.
The chapter is structured as follows: Section 3.1 introduces the chapter by motivating the need of scientific research in the field of new techniques for the automatic commissioning of velocity and position control loops and reviews the available literature on the subject; Section 3.2 formulates the problem and the structure of a general autotuning procedure; Section 3.3 introduces the first autotuning procedure, based on the open-loop identification of generic mechatronic systems, and presents the tuning of the control loops for both velocity and position control; Section 3.4 presents an autotuning procedure for the velocity loop tuning basing the identification phase on a closed-loop test; Section 3.5 presents the built operator interface that allows for an easy use of the presented techniques; Section 3.6 concludes the chapter.

3.1 Introduction

As mentioned in the introduction of this thesis, modern industrial mechatronic systems constantly need for increased performance in terms of both robustness and bandwidth. This increase in performance is not compatible neither with a generic tuning of the control parameters, nor with a very generic control scheme, as this generality yields non-optimal control performance. On the other hand, the main requests for industrial mechatronic systems are the immediate usability of the system with short installation times, together with high flexibility. These requests are hardly compatible with long and time-consuming manual modeling and control tuning tasks.

For this reasons, one of the major difficulties that technicians have to face during the commissioning of mechatronic systems is linked to the calibration of the control parameters for motion control. In fact, each mechatronic system is different from the others and therefore, in order to obtain the required performance, its controller must be correctly adjusted considering its peculiarities. The correct tuning of the control parameters produces an increase in overall production quality and, at the same time, a significant reduction in costs. At the control level, the answer to the demanding requests from the industrial field can therefore be found thanks to the development of techniques for the automatic tuning of control systems.

An autotuning procedure can in fact be used when the machinery is installed, in order to find good control parameters automatically without the intervention of expert operators. In this context, a methodology capable of automatically optimizing the parameters of the controller of a generic mechatronic system can reduce commissioning times. Moreover, automatic procedures for the tuning of control parameters can, in theory, help to obtain the required performance by taking into account the specific behaviour of the system.

Furthermore, the request for flexibility of the systems makes it inconvenient to have to find new control parameters manually every time the system configuration varies (e.g. changes in the load inertia) while, with an automatic calibration procedure, the reconfiguration and subsequent re-calibration of the system does not require the presence of operators with experience and knowledge in the field of control.

The performance of mechatronic systems often degrades over time, and even when the system configuration is not changed, a re-tuning of the control system may still be necessary. Finally, an automatic calibration of the control loops also allows the identification, characterization and compensation of phenomena that compromise the performance of mechatronic systems, such as the presence of static friction and elasticity that entail unwanted resonances.

Most of the control schemes for the control of mechatronic systems are based on PID controllers. The tuning (and autotuning) of PID controllers is a topic of active research: many publications can be found on the autotuning procedures for PID controllers in the field of process control [4]. On the contrary, only a few of them can be found on the tuning of PID controllers for motion control applications so that it is even more essential to increase the focus on the topic [12]. In fact, it is rarely possible to extend autotuning procedures tailored on process control applications to mechatronic systems due to a number of differences between the two fields. In particular, mechatronic systems are characterised by hard constraints on physical values like torque, acceleration and velocity that strongly depends on the application, on the available setup, or both. Furthermore, mechatronic systems are characterized by a series of nonlinearities (e.g. viscous and static friction, backslash) which not only have to be taken into account, but that must be identified and compensated in order to reach the performance that are required by modern applications. Lastly, most of the available tuning procedures and tuning rules for process control are explicitly based on First Order Plus Dead Time (FOPDT) or, sometimes, Second Order Plus Dead Time (SOPDT) approximations of the system. In mechatronics, the system itself lack of the delay, and the delays introduced by the control are, most of the time, negligible with respect to the dynamics of the system, so that those kinds of models (and therefore, tuning rules) are not suitable for mechatronic systems.

For defining a complete autotuning procedure, as will be explained in detail in the next session, three important phases need to be addressed: the excitation phase, the system identification phase and the tuning phase. While the excitation phase and the system identification phase differ from each other, as the former addresses the definition of the input to the system and the latter the processing of the resulting signals, they can be treated as complementary phases. For this reason, various research results for the excitation and identification procedures have been proposed in the last decades, and a small part of them specifically addressed mechatronic systems. Pseudo-Random Binary Sequence (PRBS) is a binary signal which exhibits quasi-random behaviour. Variations of PRBS signals have been applied as input torque to mechatronic systems in [5, 76, 78], as their random behaviour helps in exciting all their frequencies. Alternatively to a PRBS signal, a swept sine signal can be exploited, as it is also able to excite all the frequencies of interest for control purposes. In [30, 32] an identification methodology based on the use of swept sine waves deals also with measurement noise and nonlinearities. The Frequency Response Function (FRF) of the system is then generally obtained by applying the Fast Fourier Transform (FFT) to the input and output signals, or the more accurate Welch method [76].

While the works cited solved the problem of computing, with various approaches, the FRF of the system, the FRF of the system is not a straightforward tool for control purposes, as it cannot be easily expressed with an analytical form. Another approach is to identify the system using parameters identification by defining a parametric model of the system and estimating the value of those parameters using, for example, Least Mean Squares or Nonlinear Optimization approaches. In [36] the so-called Optimal Bandwidth Search Method is used to identify the inertia of the system, and in [79] the autotuning procedure identifies the physical parameters of the system in order to tune the PID controllers for the current, velocity and position control loops. Nonetheless, none of the approaches explicitly addresses the presence of oscillations and vibrations in the system dynamics; moreover, they do not take into account the physical limits of the system during the identification phase, so that they are suitable only for unconstrained systems.

Once the system has been identified, the controller parameters have to be tuned. Most of the proposed approaches relies on well-known PID controllers based cascade schemes for the velocity-position control of the system. In [10] an optimization procedure is used for the tuning of the PID parameters. In [31, 80] the presence of oscillations is explicitly addressed during the tuning of the PID controllers. In [17], several methods for the compensation of oscillations on the systems are compared. Most of them are based on filtering the command signal with low-pass, notch or bi-quadratic filters, while the others rely on the presence of an external observer.

In this chapter, different autotuning procedures specifically tailored for mechatronic systems are proposed. The proposed techniques explicitly take into account the physical limits of the system in terms in maximum torque, maximum velocity and maximum position. The ability of including the respect of the limits during the identification procedure is crucial, as for many applications overstepping the bounds could result in serious consequences and, in some cases, in the complete machine breakdown. Resonances and antiresonances are identified and compensated for by introducing opportunely tuned biquadratic filters inside the control scheme. Moreover, particular attention was paid to the implementability of the proposed approaches in standard industrial systems, and the related issues (reduced memory capability, reduced computation power, etc.) have been addressed.

3.2 Problem formulation

The aim of an autotuning procedure can be expressed as finding the best values for the control scheme while satisfying some common requirements, such as the minimum involvement of the operator (who, ideally, should not be involved at all in the procedure) and a minimum execution time, in order

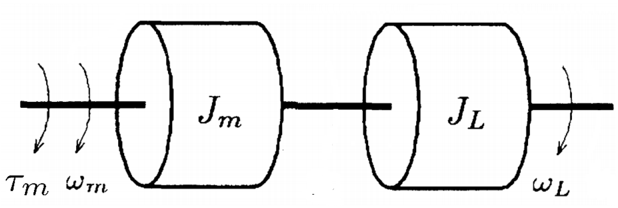


Figure 3.1: Two-inertia system with rigid shaft.

to reduce the overall commissioning costs.

The definition of autotuning procedures for mechatronic systems should start by the definition of suitable model for a generic mechatronic system. A general mechatronic system can be modelled, in a first approximation, as a two-inertia system, as shown in Figure 3.1, where J_m is the inertia of the motor, J_L is the inertia of the load, τ_m the torque on the motor and ω_m and ω_L the motor and load velocity respectively. If the two bodies are connected together by means of a rigid shaft, it results $\omega_m = \omega_L$.

The dynamics of the system in Figure 3.1 can be described by the following system of differential equations:

$$\begin{cases} \ddot{\theta} & = \frac{1}{J_{tot}} \left[\tau_m - K_f \operatorname{sgn}(\dot{\theta}) - B_m \dot{\theta} \right] \\ \ddot{\theta}_L & = i \ddot{\theta} \\ \dot{\tau}_m & = \frac{1}{t_e} (\tau - \tau_m) \end{cases} \tag{3.1}$$

where i is the transmission ratio, $J_{tot} = J_m + J_L/i^2$, τ_m is the real torque acting on the motor, τ is the torque reference, t_e is the time constant of the current loop and B_m is the dynamic friction coefficient.

The system in (3.1) can be expressed, by neglecting the effects of static friction, in the form of a transfer function between the torque on the motor τ_m and the velocity of the load ω_L as

$$G_R(s) = \frac{1}{J_m s + B_m} \frac{1}{t_e s + 1} \tag{3.2}$$

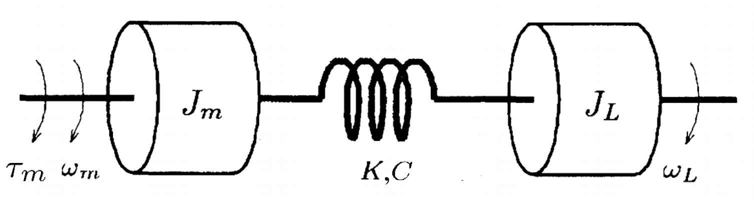


Figure 3.2: Two-inertia system with elastic shaft.

which can be further simplified by the assumption of a high-bandwidth current loop in

$$G_R(s) = \frac{1}{J_m s + B_m}. \tag{3.3}$$

The neglection of the effects of the static friction is motivated by the fact that such effects can be compensated by identifying the static friction coefficient K_f with feedforward actions.

In most applications, the approximation of the shaft as a rigid body is not sufficient, as the dynamics described by (3.1) is not able to describe the oscillatory dynamics that affects the load. For this reason, a more accurate model can be obtained by introducing an elasticity in the shaft, as shown in Figure 3.2, where K is the elastic constant and C is the damping coefficient of the shaft. The dynamics of the system in Figure 3.2 is described by the following system of differential equations:

$$\begin{cases} \ddot{\theta} & = \frac{1}{J_m}\left[\tau_m - K_f \operatorname{sgn}(\dot{\theta}) - B_m\dot{\theta} - \frac{K}{i}d\theta - \frac{C}{i}d\dot{\theta}\right] \\ \ddot{\theta}_L & = \frac{K}{i}d\theta + \frac{C}{i}d\dot{\theta} \\ \dot{\tau}_m & = \frac{1}{t_e}\left(\tau - \tau_m\right) \end{cases} \tag{3.4}$$

where $d\theta = \left(\frac{\theta}{i} - \theta_L\right)$, $d\dot{\theta} = \left(\frac{\dot{\theta}}{i} - \dot{\theta}_L\right)$.
By neglecting the electric pole, which typically has a very fast dynamics, the system in (3.4) can be expressed in the form of a transfer function between

the torque acting on the motor τ_m and the load velocity ω_L as

$$G_E(s) = \frac{Cs + K}{J_L J_m s^3 + (J_L B_m + C J_L + C J_m)s^2 + (C B_m + J_L K + J_m K)s + K B_m}. \tag{3.5}$$

Most of the industrial mechatronic applications lacks for sensor on the load side, as they would increase the overall cost of the system. Instead, the standard approach is to control the motor in closed loop and relying on the rigidity of the shaft in order to have good performance also on the load side. It is therefore useful to divide transfer function (3.5) in two different transfer function as

$$G(s) = M(s)L(s) \tag{3.6}$$

where

$$M(s) = \frac{J_L s^2 + Cs + K}{J_L J_m s^3 + (J_L B_m + C J_L + C J_m)s^2 + (C B_m + J_L * K + J_m K)s + K B_m} \tag{3.7}$$

and

$$L(s) = \frac{Cs + K}{J_L s^2 + Cs + K} \tag{3.8}$$

where $M(s)$ is the transfer function between the torque of the motor τ_m and the velocity of the motor ω_m and $L(s)$ is the transfer function between the velocity of the motor ω_m and the velocity of the load ω_L.

The aim of an autotuning procedure is to find good tuning parameters for a predefined control scheme in order to enhance the overall performance of the system. Autotuning procedures can generally be divided into three different phases, which will be explained in the following sections. Another phase, which is not present in all autotuning procedures but which is included in the proposed algorithms, is the static friction identification. A correct static friction identification allows a compensation of the static friction on both the identification phase and on the subsequent closed-loop control by acting with a feedforward torque signal. The compensation of the static friction means the cancellation of the main source of nonlinearities of the system, which in turns allows for a description of the system in the form of (3.5).

Excitation phase

In this phase, the system is excited by means of various signals and trajectories, that can vary depending on the system on which they are applied. If the system can be excited in open loop, a torque signal has to be designed in order to fully excite the dynamics of the system at every frequency. If the system must be moved in closed loop, trajectories in velocity and or position must be designed in order to move the system between its limits, and a torque feedforward signal can be used to excite high frequency dynamics which would be cut off by the close loop bandwidth.

In this phase, it is important to guarantee the respect of the limits of the system in terms of maximum position, velocity and torque. This is not trivial in particular in the case of open-loop excitation experiments, as the system is generally not known in advance and, in accordance with the autotuning approach, the operator intervention should be as limited as possible.

Model estimation

From the raw data acquired by the excitation phase, a model of the system is needed in order to apply analytic tuning for obtaining the control parameters needed. One of the possible approaches is to define a parametric form of the model (e.g. the transfer function in (3.5)) and to obtain the values of the parameters that better approximate the response in time of the system to the given input signal by means of optimization algorithms (LMS, RLS, genetic algorithms, nonlinear optimization...). Another way of approaching the problem is to convert the obtained signals into the frequency domain and use control tools directly in the Frequency Response Function (FRF)

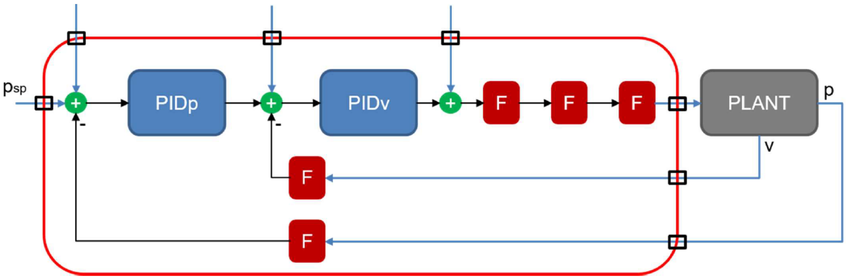


Figure 3.3: Reference I-MECH BB6 control architecture.

of the system or to get a mathematical approximation of the (FRF) (for example, with a transfer function) and than working on that for defining suitable control parameters.

Control parameters tuning

Once the model is obtained, the control system parameters must be tuned. First, it is important to define a suitable control scheme.

The control scheme, shown in Figure 3.3, is a standard cascade control for the position control of mechatronic systems. The possibility of introducing filters on both the input of the system and the feedback lines is foreseen.

Taking a cue from the scheme in Figure 3.3, the control schemes used for the following algorithms are shown in Figure 3.4 for the velocity control and in Figure 3.5 for the position control of the system. Both the controllers of the velocity and position loops are standard PID controllers, while F_1 and F_2 are two biquadratic filters for the setpoint and for the torque input of the system respectively.

Once the control scheme has been decided, if resonances and antiresonances are identified, the filters parameters have to be tuned in order to compensate for them. Finally, the PID controllers have to be tuned.

Two different algorithms will be presented, each one responding to particular needs in terms of technique to be used during the excitation phase.

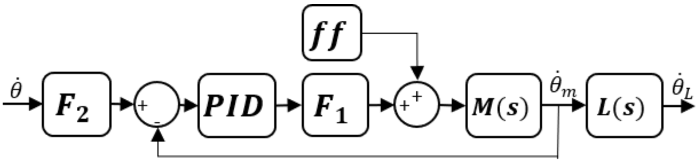


Figure 3.4: Proposed velocity control scheme: biquadratic filters F_1, F_2, PID velocity controller and compensation of the static friction by means of a feedforward torque signal ff.

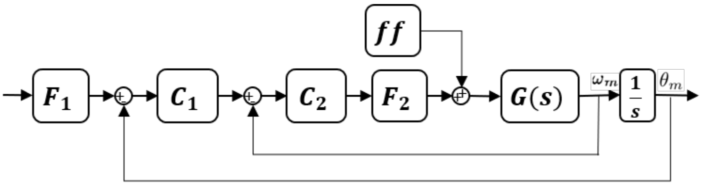


Figure 3.5: Proposed position control scheme: biquadratic filters F_1, F_2, PID position and velocity controllers C_1, C_2 and compensation of the static friction by means of a feedforward torque signal ff.

3.3 Autotuning with open-loop excitation phase

The first proposed algorithm is focused on the open loop-excitation of the system and the subsequent calibration of filters and PID controller for both velocity and position control control.
The proposed algorithm follows the division in phases presented for the general autotuning procedure, but here an additional step is introduced for the static friction coefficient K_f identification, so that the nonlinearity introduced by the static friction can be compensated by acting on the feed-forward signal ff in Figure 3.4. The operator intervention is limited to the definition of the desired bandwidth, as an analytical procedure has been designed for automatically detecting significant peaks in the FRF for the identification of resonances and antiresonances. The transition from FRF to transfer function is executed by using an analytical approach first, based on the identification of key points in the FRF (mechanical pole, resonance and antiresonance peaks), and subsequently on nonlinear optimization algorithms. Filters are tuned not only to increase the performance on the motor side, but also to cancel the oscillations on the load side. The algorithm has been implemented on the HIL setup composed by off-the-shelf industrial hardware described in Chapter 1.4.4, demonstrating its potential for industrial applications.

The flowchart of the algorithm is shown in Figure 3.6.

Static friction identification

First, with the system idle, the noise on the velocity measurement is characterized for a brief amount of time. In particular, the maximum value of the signal is stored, as it corresponds to the maximum level of noise too. Then, a ramp with shallow slope is given as torque input to the signal. The signal is increased at every sampling period T_s by a measure that depends on a tunable parameter N_{steps} and the maximum available torque $\bar{\tau}$, so that

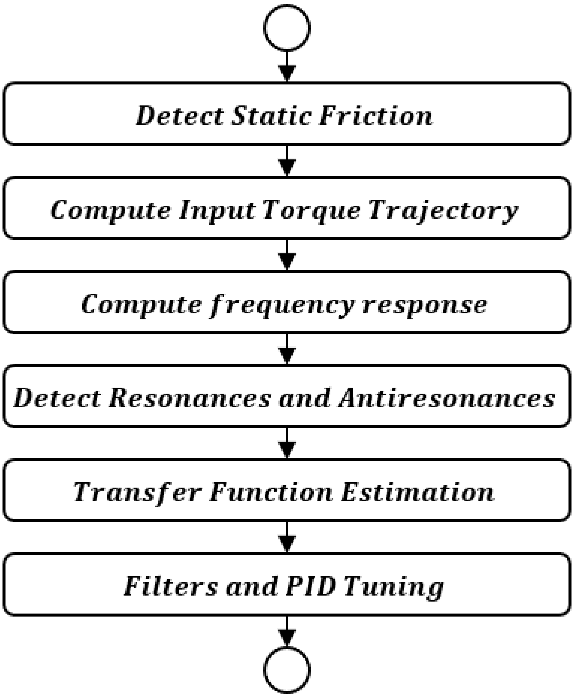


Figure 3.6: Autotuning procedure flowchart.

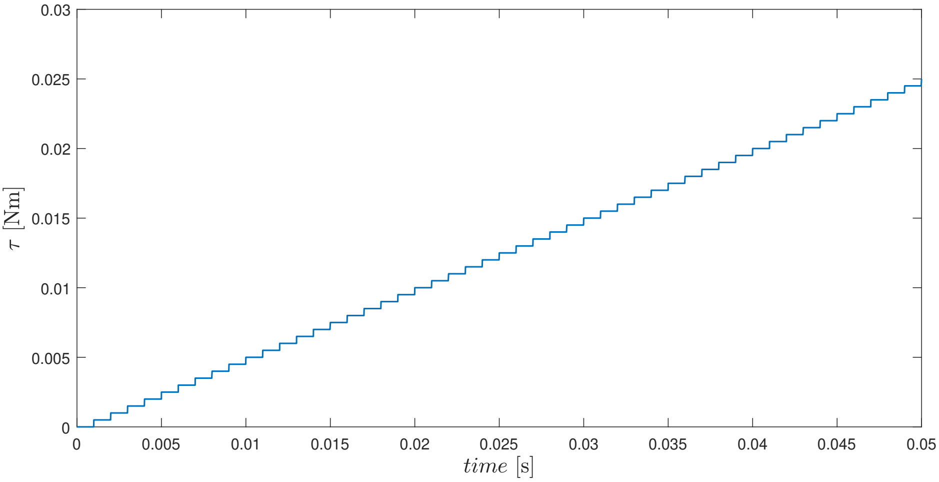


Figure 3.7: Torque ramp signal for the identification of the static friction coefficient K_f.

the increasing on the torque $\Delta\tau$ can be calculated as

$$\Delta\tau = \frac{\bar{\tau}}{N_{steps}}. \tag{3.9}$$

In this way the ramp slope is easily linked to the maximum torque of the system, which is a parameter that the operator must introduce together with the maximum position $\bar{\theta}$, maximum velocity $\bar{\dot{\theta}}$ and maximum acceleration $\bar{\ddot{\theta}}$ of the system.

The torque is continuously increased until the measured velocity reaches a value equal 2 times the measured noise, that is, when the system moves. The corresponding value of torque is saved as estimated static friction coefficient $\hat{K}_f$.

The torque signal used during the static fiction identification phase is shown in Figure 3.7.

Excitation phase

During the excitation phase, the physical limits of the system, introduced by the operator, must be taken into account. The excitation signal generation is therefore generated based on the position, velocity and torque limits $\bar{\theta}$, $\bar{\dot{\theta}}$ e τ_{max}. The torque signal is computed as follows. Cycles are repeated by applying random values of torque steps until the total identification time reaches the value of T_{end}, which can be modified by the operator as it defines the frequency resolution of the FRF as

$$\Delta f = \frac{1}{T_{end}}. \tag{3.10}$$

At the beginning of each cycle, a torque value τ_{new} is chosen in a random fashion in the range that goes from the estimated static friction value $\hat{K}_f$ and the maximum torque $\bar{\tau}$. Every cycle is composed by four phases, that bring the system back to the initial position at the end of the cycle:

- positive torque τ_{new},
- negative maximum torque $-\tau_{max}$,
- negative torque $-\tau_{new}$,
- positive maximum torque τ_{max}.

τ_{new} is applied to the system until condition C_1 is satisfied; then the system decelerates with the negative maximum torque until condition C_2 is reached; subsequently, the system accelerate in the negative direction as a negative torque $-\tau_{new}$ is applied to the system. Finally, when condition C_3 is satisfied, the system is decelerated with the maximum torque $\bar{\tau}$ until condition C_4 is satisfied. The computation of the torque signal is schematized in Figure 3.8.

In particular, condition C_1 is satisfied when the velocity of the motor reaches the maximum velocity $\bar{\dot{\theta}}$ or when the motor position reaches half of the maximum position $\bar{\theta}$; condition C_2 is satisfied when the motor velocity

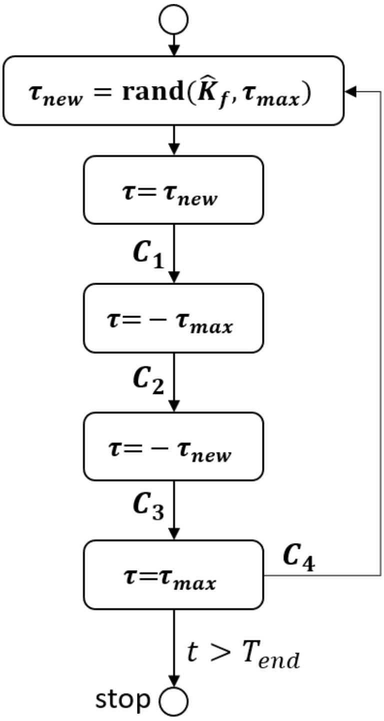


Figure 3.8: Exciting open-loop torque signal generation flowchart.

is negative, while C_3 is the symmetric of condition C_1. Condition C_4 is reached when the motor velocity becomes positive. The conditions can be resumed as follows:

$$\begin{aligned}
C_1 &= \begin{cases} \text{TRUE if } \dot{\theta} \geq \bar{\dot{\theta}} \text{ OR } \theta \geq \dfrac{\bar{\theta} + \theta_0}{2} - \theta_0 \\ \text{FALSE otherwise} \end{cases} \\
C_2 &= \begin{cases} \text{TRUE if } \dot{\theta} \leq 0 \\ \text{FALSE otherwise} \end{cases} \\
C_3 &= \begin{cases} \text{TRUE if } \dot{\theta} \leq -\bar{\dot{\theta}} \text{ OR } \theta \leq -\left(\dfrac{\bar{\theta} + \theta_0}{2} - \theta_0\right) \\ \text{FALSE otherwise} \end{cases} \\
C_4 &= \begin{cases} \text{TRUE if } \dot{\theta} \geq 0 \\ \text{FALSE otherwise} \end{cases}
\end{aligned} \tag{3.11}$$

Model estimation

The measured velocity is stored together with the torque signal at which the static friction is subtracted as

$$\tau_{mod} = \tau_m - \hat{K}_f \operatorname{sign}(\dot{\theta}) \tag{3.12}$$

obtaining the modified torque vector τ_{mod}.

The stored signals are then processed with the Welch method [76] using a Blackman windowing of the signal in order to obtain the FRF. Subsequently, the transfer function between motor torque and motor velocity is estimated using a nonlinear optimization algorithm [48]. The problem of nonlinear optimization is that, if the starting values of the model parameters are not close enough to the real values, the optimization can converge to a local minimum which can result in a wrong model of the system. While the setting the initial parameters could be delegated to the operator, the intervention of the operator during the autotuning procedure should be minimized. For these reason, an analytical approach to the estimation of the transfer function like the one proposed in [32] can give a nice starting point for the following optimization.

The analytical procedure requires three points in the phase and three points in the amplitude of the FRF. In order to acquire some significant points in the FRF, the three points are chosen as the mechanical pole, and the resonance and antiresonance peaks. The resonance and antiresonance peaks are obtained by detecting the most important peak in the phase of the FRF. To do so, the product of the prominence and the width of the peaks has been chosen as the yardstick.

Once the phase peak has been identified, the resonance and antiresonance peaks can be approximately placed at the frequencies corresponding to values at which the phase is equal to the phase peak minus half of its prominence. The position of the mechanical pole is identified by detecting a decrease on the FRF amplitude greater than 3dB with respect to the lowest frequencies amplitudes.

Once the six points have been obtained (3 for the phase, 3 for the amplitude), the analytical method in [32] gives a transfer function in the form

$$G(s) = \frac{K}{s+p}\frac{s^2 + 2\xi_a\omega_a s + \omega_a^2}{s^2 + 2\xi_r\omega_r s + \omega_r^2} \tag{3.13}$$

which can be traced back to (3.5).

Due to the presence of noise in the FRF, the analytical method can return biased parameters, but that nonetheless can be used as starting parameters for the Levenberg–Marquardt nonlinear optimization.

Control parameters tuning

Once the model of the system has been obtained, the oscillatory dynamics caused by the presence of the imaginary poles and zeroes couples in (3.13) can be cancelled thanks to the use of biquadratic filters. Referring to Figure 3.5, the internal filter F_2 can cancel the resonance and antiresonance couple, obtaining the normalized filter

$$F_2(s) = \frac{\omega_a^2}{\omega_r^2}\frac{s^2 + 2\xi_r\omega_r s + \omega_r^2}{s^2 + 2\xi_a\omega_a s + \omega_a^2}. \tag{3.14}$$

Defining

$$\bar{K} := K\frac{\omega_a^2}{\omega_r^2},$$

the transfer function resulting by joining together the filter and the system is

$$G(s)F_2(s) = \frac{\bar{K}}{s+p}. \tag{3.15}$$

The PI velocity controller, in the form

$$C_2(s) = K_{p_2}\left(1 + \frac{1}{T_{i_2}s}\right)$$

is tuned imposing a phase margin Φ_m and a crossover frequency ω_c to the closed-loop system. The tuning of the parameters can be expressed in an analytical form as

$$\begin{aligned} T_{i_2} &= \frac{\tan\left(\phi_m - \frac{\pi}{2} + \arctan\left(\frac{\omega_c}{p}\right)\right)}{\omega_c} \\ K_{p_2} &= \frac{T_{i_2}}{\bar{K}\sqrt{1 + T_{i_2}^2\omega_c^2\left(\omega_c^4 + \omega_c^2 p^2\right)}} \end{aligned} \tag{3.16}$$

Accordingly to the literature, the external loop of a cascade control scheme should be in the order of 10 times slower than the internal loop. For this reason the crossover frequency of the position loop is chosen as

$$\omega_{c_p} = 0.1\omega_{c_v} \tag{3.17}$$

Considering a proportional position controller in the form

$$C_1(s) = K_{p_1},$$

the proportional constant K_{p_1} is given by

$$K_{p_1} = \frac{1}{K_{p_2}\bar{K}}\sqrt{\frac{\omega_{c_p}^4 T_{i_2}^2 (P + K_{p_2}\bar{K})^2 + \omega_{c_p}^2 (K_{p_2}\bar{K} + T_{i_2}\omega_{c_p}^2)^2}{T_{i_2}^2 \omega_{c_p}^2 + 1}}. \tag{3.18}$$

While with the tuning of the internal filter and the velocity and position controllers the requests on the performance of the motor can be considered as met, the load still oscillates due to the effect of transfer function (3.8) which, being introduced by the presence of the elastic shaft, is outside of the closed-loop control. Even if it is not possible to directly measure the transfer function, as no sensor is available on the load side, its form is known as it is known its dependence on the identified resonance and antiresonance in (3.13). In fact, (3.8) has a couple of imaginary poles which corresponds to the couple of imaginary zeros in (3.13), and which cause the oscillatory dynamic on the load side.
In order to compensate for this oscillation, the imaginary poles can be cancelled by adding an external filter F_1 on the position setpoint

$$F_1(s) = \frac{s^2 + 2\xi_a \omega_a s + \omega_a^2}{s^2 + 2\omega_a s + \omega_a^2}. \tag{3.19}$$

Finally, the static friction is compensated by adding the feedforward torque signal calculated as

$$ff = \hat{K}_f \operatorname{sign}(\dot{\theta}_{ref}), \tag{3.20}$$

where $\dot{\theta}_{ref}$ is the velocity reference of the internal velocity loop.

HIL results

In order to validate the effectiveness of the proposed autotuning procedure in industrial application, it has been implemented on the HIL setup described in Chapter 1.4.4. The SM react to the movement generated by the MUT simulating the feedback torque response of the system in Figure 3.2, whose

Parameter	Value	Parameter	Value
C	0.003	K	1.0
J_m	0.0079 kgm	J_L	0.0079 kgm
B_m	0.0027	K_f	0.25
θ_{max}	300 rad	$\dot{\theta}_{max}$	280 rad/s
τ_{max}	5 Nm		

Table 3.1: Parameters values of the simulated system on the HIL setup.

dynamics is described by system (3.4).
The parameters value of the simulated system are shown in Table 3.1. By substituting the values in (3.7), the transfer function of the system can be expressed by decomposing the poles as

$$G(s) = \frac{126.58(s^2 + 0.3797s + 125.9)}{(s + 0.1709)(s^2 + 0.9304s + 253.1)}. \tag{3.21}$$

As the maximum torque of the motor is $\tau_{max} = 5$ [Nm] and the cycle time of the control is $T_s = 0.001$ [s] the tunable parameters for the static friction identification is set to $N_{steps} = 300000$, in order to have an accurate estimation of the static friction coefficient. The result of static friction identification phase, shown in Figure 3.9, is the estimated coefficient $\hat{K}_f = 0.2603$ [Nm].

The excitation signal is executed for a total time $T_{end} = 80$ [s], which gives a frequency resolution of

$$\Delta f = \frac{1}{T_{end}} = 00125 \text{ [Hz]}.$$

The sapling time of $T_s = 0.001$ [s], for the Nyquist theorem, guarantees a maximum identifiable frequency of $f_{max} = 500$ [Hz]. Nonetheless, for the system under analysis, resonances and antiresonances are not expected to

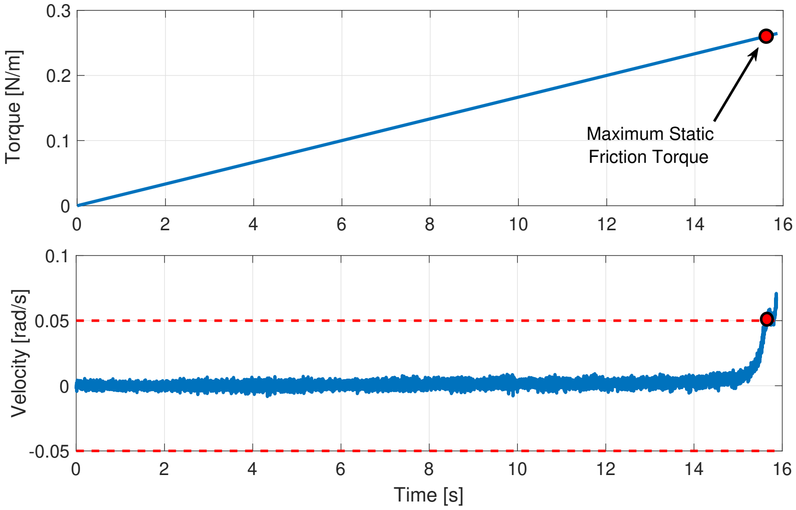


Figure 3.9: Static friction identification phase. The maximum static friction torque is identified as the torque given to the system when the velocity exceeds the limits (dashed red line), which value is calculated as the maximum value of the noise on the velocity multiplied by a safety factor of 5.

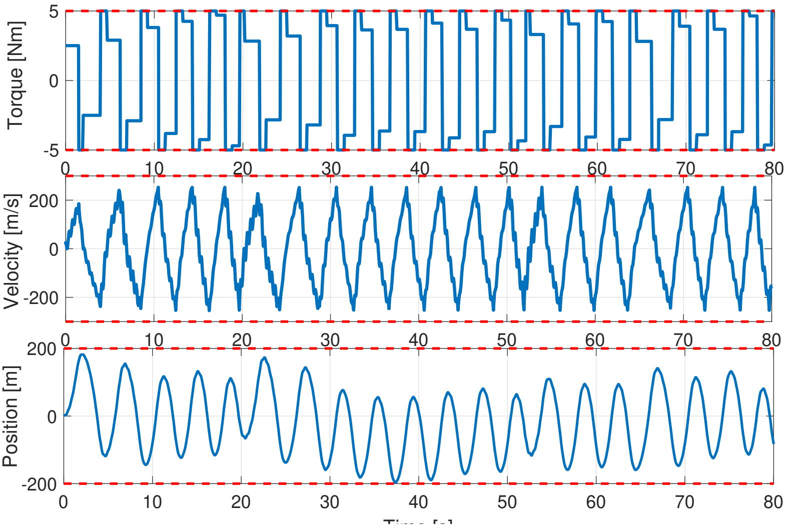


Figure 3.10: Trajectory of the system during the autotuning procedure. The limits of the system are shown in dashed red line.

be at high frequencies, and it is reasonable to expect a digital filter with bandwidth not higher than 100 [Hz] running on a control system cycling at $T_s = 0.001$ [s]. For these reasons, the analysis is limited in the range $[0.0125, 100]$ [Hz].

The trajectory of the system during the excitation phase is shown in Figure 3.10. It can be seen that, while the limits are always respected, the values of position, velocity and torque span over the whole range between the limits. The FRF is then computed by applying the Welch method to the stored signals of torque and velocity, obtaining the results in Figure 3.11. The six points needed for the analytical transfer function approximation procedure of [32] are highlighted with red boxes on Figure 3.11.

The transfer function parameters resulting from the analytical transfer function approximation are then used as starting parameters for the nonlinear Levenberg-Marquardt procedure, which returns the estimated transfer

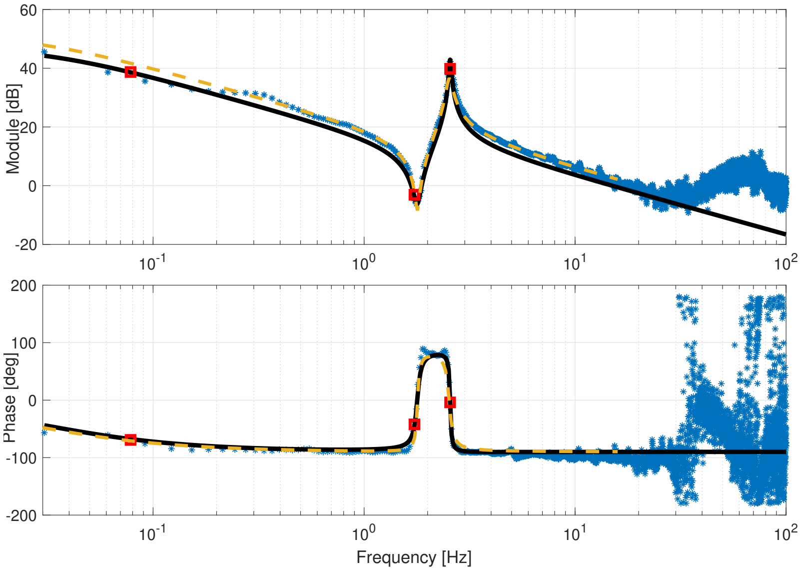


Figure 3.11: Bode diagrams of the system. The theoretical transfer function $G(s)$ (yellow dashed line), the spectral density obtained with the Welch method (blue stars), the three points for the analytical identification (red squares) and the identified system after the optimization (black solid line).

function

$$\hat{G}(s) = \frac{92.724(s^2 + 0.6957s + 125.9)}{(s + 0.1996)(s^2 + 0.3407s + 258.5)}, \tag{3.22}$$

which is also shown in Figure 3.11. The estimated transfer function $\hat{G}(s)$ is a good approximation of the theoretical transfer function $G(s)$.

Once the transfer function of the system has been identified, the crossover frequency ω_{c_v} of the velocity loop and its phase margin Φ_m are set to $\omega_{c_v} = 30$ [rad/s] and $\Phi_m = 85$. The parameters of the PI controller are give by (3.16), and the controller results in

$$C_2(s) = 0.6614\left(1 + \frac{1}{0.3539s}\right) \tag{3.23}$$

while the filters, given (3.14) and (3.19), result in

$$\begin{aligned} F_1(s) &= \frac{s^2 + 0.6957s + 125.88}{s^2 + 22.44s + 125.88} \\ F_2(s) &= \frac{0.487s^2 + 0.1659s + 125.9}{s^2 + 0.6957s + 125.9}. \end{aligned} \tag{3.24}$$

The position proportional controller results from (3.17) and (3.18) in

$$C_1(s) = 3.1646. \tag{3.25}$$

In order to verify the performance of the control loops tuned with the proposed autotuning procedure, the system is tested with a series of steps in the position reference. It has to be noted that, for the position control of mechatronic systems, the use of steps in the position reference is generally avoided, as it generates saturations in the actuators, while other typologies of trajectories are preferred (e.g. 7-traits trajectories). Nonetheless, the use of steps is here motivated as the step response is a standard yardstick for the performance assessment of control systems.

The response of the system with the tuned controllers activated and the filters deactivated is shown in Figure 3.12. While the motor reacts as expected, the elastic transmission introduces a series of oscillations on the

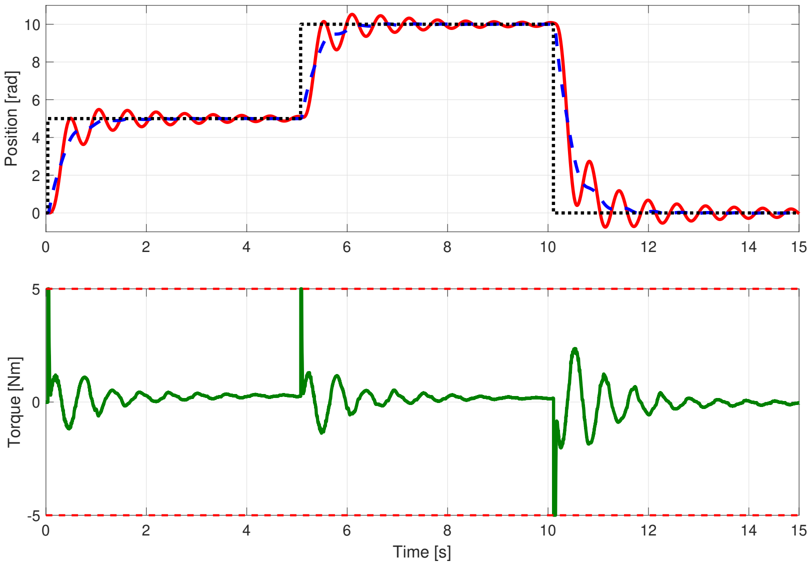


Figure 3.12: Step response of the closed-loop system with the tuned PID controllers. Setpoint signal (black dotted line), motor position (blue dashed line) and load position (red solid line). The corresponding torque is shown in green solid line, along with the torque limits in dashed red line.

load side. The response obtained by activating the tuned filters is shown in Figure 3.13. The filter acting on the setpoint slightly reduce the performance on the motor side in favor of the absence of oscillations on the load side.

The autotuning procedure here described, along with the presented experimental results, has been published firstly in [22] an then extended in this final form in [26]

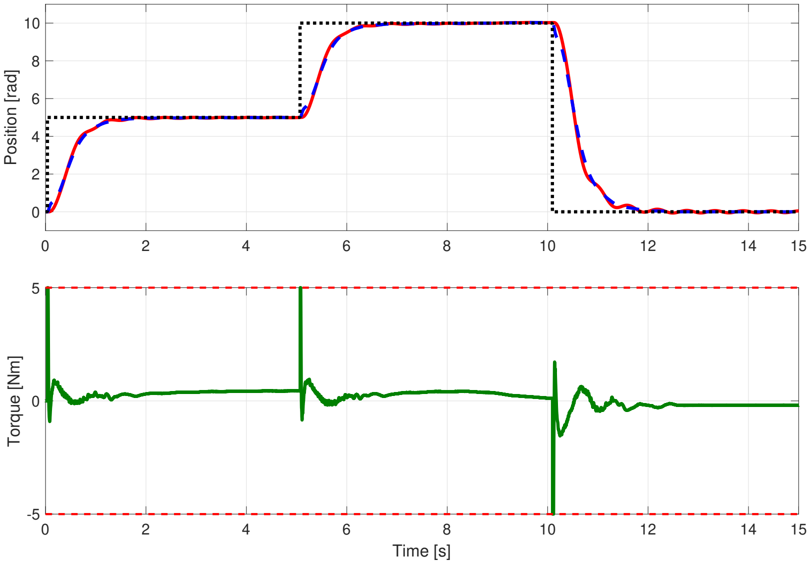


Figure 3.13: Step response of the closed-loop system with the tuned PID controllers and the internal and external filters F_1 and F_2. Setpoint signal (black dotted line), motor position (blue dashed line) and load position (red solid line). The corresponding torque is shown in green solid line, along with the torque limits in dashed red line.

3.4 Autotuning with closed-loop excitation phase

Algorithms 1 and 2 are both characterized by an open-loop excitation phase. While they are able to respect the imposed constraints in terms of position, velocity and torque, as demonstrated by the presented tests in both SIL and HIL, the respect of these constraints is strictly linked with the assumptions that have been made on the system and on the effects of non-modelled nonlinearities.

For example, while the assumption made in the excitation phase of Chapter 3.3 on the inertia of the load is based on good practice [28], the true value of the load can differ significantly. An error in the estimation of system inertia can bring significant deviations in the trajectory during the excitation phase, thus potentially bringing system position and/or velocity over the imposed limits.

In order to increase the robustness during the excitation phase, a closed-loop control of the system would be requested. Nonetheless, the presence of a pre-existent controller cannot be taken for granted, and tuning the parameters of a controller from scratch without a deep knowledge of both system and application is not trivial. Moreover, both trajectory and feedforward signals must be specifically devised, as the feedback loop changes the way in which the FRF must be computed.

This third algorithm deals with the problem of devising an autotuning procedure based on closed-loop control of the system in every phase, to ensure the respect of imposed limits also in the case of errors in the model parameters and unmodelled nonlinearities.

Also in this case, the resonances introduced by the elastic shaft are detected and suppressed by means of appropriately tuned biquadratic filters.

The proposed method is shown in Figure 3.14.

Generally, the velocity control of a mechatronic system can be schematized as in Figure 3.15. The loop is closed on the motor side, as sensors on the load side are rarely available. For this reason, the system can effectively be divided into two different parts: the closed-loop part, represented by

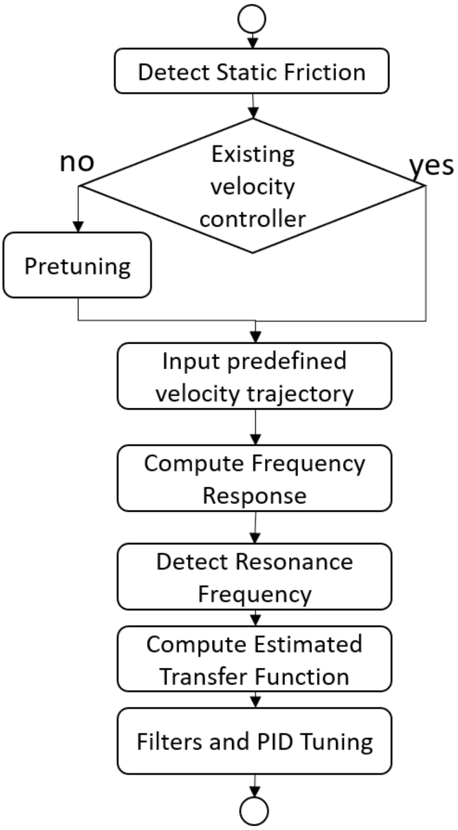


Figure 3.14: Flowchart of the closed-loop autotuning procedure.

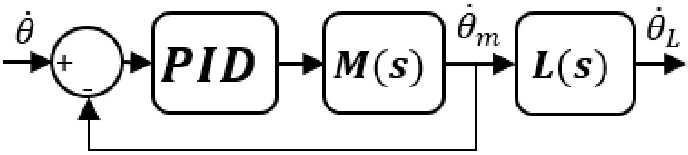


Figure 3.15: Closed-loop velocity control scheme for a mechatronic system, with feedback on the motor side.

$M(s)$ in 3.7, which is the transfer function between the torque acting on the motor and its velocity, and an open-loop part, $L(s)$ in (3.8), which links the velocity of the motor to the velocity of the load.

The proposed velocity control scheme is the one shown in Figure 3.4 which, with respect to the standard control scheme in Figure 3.15, adds two biquadratic filters for the compensation of the oscillations generated by the elastic shaft, and a feedforward signal, used both during the excitation phase of the autotuning procedure and the static friction compensation once the newly tuned control is in position.

Here, a controller is assumed to be already in place, and that it needs to be retuned. The retuning of a controller can be required in a number of different occasions (e.g. change of the load inertia, change in the static friction value). It could also happen that no controller is already in place, for example when commissioning the system; for this case, a method for tuning a basic controller and adopting this closed-loop autotuning procedure is presented at the end of the section.

Static friction identification

In order to identify the static friction value, a procedure similar to the one proposed in Section 3.3 is applied. In this case, as it is unpractical and sometimes not possible to act directly on the system opening the control loop, a ramp in the velocity reference is built. The ramp, starting from $\dot{\theta}_{ref} = 0$, must increase slowly. As the static friction impedes the movement,

the system will not move instantly. The static friction K_f is identified as the value of the input torque when a movement of the system is detected.
In order to link the slope of the system with a physical value, the parameter N_{steps} is introduced, so that

$$\Delta\dot{\theta}_{ref} = \frac{\dot{\theta}_{max}}{N_{steps}}, \tag{3.26}$$

where $\Delta\dot{\theta}_{ref}$ is the increase in the velocity reference at each cycle time T_s.

Excitation phase

A velocity reference signal is built taking into account the limits of the system, which depends on the application and are set by the operator. The limits are the maximum position $\bar{\theta}$, the maximum velocity $\bar{\dot{\theta}}$ and the maximum motor torque $\bar{\tau}$.
A random value $\dot{\theta}_R$ is chosen in the range $[0.2\bar{\dot{\theta}};\ \bar{\dot{\theta}}]$. A velocity step, with amplitude equal to $\dot{\theta}_R$, is set as reference for the velocity loop and is kept constant for a period of time T_p. T_p is calculated so that the position limit is respected, thus $T_p = \dfrac{\bar{\theta}}{\dot{\theta}_R}$. Then, the system is brought back to its initial position by a step in the velocity reference with amplitude $-\dot{\theta}_R$ for the same amount of time T_p. In order to avoid the undesired effects introduced by a step in the velocity reference, the velocity reference is filtered by imposing a limit to its slope to a value equal to

$$\Delta\dot{\theta}_{ref_{max}} = \frac{\tau_{max}}{J_m} \tag{3.27}$$

where J_m is the inertia of the motor, which is typically printed on the motor plate.
It has to be stressed that, by not considering the load inertia J_L, the total inertia of the system is probably greater that the motor inertia J_m so that, even by limiting the velocity slope with (3.27), the actuators will probably saturate. Nonetheless, with no information on the load, limiting the slope as

done in (3.27) helps to at least reducing the effects of protracted saturations (e.g. integral windup in PID controllers).

This procedure is iterated, modifying the value of $\bar{\dot{\theta}}$ in a random fashion, for a total time of T_{tot} which is chosen based on the required spectral resolution, as the spectral resolution is given by

$$\Delta f = \frac{1}{T_{tot}}. \tag{3.28}$$

A total excitation time of $T_{tot} = 80$ [s] is sufficient for most mechatronic applications, giving a spectral resolution (and a minimum frequency for the FRF analysis) of 0.0125 [Hz].

In order to enrich the frequency components on the input torque signal, a feedforward *chirp* signal (or *sweep* signal) is given to the system along with the steps in the velocity reference. The chirp signal is build in order to be linear in the logarithmic scale, and it can be expressed as a function of the total time T_{tot} and the initial and final frequency f_0 and f_1 as

$$ff(t) = A \sin\left(2\pi f_0 \frac{k^t}{\log(k)}\right) \tag{3.29}$$

with

$$k = \left(\frac{f_1}{f_0}\right)^{\frac{1}{T_{tot}}}. \tag{3.30}$$

f_0 and f_1 are chosen based on the total time T_{tot} and the sampling period T_s; given the frequency resolution in (3.28) and the Nyquist theorem, the frequencies are given by

$$\begin{cases} f_0 = \dfrac{1}{T_{tot}} \\ \\ f_1 = \dfrac{0.5}{T_s} \end{cases} \tag{3.31}$$

The amplitude A of the feedforward sweep signal is set to 20% of the maximum torque $\bar{\tau}$.

Model estimation

For the computation of the Frequency Response Function, due to the presence of the feedback loop, it is not possible to use the same approach as in Section 3.3. For this reason, the Ljung method is here used for the FRF computation [18]. Starting from the velocity reference signal, the torque feedforward, the motor torque and the motor velocity, computes a high order transfer function of the system. The so found high-order transfer function is hardly manageable for control purposes, and must therefore be reduced to a simplified transfer function in the form 3.7.
This step is made possible by the use of the analytical procedure used in Section 3.3, that is, by finding the mechanical pole as the frequency decreases more that 3dB under its lowest frequency value, and by detecting the resonance and antiresonance peaks by comparing the product of the prominence and the width of the peaks in the phase. The difference with respect to the approach in Section 3.3 is that the frequency data are now computed by sampling amplitude and phase of the high order transfer function obtained with [18], and not directly from the computed FRF data.
Once the six points have been identified (two points for every identified frequency, one in the phase and one in the amplitude), the method in [32] returns a set of six parameters identifying a transfer function in the form

$$M(s) = \frac{a_1}{s + a_2} \frac{s^2 + a_3 s + a_4}{s^2 + a_5 s + a_6}. \tag{3.32}$$

This set of parameters can then be used as a starting set for a nonlinear Levenberg–Marquardt optimization [48].

Control parameters tuning

Once the model of the system has been obtained in the form (3.32), the oscillatory dynamics must be cancelled in order to enhance the control performance. To do so, the biquadratic filters of the scheme in Figure 3.4 are tuned. The internal biquadratic filter cancel the resonance/antiresonance couple in (3.32), obtaining the normalized filter

$$F_1(s) = \frac{\frac{1}{a_6}s^2 + \frac{a_5 s}{a_6}s + 1}{\frac{1}{a_4}s^2 + \frac{a_3}{a_4}s + 1}. \tag{3.33}$$

The filtered system is therefore given by

$$M(s)F_1(s) = \frac{a_1 a_4}{a_6(s + a_2)} \tag{3.34}$$

A PI controller, with analytical form

$$C(s) = K_p\left(1 + \frac{1}{T_i s}\right) \tag{3.35}$$

is tuned imposing a phase margin Φ_m and a crossover frequency ω_c. These parameters are set by the operator depending on the application performance requirements.
The controller results from the parametric computation of its parameters, which can be expressed as

$$\begin{aligned} T_i &= \frac{\tan\left(\phi_m - \frac{\pi}{2} + \arctan\left(\frac{\omega_c}{a_2}\right)\right)}{\omega_c} \\ K_p &= \frac{T_{i_2}}{\frac{a_1 a_4}{a_6}\sqrt{1 + T_{i_2}^2\omega_c^2\left(\omega_c^4 + \omega_c^2 a_2^2\right)}}. \end{aligned} \tag{3.36}$$

While closed-loop control alone can entail goo performance on the motor side, the open-loop transfer function $L(s)$ in Figure 3.4, which corresponds to the transfer function that links the motor velocity with the load velocity, present two complex poled. This poles, which are due to the presence of the elastic transmission, are responsible for important oscillations in the load dynamics, even when the motor is under control, as $L(s)$ acts outside of the closed-loop.
In order to cancel the oscillatory dynamics in $L(s)$, the external filter is

tuned so that its zeros cancel the imaginary poles of $L(s)$. While it is true that $L(s)$ cannot be estimated directly, due to the lack of sensors on the load side, it can be obtained by (3.7), (3.8) and (3.32).
As the poles of $L(s)$ are indeed equal to the zeros in $M(s)$, the zeros of the filter are set to

$$\mathrm{num}_{F_2}(s) = s^2 + a_3 s + a_4; \tag{3.37}$$

the poles p_1, p_2 of the filter are set at a frequency equal to the one of the zeros, but real in order not to introduce other oscillatory dynamics, so that

$$p_1 = p_2 = -\frac{1}{2}\sqrt{a_3^2 + \left|a_3^2 - 4a_4\right|}. \tag{3.38}$$

The normalized filter results from (3.37) and (3.38) in

$$F_2(s) = \frac{\frac{1}{a_4}s^2 + \frac{a_3 s}{a_4} + 1}{\frac{1}{p_1^2}s^2 + \frac{2}{p_1}s + 1}. \tag{3.39}$$

The feedforward signal in Figure 3.4 is used to compensate for the static friction, and is set as

$$ff = \hat{K}_f \, \mathrm{sign}(\dot{\theta}_{ref}). \tag{3.40}$$

Controller pre-tuning

In the case of the commissioning of a new mechatronic system, the presence of a controller is not obvious. Nonetheless, in some cases a closed-loop identification is a requisite, as the system could not be moved in open loop (e.g. for robustness problems, for the acceleration gravity contributes etc. . .).
In this case, a first controller to be used during the excitation of the system can be tuned as follows. Given the motor inertia J_m, which is typically printed directly on the motor plate, and the viscous friction B_m, a proportional controller is tuned, considering the system as described by the simplified transfer function (3.3).
As the static friction can be supposed to have already been identified (with the method proposed in section 3.3), the controller parameter K_p of the

proportional controller $C(s) = K_p$ can be set so that a value in the velocity reference equal to the maximum velocity $\bar{\dot{\theta}}$ would give a motor torque

$$\tau_m = 2\hat{K}_f. \tag{3.41}$$

While this probably results in a very conservative control, by setting the controller gain with (3.41) the movement of the system is guaranteed, and there is no risk for the torque to get stuck under the static friction value.

Subsequently, a step with value equal to

$$\dot{\theta}_{ref} = \bar{\dot{\theta}}$$

is given as reference to the system, for a period of time calculated based on position limits, thus

$$T_p = \frac{\bar{\theta}}{\bar{\dot{\theta}}}.$$

The last values of motor velocity $\dot{\theta}_m$ and motor torque τ_m before time T_p are used for estimating the value of B_m. It is supposed that, by that time, the motor velocity has reached its steady state value.
he value of B_m can therefore be calculated as

$$B_m = \frac{\tau_m - \hat{K}_f}{\dot{\theta}_m}. \tag{3.42}$$

Then, knowing that the steady state error of the loop composed by a proportional controller and the first order system (3.3) is given by

$$e_{ss} = 1 - \frac{K_p}{K_p + B_m}, \tag{3.43}$$

the proportional gain of the controller can now be retuned based on the desired steady state error during the excitation phase.
As a sluggish controller is sufficient during the autotuning, an error equal to 0.1 times the reference is acceptable. Given that, the proportional gain of the controller can be set equal to

$$K_p = 9B_m = 9\frac{\tau_m - \hat{K}_f}{\dot{\theta}_m}. \tag{3.44}$$

Table 3.2: Limits and parameters of the simulated system for HIL validation.

Parameter	Value	Parameter	Value
C	0.003 Nms/rad	K	1.0 Nm/rad
J_m	0.0079 kgm	J_L	0.0079 kgm
B_m	0.005 Nms/rad	K_f	0.3 Nm
θ_{max}	400 rad	$\dot{\theta}_{max}$	150 rad/s
τ_{max}	5 Nm		

HIL results

The autotuning algorithm has been validated directly on the HIL setup described in Chapter 1.4.4 and shown in Figure 1.6. Limits and parameters of the simulated system are shown in Table 3.2. With the parameters shown in Table 3.2, the transfer function of the motor, calculated as (3.7) and reduced into pole-zero representation becomes

$$M(s) = \frac{129.7(s^2 + 0.9193s + 124.2)}{(s + 0.3719)(s^2 + 1.104s + 257.5)}. \tag{3.45}$$

With an initial PI controller defined by the control parameters $K_p = 0.05$ and $T_i = 2.0$, the performance of the system are shown in Figure 3.16. This performance is unsatisfactory, both due to the sluggish step response and the oscillations affecting motor and load, thus a retuning is required.

The static friction is identified, and the estimated value is $\hat{K}_f = 0.28$ [Nm]. The total execution time for the identification trajectory is set to $T_{tot} = 80$ [s], giving a frequency resolution of

$$\Delta f = f_{min} = \frac{1}{T_{tot}} = 0.0125 \text{ [Hz]}. \tag{3.46}$$

With a sampling period $T_s = 0.001$ [s], the Nyquist theorem sets a maximum identifiable frequency of $f_{max} = 500$ [Hz]. Nonetheless, the frequency

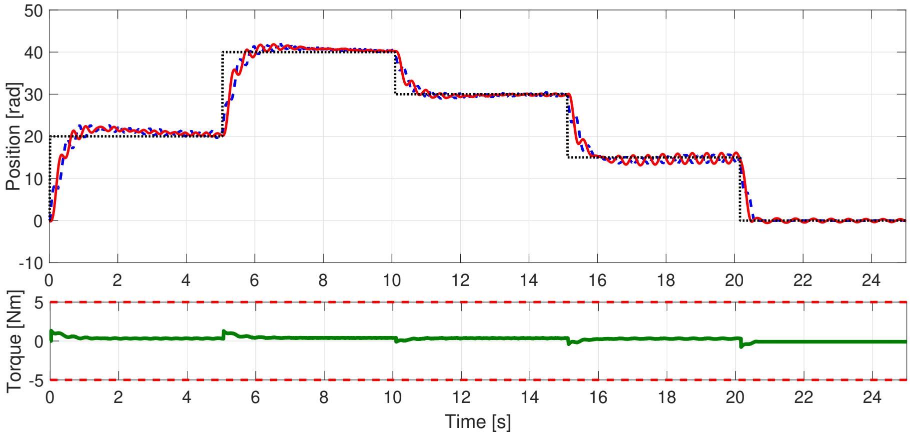


Figure 3.16: Performance of the system with the initial PI controller. Velocity reference (black dotted line), motor velocity (blue dashed line) and load velocity (red solid line). On the graphic below, the corresponding torque acting on the motor (green solid line), with the torque limits shown in dashed red lines.

analysis has been limited 100 [Hz], as no interesting dynamics are expected at higher values of frequency. The excitation trajectory is executed, and the data of motor velocity, motor torque, feedforward torque and velocity reference are stored and analyzed. It must be noted that, during the identification trajectory shown in Figure 3.17, the imposed limits have always been respected.

The collected data are then analyzed through the Ljung method, which results in a high order transfer function. The mechanical pole is identified, together with the resonance and antiresonance peaks, as shown in Figure 3.18.

The analytical method and the subsequent nonlinear optimization result in the low order estimated transfer function

$$\hat{M}(s) = \frac{129.7(s^2 + 0.9193s + 124.2)}{(s + 0.3719)(s^2 + 1.104s + 257.5)}. \tag{3.47}$$

In Figure 3.18 the theoretical transfer function of the system is compared with the high order transfer function of the system resulting from the Ljung method and the low order transfer function after the analytical method and the nonlinear optimization. It can be seen that the identified transfer function reflects with good approximation the real transfer function.

For the tuning of the PI velocity controller, the phase margin Φ_m and the crossover frequency ω_c are set to $\Phi_m = 80$ and $\omega_c = 20$ [rad/s], obtaining

$$C(s) = 0.3196\left(1 + \frac{1}{2.6892s}\right). \tag{3.48}$$

In Figure 3.19 the performance of the system is shown for the case with the tuned controller, but without the introduction of biquadratic filters. It can be seen that, while the performance on the motor side is greatly increased, significant oscillations affect the load side. For this reason, it is necessary to introduce the biquadratic filters, which are tuned as

$$F_1(s) = \frac{0.003884s^2 + 0.004288s + 1}{0.008049s^2 + 0.0074s + 1} \tag{3.49}$$

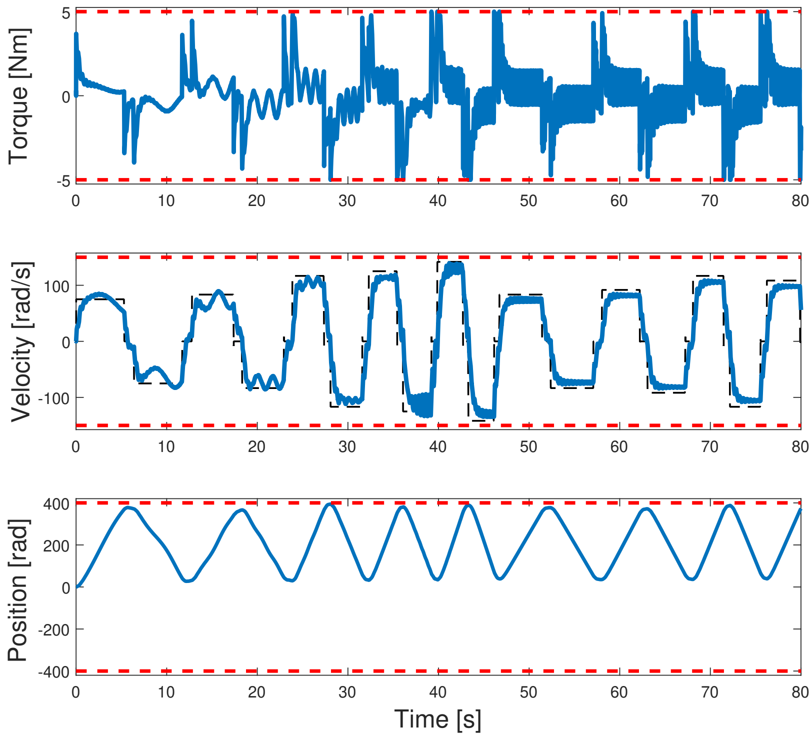


Figure 3.17: Motor torque, velocity and position during the identification phase. The velocity reference is the black dashed line, while the limits are shown as red dashed lines.

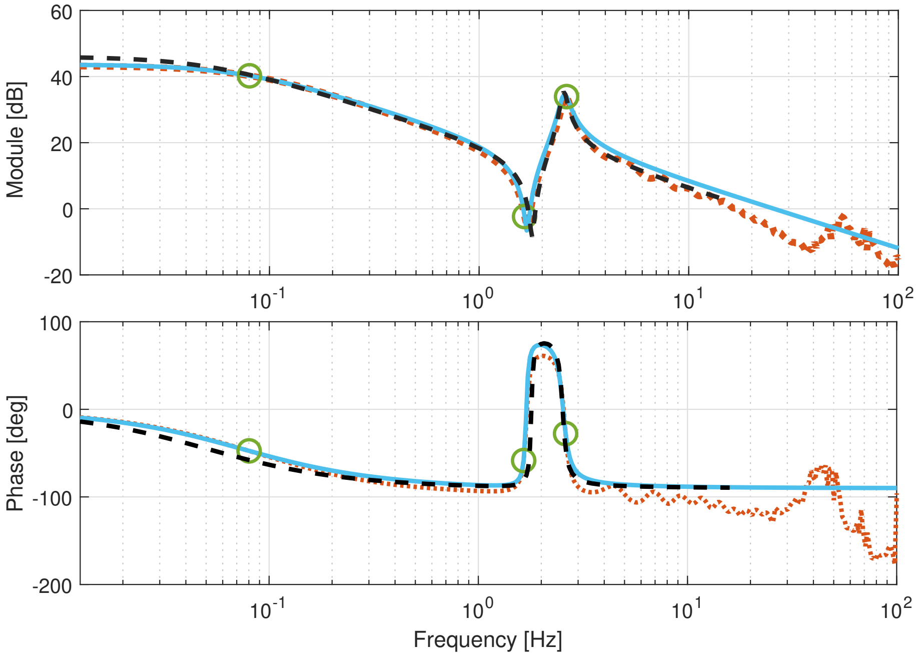


Figure 3.18: Bode plot of the transfer function (3.21) (dashed black line), of the high-order transfer function obtained using the Ljung method (orange dotted line) and of the reduced-order transfer function (blue solid line). The three points necessary for the analytic order reduction are shown with green circles.

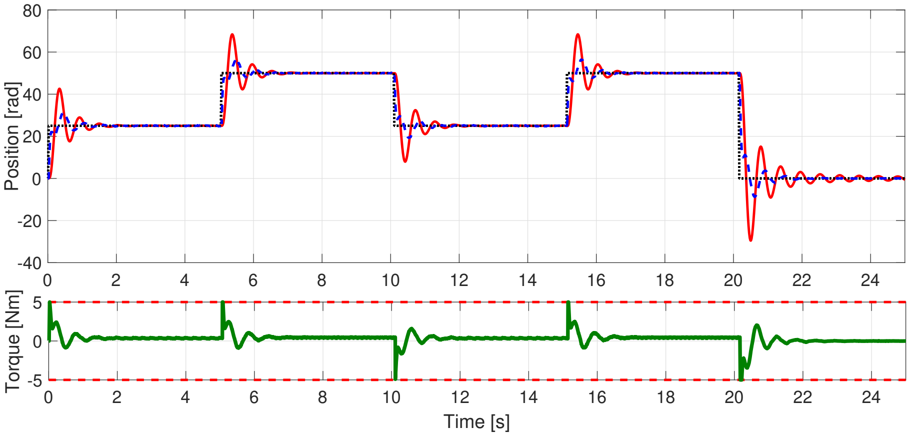


Figure 3.19: Performance of the system with the obtained PI velocity controller, without the biquadratic filters. Load velocity (red solid line), motor velocity (blue dashed line), velocity reference (black dotted line) and torque (green solid line). The torque limits are shown in red dashed line.

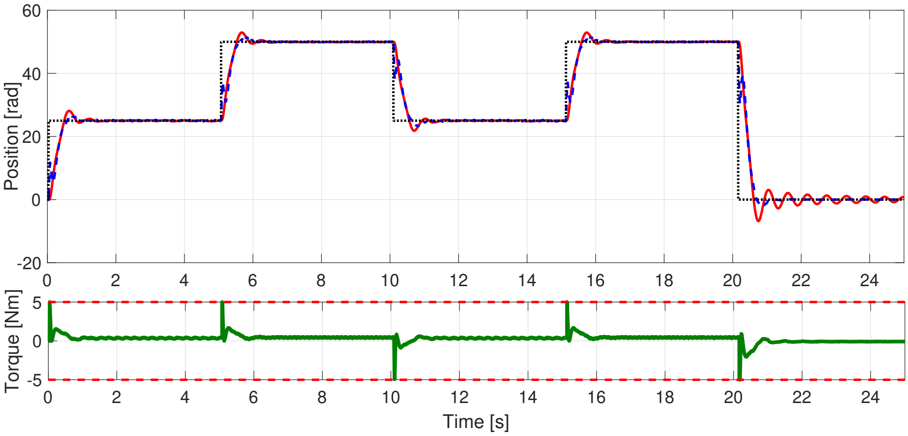


Figure 3.20: The performance of the control system with tuned PI controller, biquadratic filters and feedforward compensation of the static friction. Velocity reference (dotted black line), motor velocity (dashed blue line), load velocity (solid red line) and motor torque (solid green line). Limits on the torque are shown with dashed red lines.

and

$$F_2(s) = \frac{0.008049s^2 + 0.0074s + 1}{0.008049s^2 + 0.1794s + 1}. \tag{3.50}$$

The performance of the control system, with tuned PI controller, biquadratic filters and feedforward compensation of the static friction is shown in Figure 3.20.

With respect to the detuned system in Figure 3.16, both motor and load have increased their step response performance. With respect to the use of only the tuned PI controller, the performance in Figure 3.20 are increased. In fact, while the filters reduce the promptness of the step response on the motor side, the load follows the motor without oscillating.

The procedure of this algorithm and the obtained results are published in [27].

3.5 Operator interface

The different algorithms developed have been implemented and included on a Matlab/Simulink library. This allows for a consistent interoperability between different industrial systems, as Matlab environment provides for interfaces towards a number of different industrial communication protocols. Moreover, the entire library can be exported thanks to the Matlab auto-coding toolbox.

In order to ease the usability of the autotuning algorithms for industrial operators, a Graphic User Interface (GUI) has been developed, which collects all the operator inputs required by the autotuning procedure and allows for the expert user intervention during the tuning phase.

The information that are available on the GUI are the following:

- maximum and minimum torque during the autotuning procedure
- maximum and minimum velocity during the autotuning procedure
- maximum and minimum position during the autotuning procedure
- maximum and minimum torque during the normal application execution
- maximum and minimum velocity during the normal application execution
- sensors position (load side / motor side)
- velocity closed-loop enabling
- position closed-loop enabling
- sampling time
- phase margin
- control bandwidth

- encoder resolution
- ...

Figures 3.21-3.24 show in details the autotuning user interface built with Matlab Appdesigner. The interface directly connects with the drive by means of modbus communication. The use of Matlab Appdesigner for the user interface does not limit its applicability in general industrial systems, as it is possible to export the whole application together with Matlab runtime to an executable file. The interface use is straightforward. The operator must select the limits in terms of torque, velocity and position, the type of control (velocity, position with or without cascade control) and, if needed, the tuning requirements such as phase margin and bandwidth, and then launch the automatic procedure. The interface interactively shows the status of the autotuning procedure, as it constantly communicates with the drives by means of modbus communication. The identification trajectory is shown, and the FRF of the system is shown as well together with the identified transfer function of the system. An expert operator can choose to include a filter in the control scheme, in the case where relevant resonance peaks are identified. The tuning parameters are computed, and the controller can be validated as the found tuning parameters can then be uploaded directly to Gefran ADV200 drives.

This approach is viable in industrial field. In fact, the presence of a simple user interface, together with valid autotuning procedures, can speed up the commissioning process of mechatronic systems. The operator can tune the controllers directly on-site with his own PC, as the communication between the drives happens directly via a standard industrial communication protocol such as modbus.

DriveName ADV200SynAUTOTUNING
Baudrate 38400 115200
COM 0
☐ Verbose

Max Torque [Nm] 5
Max Speed [rpm] 500
Max Position [Turns] 10
Min Position [Turns] -10

Min Freq [Hz] 0
Max Freq [Hz] 0

Duration [s] 0
FF Type None
FF Amp [Nm] 0
ChirpFType Linear
ChirpFStart [Hz] 0
ChirpFStop [Hz] 0

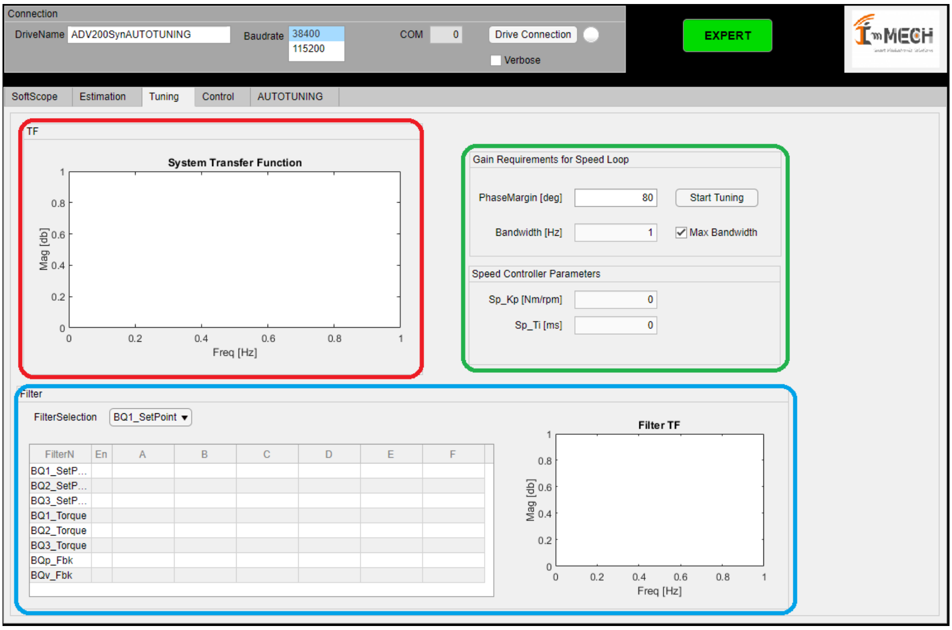
System Name Experiment

AcqSel 1

Friction [Nms/rad] 0
Inertia [kgm2] 0

pole 0

Figure 3.21: HMI for the application of the autotuning solutions, estimation window. The torque, velocity and position limits that must be respected during the excitation phase can be defined by changing the values inside the red box. The data for the excitation (e.g. max. and min. chirp frequencies) are defined by the values in the green box. The commands are given by means of the buttons inside the orange button, where it is possible to launch different identification procedures (e.g. low frequencies identification, full-range frequencies identification). The results are then shown in the graphs of the blue box.



DriveName ADV200SynAUTOTUNING

Baudrate 38400 115200

COM 0

☐ Verbose

PhaseMargin [deg] 80

Bandwidth [Hz] 1

☑ Max Bandwidth

Sp_Kp [Nm/rpm] 0

Sp_Ti [ms] 0

FilterSelection BQ1_SetPoint

FilterN	En	A	B	C	D	E	F
BQ1_SetP...							
BQ2_SetP...							
BQ3_SetP...							
BQ1_Torque							
BQ2_Torque							
BQ3_Torque							
BQp_Fbk							
BQv_Fbk							

Figure 3.22: Tuning phase interface. The Bode plot of the identified estimated transfer function is shown in the graph of the red box. The desired requirements, in terms of phase margin and bandwidth, can be specified in the green box, where the tuning procedure can be started. It is also possible to insert a manually tuned biquadratic filter by selecting the appropriate parameters in the blue box.

DriveName ADV200SynAUTOTUNING

Baudrate 38400 115200

COM 0

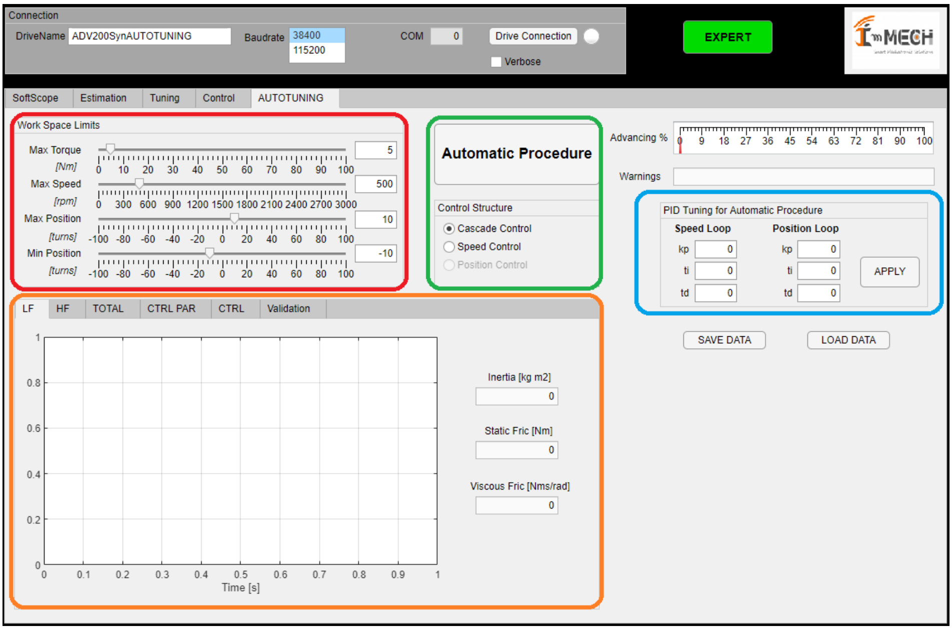
Kp [Nm/rpm] 0

Ti [ms] 0

StMinSp 200

StMaxSp 500

Figure 3.23: Validation interface. The tuned control is graphically shown in the scheme inside the red box, where the various filters can be switched on and off. The validation tests can be launched acting on the buttons inside the blue box, and the resulting performance can be compared to the expected ones by checking the graphs inside the green box.



Connection

DriveName: ADV200SynAUTOTUNING

Baudrate: 38400 / 115200

COM: 0

☐ Verbose

Work Space Limits

Max Torque [Nm]: 5

Max Speed [rpm]: 500

Max Position [turns]: 10

Min Position [turns]: -10

Control Structure

● Cascade Control

○ Speed Control

○ Position Control

PID Tuning for Automatic Procedure

	Speed Loop	Position Loop
kp	0	0
ti	0	0
td	0	0

Inertia [kg m2]: 0

Static Fric [Nm]: 0

Viscous Fric [Nms/rad]: 0

Figure 3.24: Autotuning procedure interface. While the other interfaces are accessible only in expert operator mode, the completely automatic tuning procedure can always be launched. The constraints on the system are set in the red box. The automatic procedure can then be launched from the button inside the green box. The tuned control parameters are shown inside the blue box after the identification and tuning phase. The operator must push the button to apply the found parameters to the controller. The performance of the tuned can finally be validated by checking the various available graphs inside the orange block.

3.6 Conclusions and contributions

Answering to the needs of industrial mechatronic systems for the increased performance that can be brought by ad-hoc control schemes and parameters tuning, this chapter has investigated the implementability of automatic procedures for the tuning of the control parameters and the suppression of oscillatory dynamics.
The proposed algorithms, differently from the solutions present in literature, explicitly take into account the limits of the system, which are set by the operator depending on the specific application on which the tuning procedure must be applied. The constraints that are possible to include are position, velocity and torque limits. Not only these constraints are respected during the algorithm identification phase, but the trajectory executed during this phase also brings the system near its limits, resulting in identified model that takes into account data spanning from all over the system workspace, resulting in a more accurate model of the system.

Moreover, differently from the procedures present in literature, all the proposed algorithms directly address the presence of oscillatory dynamics due to the presence of non-rigid shafts. These dynamics degrade the overall performance of the system, in particular in the case where only sensors on the load side are available. In this case, a high-bandwidth control of the motor is possible, but it could even result in a worsening of the positioning performance on the load side if the dynamics introduced by the elastic shaft is not taken into account, as demonstrated by the experiments in Section 3.4. The classical solution in presence of oscillatory dynamics is to lower the bandwidth of the controlled system, decreasing the overall performance of the system. Instead, the proposed algorithms are based on the identification and compensation of the oscillatory dynamics by means of properly calibrated biquadratic filters. The cancellation of the oscillatory dynamics is performed not only on the motor control loop, but also on the load in an open-loop fashion. The fact that this is possible by analyzing data acquired only on the motor side, as it is the standard for industrial mechatronic systems, is an important step forward compared to classical autotuning techniques.

These improvements have been included in control techniques that act, at the operator's choice, both with open-loop and closed-loop control systems. This possibility of choice makes it possible to apply the proposed techniques to virtually all industrial applications.

The techniques have been demonstrated both through simulations in the Simulink/Matlab environment, and by using an experimental setup based on industrial hardware, effectively verifying the applicability of the approach in the industrial sector through implementation on Hardware-In-the-Loop (HIL) setup.

The results of this work were provided to the partners of the I-MECH project as part of Building Block 6 (BB6), respecting the requests for interoperability with the other BBs of the project and providing a clear and complete description of the functionality of the algorithms and of the interface, with particular focus on the delineation of the inputs necessary for the self-calibration block, the selectable parameters and the subsequent outputs.

The use of the algorithms presented through an intuitive and effective user interface (GUI) has also been made available. This allows the operator to quickly calibrate the control system during system commissioning, but it also allows the expert operator to access additional parameters for a more precise calibration, which obviously requires the intervention and is reserved to expert personnel.

Finally, it is necessary to underline how the techniques proposed in this area have been implemented and tested on an experimental setup composed entirely of industrial type hardware (ADV200S drive and Gefran brushless motors). This makes them immediately integrable in different industrial applications.

CHAPTER 4

Iterative and repetitive control

Each industrial mechatronic system, during its operation, is affected by different types of disturbances which affect the performance of the system. These disturbances can come from the outside, or they can be intrinsic to the system itself (for example caused by eccentric bodies in rotation, by oscillating effects generated by electric motors, etc.). In general, if the disturbances come from the outside, these are not known, while if the disturbances are generated by the system itself (due, for example, to unbalanced rotating masses) these are or can be known. If the disturbances are known and have a periodic or repetitive character, they can be compensated thanks to the use of particular controllers, to be added to the feedback control structure, which allow, in the ideal case, the perfect tracking of the reference by completely compensating for these disturbances [72].

In this chapter, different repetitive and iterative control strategies are described, and their implementation in standard industrial control hardware is addressed and validated. These strategies allow the complete or partial suppression (or attenuation) of periodic or repetitive disturbances affecting generic mechatronic systems.

4.1 State of the art

The suppression of repetitive and/or periodic disturbances is a much discussed topic in the control field, and various repetitive or iterative controllers have been proposed in the last 30 years [77].

The repetitive controller is usually applied independently of the used feedback control [16]. This type of control ensures very high control gain at the frequency of the noise. However, the performance of the method drops dramatically if the disturbance period is not perfectly known, or if this period varies over time [70] such as in the case of position-dependent disturbances [14].

Different approaches have been used to try to solve the problem of the variation of the disturbance period. These approaches are divided substantially into two groups according to the applied methodology:

- approaches based on the variation of the relationship between the disturbance period and the control period, approximating this relationship to the nearest integer [37, 73];
- approaches based on the variation of the control period, keeping the relationship between the disturbance period and the control period constant by approximating this value to the nearest integer [3, 11, 33, 70, 71, 73].

However, the performance of these approaches is effective only when the variation of the disturbance period is not excessive, and especially if its variation is not too rapid. An excellent survey on repetitive control can be found in [34].

Iterative learning control, a more recent approach with respect to repetitive control, can be applied both to improve the transient response of the systems and to suppress repetitive and/or periodic disturbances, especially in cases where the system continuously repeats the same operations [47]. The concept behind iterative control is simple: at each repeated cycle, the controller modifies its control action in order to minimize the control error. The main advantage of the classic iterative control lies in the fact that it is not necessary to have a precise mathematical model of the system to be controlled [8], even if, in some cases [41], the system model can be useful in order to keep into consideration any constraints and find optimal learning solutions.

Just like the classic repetitive control, iterative control is usually applied independently of the feedback control. Iterative control, in fact, is used for the compensation of repetitive errors, while the feedback control on top of which it works is used to compensate also generic disturbances acting on the system. For an insight on repetitive control, consider [2].

The main difference between iterative controllers and repetitive controllers lies in the fact that while the repetitive controllers are suitable for continuous and non-varying operations (e.g. movements at constant speed), iterative controllers can also be employed for discontinuous operations (such as the movements of industrial robots). This difference can be seen as residing in the conditions of the two methods. The initial conditions of the repetitive controllers, at each cycle, depend on the final conditions of the previous cycle, while the initial conditions of the iterative controllers are always the same for each cycle [44].

The main contribution of this chapter is the implementation and testing of state-of-the-art techniques in the field of iterative and repetitive control directly in industrial drives. The implemented technique are validated on a HIL testbed composed by industrial hardware, demonstrating the effectiveness of the approach.

4.2 Repetitive Control (RC)

Repetitive control in the time domain [16, 34, 77] is a type of control based on the principle of the internal model [19] which aims at the perfect tracking of the reference, in steady state, in the presence of periodic or repetitive disturbances [72]. The general scheme of a repetitive controller is shown in the area surrounded by the red line in Figure 4.1.

The internal model of a repetitive controller, which is basically a periodic signal generator, corresponds to the scheme within the area surrounded by the green dotted line. This model allows adding to the current input of the controller the input of the controller at a certain previous instant. Block D, in fact, represents a definite time delay

$$D(s) = e^{-Ls}, \tag{4.1}$$

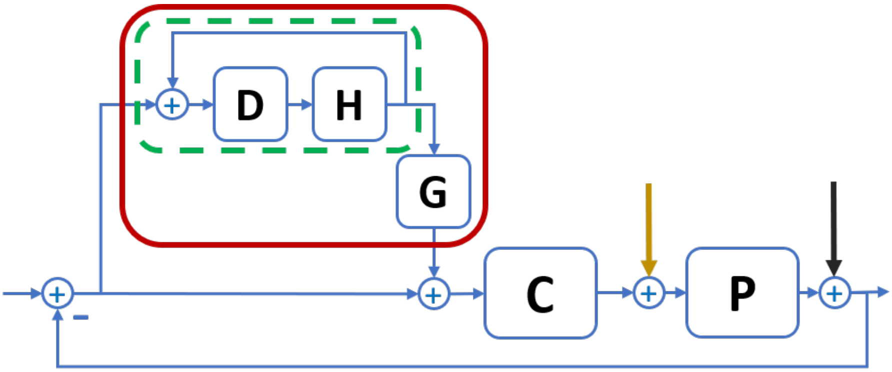


Figure 4.1: Repetitive control scheme in the time domain. Orange arrow: load disturbance. Black arrow: disturbance on the output.

which, in a discrete form, becomes

$$D(z) = z^{-(N+d-1)} \tag{4.2}$$

where $N = \text{round}(T_N/T_s)$ with T_N the disturbance period, T_s is the sampling period of the repetitive controller and d an adjustment parameter that allows taking into account possible delays within the feedback control loops. The $H(s)$ function represents a generic stability filter, which is a low pass filter which allows cutting down high frequency noises and can be written in the form

$$H(s) = K_r \frac{1}{\tau_f s + 1} \tag{4.3}$$

where τ_f is the time constant of the filter and K_r is the gain of the repetitive controller. The $G(s)$ function, on the other hand, has the task of correctly phase-shifting and amplifying the output signal from the internal model in order to ensure the suppression of the noise on the output in the correct manner. Indeed,

$$G(s) = \frac{1}{F(s)}, \tag{4.4}$$

where

$$F(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.5}$$

represents the transfer function between the repetitive controller input and the system output, where $C(s)$ is the transfer function of the controller while $P(s)$ is the transfer function of the system to be controlled.

4.2.1 MIL validation

RC control has been validated by means of MIL testing.

The simulated and co-simulated results were obtained considering a generic mechatronic system with the following characteristics:

- total inertia $J = 0.3$ [kg/m^2];
- static friction $c_0 = 0.2$ [Nm];
- viscous friction $c_1 = 0.02$ [Nms/rad];

The linear model of the system can be obtained by not considering the effects of the static friction, thus obtaining the transfer function

$$P(s) = \frac{1}{Js + c_1} = \frac{1}{0.3s + 0.02}. \tag{4.6}$$

The PI controller used has been calibrated with the following parameters:

- proportional gain $K_p = 5.0$ [Nms/rad];
- integral time constant $T_i = 1.0$ [s];
- sampling interval $T_s = 0.001$ [s].

The transfer function of the controller is

$$C(s) = k_p \left(1 + \frac{1}{t_i s}\right) = 5.0 \left(1 + \frac{1}{1.0s}\right). \tag{4.7}$$

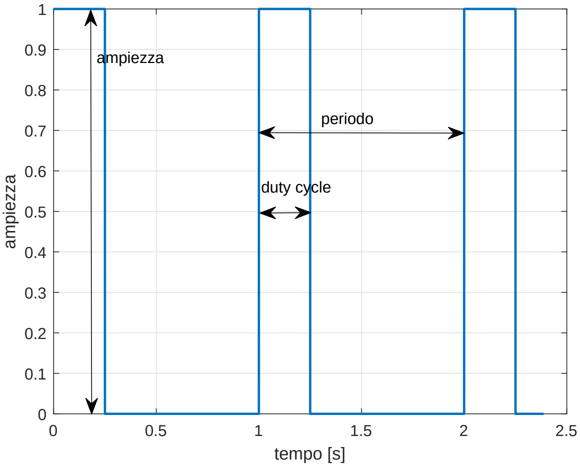


Figure 4.2: Generic form of the disturbance used in simulations and co-simulations.

The disturbance used in the simulations and co-simulations has the form shown in Figure 4.2.

For the MIL validation, the approach has first been tested with both the controller and the system running on Simulink. Later, the model of the system, together with the external disturbances, have been implemented on Simcenter Amesim, and the controller was run on Simulink in co-simulation.

The repetitive controller parameters used are:

- repetitive controller gain $K_r = 0.4$;
- cutoff frequency of the stability filter $\omega_c^f = 490$ [Hz];
- delay adjustment parameter $d = -2$.

The simulated disturbances, positioned at the input and output of the system, with reference to Figure 4.2, have been created with the following characteristics:

- input disturbance period $T_N^i = 1.0$ [s];
- input disturbance duty cycle $dC_d^i = 10.0$ [%];
- input disturbance amplitude $A_d^i = 10.0$ [Nm];
- output disturbance period $T_N^u = 1.0$ [s];
- output disturbance duty cycle $dC_d^u = 10.0$ [%];
- input disturbance amplitude $A_d^u = 1.0$ [rad/s].

The stability filter $H(s)$ used in the simulations, in accordance with the data listed above, has a transfer function

$$H(s) = K_r \frac{1}{\tau_f s + 1} = 0.4 \frac{1}{\frac{1}{490 \cdot 2 \cdot \pi} s + 1}. \tag{4.8}$$

The $G(s)$ function, in the simulations of the repetitive control, is given by

$$G(s) = \frac{1}{F(s)}, \tag{4.9}$$

where

$$F(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.10}$$

represents the transfer function between the repetitive controller input and the system output.

In Figure 4.3 the Simulink scheme used for the simulations is shown. As can be seen, two sub-schemes are simulated in parallel. The sub-diagram in the centre of the Figure 4.3 (blue area) simulates a standard controller in feedback, while the sub-scheme in the bottom of the Figure 4.3 (orange area) simulates a standard controller aided by a repetitive controller activated by the command "RC_start".

These simulations were carried out to understand if the repetitive controller could give benefits on generic mechatronic systems; in addition they

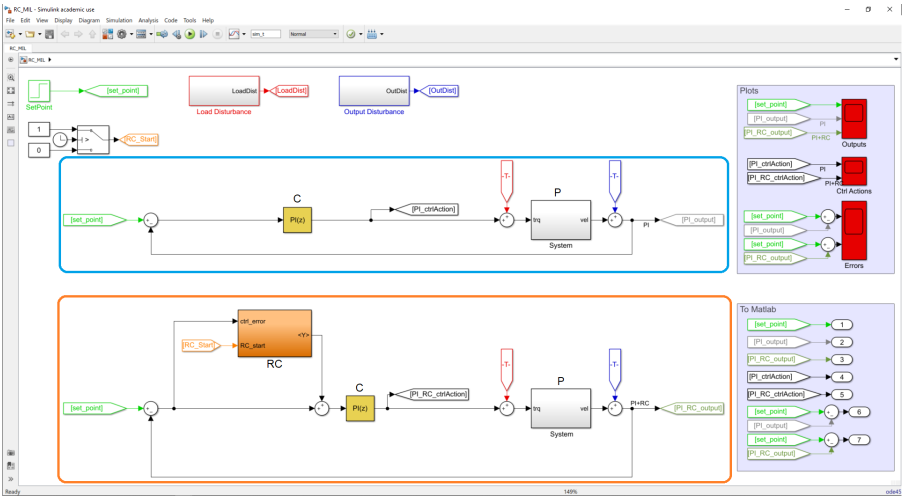


Figure 4.3: Simulink scheme used for MIL tests.

served to verify the correctness of the implementation of the repetitive controller.

Figure 4.4 shows the performance comparison between the output of the simulated system using a single feedback controller (upper graph) and the output of the simulated system using a feedback controller aided by a repetitive controller activated after 10 seconds of simulation (lower graph). Shortly after the activation of the repetitive controller, its effects begin to be relevant and the output remains much more adherent to the desired output (blue line). This is shown in Figure 4.5, where control errors are compared. In blue the error obtained using a single feedback controller, in red the error obtained using the feedback controller supported by the repetitive controller. In Figure 4.6 the respective control actions in the two cases are shown.

In Figure 4.7 the Simulink scheme used for co-simulations with Amesim is shown. The sub-scheme at the centre of Figure 4.7 (blue area) simulates a standard controller in feedback, while the sub-scheme in the lower part

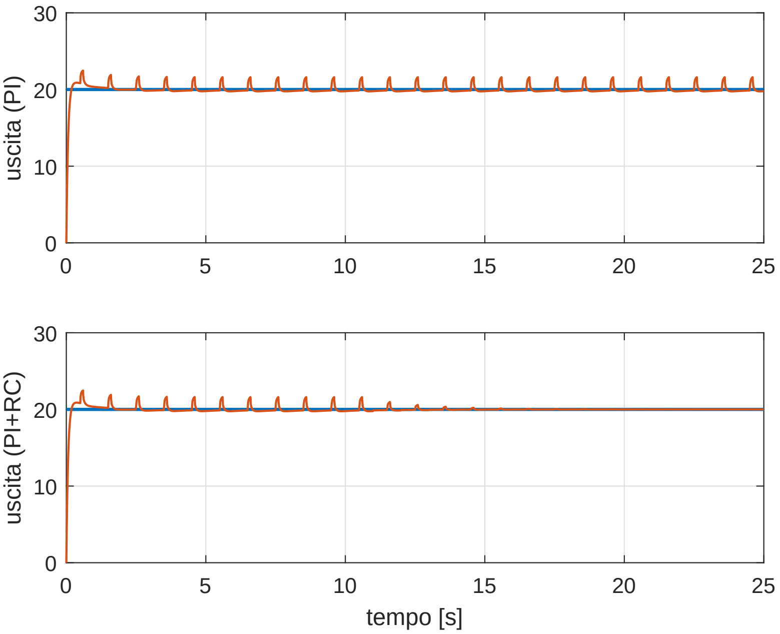


Figure 4.4: Output controlled with a PI controller (upper graph) and output controlled with a PI controller with the addition of the repetitive controller activated after 10 seconds (lower graph). The reference in both cases is set constant equal to 20.

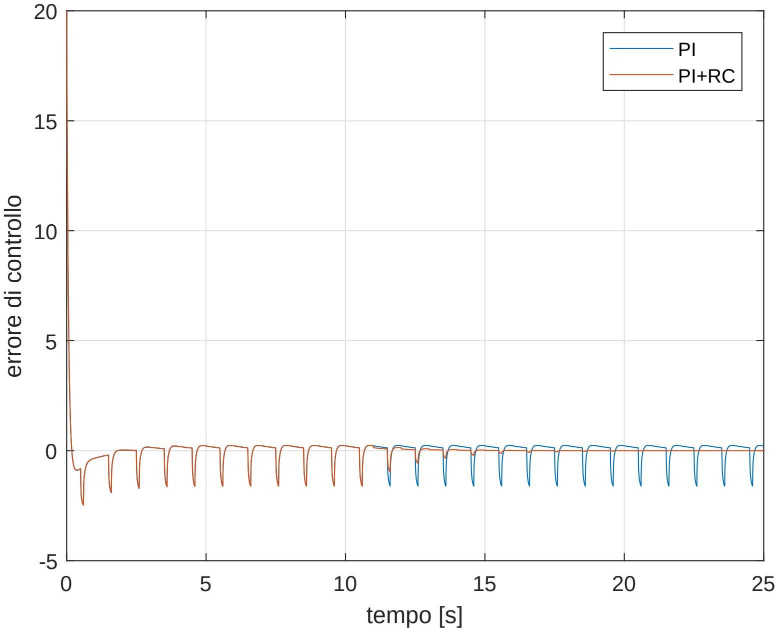


Figure 4.5: Comparison between the control error obtained with a PI controller (blue line) and the control error obtained with a PI controller and the addition of the repetitive controller activated after 10 seconds (red line).

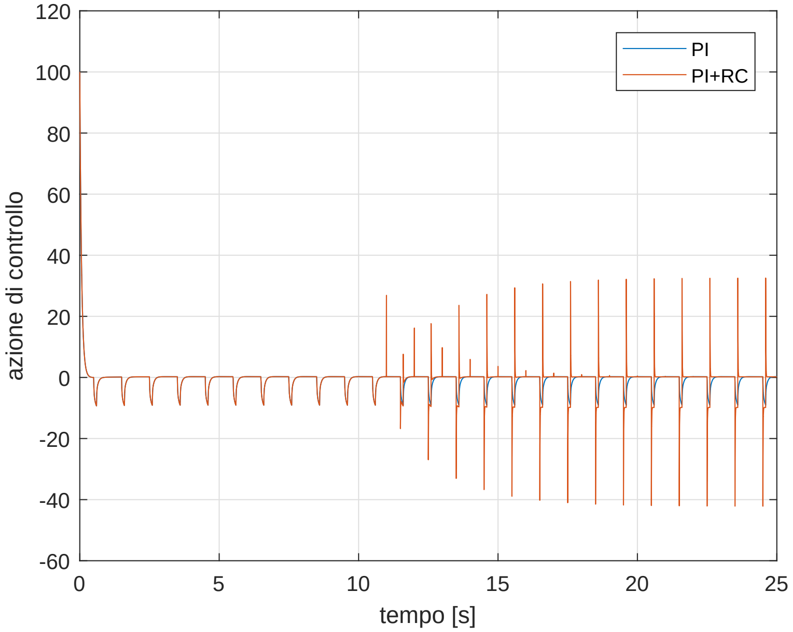


Figure 4.6: Comparison between the control action calculated with a PI controller (blue line) and the control action calculated with a PI controller and the addition of the repetitive controller activated after 10 seconds (red line).

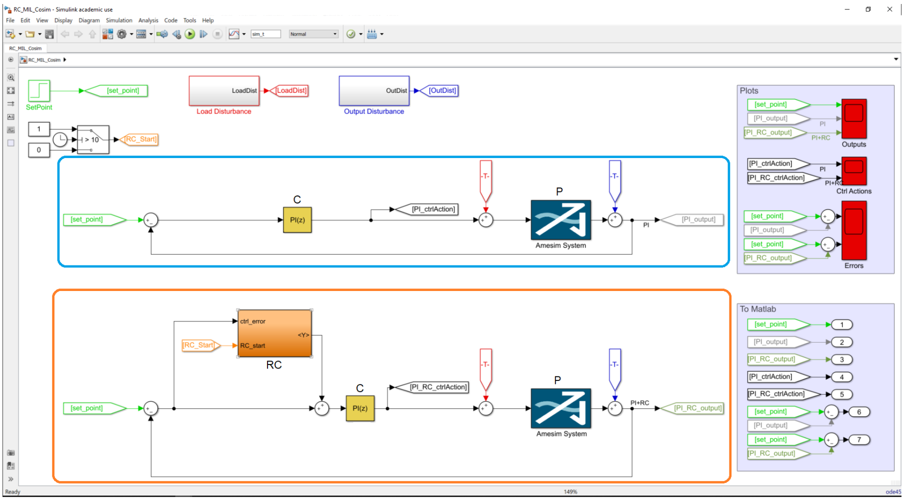


Figure 4.7: Simulink scheme used for MIL tests done in Simulink / Amesim simulation.

of the Figure 4.7 (orange area) simulates a standard controller aided by a repetitive controller activated by the command "RC_start".

Figure 4.8 shows the comparison of performance between the output of the simulated system with Amesim using a single feedback controller (upper graph) and the output of the simulated system using a feedback controller aided by a repetitive controller activated after 10 seconds of simulation (lower graph). As can also be seen in the co-simulation, shortly after the activation of the repetitive controller its effects begin to be relevant and the output of the system simulated with Amesim remains much more adherent to the desired output (blue line). This data is shown in Figure 4.9, where the control errors are compared. In blue the error obtained using a single feedback controller, in red the error obtained using the feedback controller supported by the repetitive controller. In Figure 4.10 the respective control actions in the two cases are shown.

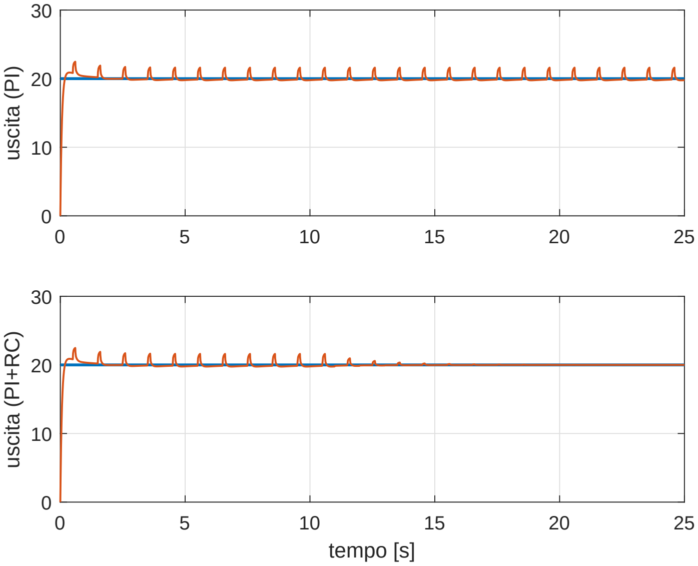


Figure 4.8: Output controlled with a PI controller (upper graph) and output controlled with a PI controller with the addition of the repetitive controller activated after 10 seconds (lower graph). The reference in both cases is set constant equal to 20.

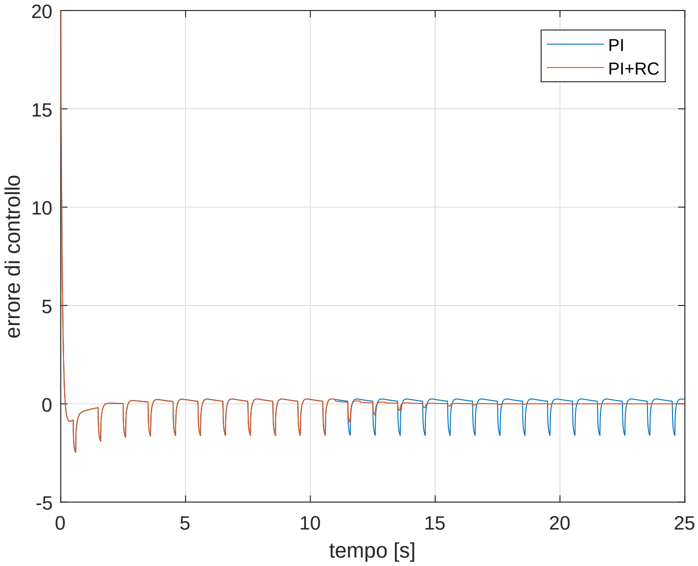


Figure 4.9: Comparison between the control error obtained with a PI controller (blue line) and the control error obtained with a PI controller and the addition of the repetitive controller activated after 10 seconds (red line).

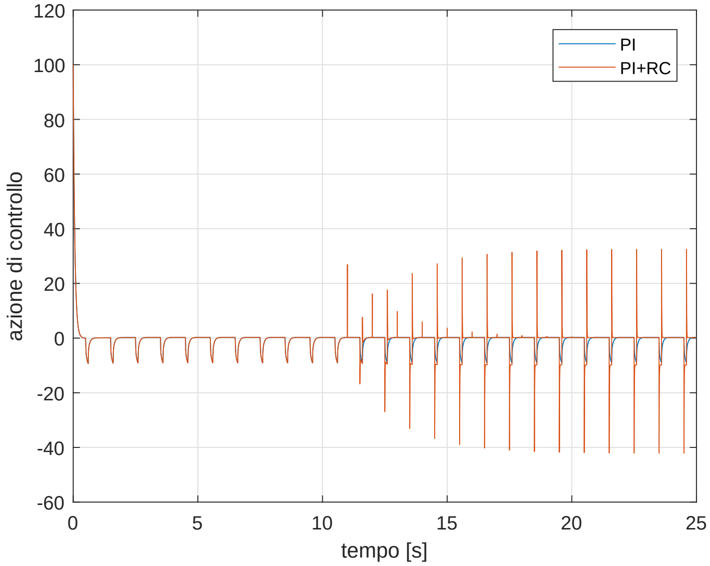


Figure 4.10: Comparison between the control action calculated with a PI controller (blue line) and the control action calculated with a PI controller and the addition of the repetitive controller activated after 10 seconds (red line).

4.2.2 HIL validation

In order to validate the implementation of the technique in industrial servo drives, the technique has been implemented in the ADV200 Gefran drive of the HIL setup described in Section 1.4.4.
The parameters of the HIL setup shown in Figure 1.6 have been identified as

- total inertia $J = 0.032$ [kg / m ^{2}] ;
- static friction $c_0 = 0.245$ [Nm] ;
- viscous friction $c_1 = 0.03$ [Nms / rad].

The SM has been used only for introducing the repetitive disturbance, as the HIL setup already presented an ideal test bench for the application of RC and ILC approaches.

The transfer function of the HIL setup is, therefore,

$$P(s) = \frac{1}{Js + c_1} = \frac{1}{0.032s + 0.03}. \tag{4.11}$$

The PI controller used has been calibrated with the following parameters:

- proportional gain $k_p = 0.1$ [Nms/rad]
- integral time constant $t_i = 1.0$ [s]
- sampling interval $T_s = 0.001$ [s].

The transfer function of the controller is

$$C(s) = k_p \left(1 + \frac{1}{t_i s}\right) = 0.1 \left(1 + \frac{1}{1.0s}\right). \tag{4.12}$$

In order to use common industrial drives with a reduced memory capability, the sampling period of the repetitive and iterative controllers has been modified in order to obtain memory vectors of a maximum predefined length (in the cases under examination 200 elements). To do this, the ratio between the disturbance period and the control period was kept constant at a value of 200:

$$N = \frac{T_N}{T_s} = 200. \tag{4.13}$$

The disturbance, introduced by the SM motor, has the same form as the one shown in Figure 4.2.

The parameters of the repetitive controller used were:

- repetitive controller control period $T_s = 0.01\ [s]$;
- repetitive controller gain $K_r = 1.0$;
- cutoff frequency of the stability filter $\omega_c^f = 30\ [Hz]$;
- delay adjustment parameter $d = -2$.

The simulated disturbance is a square wave on the system input, was created with the following characteristics:

- input disturbance period $T_N^i = 1.0\ [s]$;
- duty cycle of the input disturbance $dC_d^i = 50.0\ [\%]$;
- input disturbance amplitude $A_d^i = -2.0\ [Nm]$.

The stability filter H used in the experimental tests of the repetitive control is a first-order low-pass filter with transfer function

$$H(s) = K_r \frac{1}{\tau_f s + 1} = 1.0 \frac{1}{\frac{1}{30 \cdot 2 \cdot \pi} s + 1}. \tag{4.14}$$

The G function, in the experimental tests of the repetitive control, is given by

$$G(s) = \frac{1}{F(s)}, \tag{4.15}$$

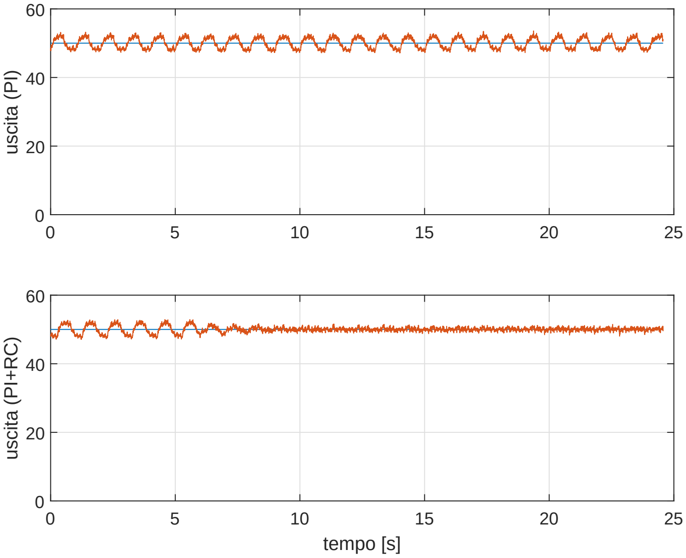


Figure 4.11: Controlled output on the test bench with a PI controller (upper graph) and controlled output with a PI controller with the addition of the repetitive controller activated after 5 seconds (lower graph). The reference in both cases is set constant equal to 50.

where

$$F(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.16}$$

represents the transfer function between the repetitive controller input and the system output.

Figure 4.11 shows the performance comparison between the output of the physical test bench using a single feedback controller (upper graph) and the output of the physical test bench using a feedback controller aided by a repetitive controller activated after 5 seconds of acquisition (lower graph).

The results obtained with the HIL validation of the RC method are

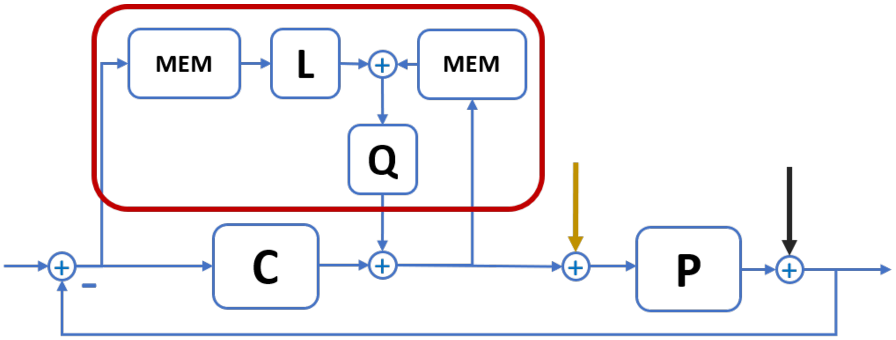


Figure 4.12: Iterative control scheme. Orange arrow: load disturbance. Black arrow: disturbance on the output.

compatible with the ones obtained during the MIL validation.

4.3 Iterative Learning Control (ILC)

The idea behind iterative learning control consists in the fact that the performance of a system which repeatedly faces the same task, while facing the same disturbance, can be improved by learning from the conditions already faced in the previous cycles [8]. Its basic scheme is shown inside the area surrounded by the red line in Figure 4.12.

ILC behaves differently than other types of learning controllers such as adaptive controllers, neural networks or repetitive controllers. ILC, in fact, does not modify the parameters of the feedback controller (as, for example, occurs in the case of adaptive controllers), but directly modifies the control action [46].

Iterative learning controllers work in the following way: at each instant k of the cycle j the control action u to be given to the system is formed by the sum of the control action of the feedback c and by the action of control provided by the iterative controller w

$$u_k^j = c_k^j + w_k^j. \tag{4.17}$$

The control action provided by the ILC controller at the instant k of the current cycle j (w_k^j) depends on the total control action at the instant k of the previous cycle (u_k^{j-1}) and from the control error instantly k of the previous cycle (e_k^{j-1}) combined considering the learning functions L_f and the stability filter Q in the formula

$$w_k^j = Q\left[e_k^{j-1} L_f + u_k^{j-1}\right]. \tag{4.18}$$

The learning function L_f allows modifying at each cycle the control action provided to the previous cycle in order to minimize errors on the output, while the stability filter Q stabilizes the system.
The blocks MEM represent a definite time delay

$$MEM(s) = e^{-Ls}, \tag{4.19}$$

which, in the discrete domain, becomes

$$MEM(z) = z^{-(N+d-1)} \tag{4.20}$$

where $N = \text{round}(T_N/T_s)$ with T_N the cycle period, T_s the sampling period of the ILC controller and d is an adjustment parameter that allows taking into account possible delays within the feedback control loops.

The transfer function of the block Q, which represents a generic stability filter, is

$$Q(s) = \frac{1}{\tau_f s + 1} \tag{4.21}$$

where τ_f is the time constant of the filter. The transfer function of the learning function L_f can be set to

$$L_f(s) = \frac{K_l}{s} \tag{4.22}$$

where K_l represents the learning gain of the iterative controller.

4.3.1 MIL validation

As done for the validation of the RC approach, the MIL validation of ILC has first been tested with both the controller and the system running on Simulink; later, the model of the system, together with the external disturbances, have been implemented on Simcenter Amesim, and the controller was run on Simulink in co-simulation. The mechatronic system parameters are the same ones used for the MIL validation of the RC method.
In the case of the simulations and co-simulations for the testing of ILC control, the following control parameters have been used:

- iterative controller gain $K_l = 0.4$;
- cutoff frequency of the stability filter $\omega_c^f = 490$ [Hz];
- delay adjustment parameter $d = -2$.

The simulated disturbances have been constructed as square waves at the input and output of the system with the following parameters:

- input disturbance period $T_N^i = 1.0$ [s];
- duty cycle of the input disturbance $dC_d^i = 10.0$ [%];
- input disturbance amplitude $A_d^i = 10.0$ [Nm];
- output disturbance period $T_N^u = 1.0$ [s];
- output disturbance duty cycle $dC_d^u = 10.0$ [%];
- output disturbance amplitude $A_d^u = 1.0$ [rad/s].

The transfer function of the Q block, according to the data listed above, is

$$Q(s) = \frac{1}{\tau_f s + 1} = \frac{1}{\frac{1}{490 \cdot 2 \cdot \pi} s + 1} \tag{4.23}$$

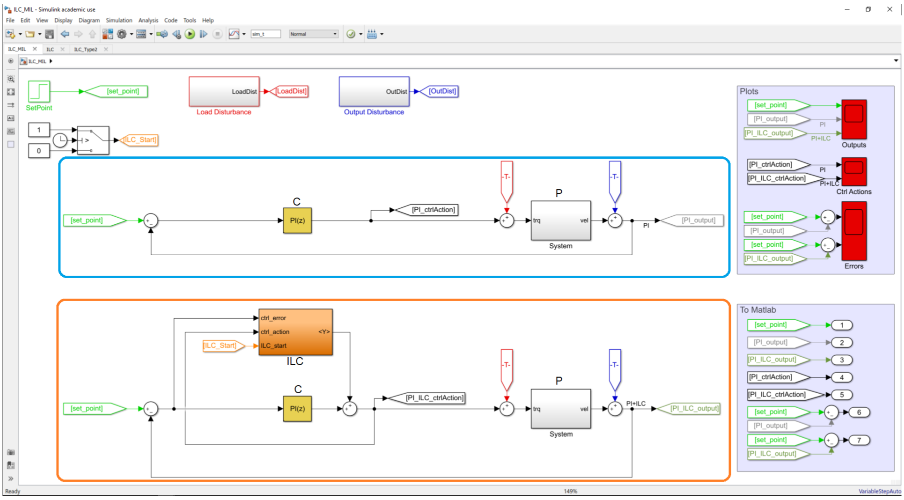


Figure 4.13: Simulink scheme used for the MIL validation of the ILC method.

while the transfer function of the learning function L_f is

$$L_f(s) = \frac{K_l}{s} = \frac{0.4}{s}. \tag{4.24}$$

In Figure 4.13 the Simulink scheme used for the simulations is shown. As can be seen, two sub-schemes simulate in parallel. The sub-scheme in the centre of the Figure 4.13 (blue area) simulates a standard controller in feedback, while the sub-scheme in the lower part of the Figure 4.13 (orange area) simulates a standard controller aided by an iterative controller activated by the command "ILC_start".

In Figure 4.13 the Simulink scheme used for the simulations is shown. These simulations were carried out to understand if the ILC controller could give benefits on generic mechatronic systems, also they served to verify the implementation of the ILC controller. Figure 4.14 shows the performance comparison between the output of the simulated system using a single feedback controller (upper graph) and the output of the simulated system using

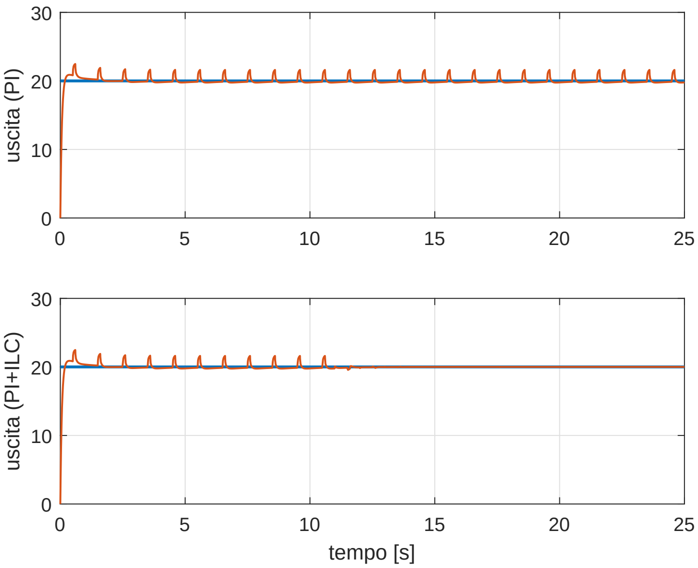


Figure 4.14: Output controlled with a PI controller (upper graph) and output controlled with a PI controller with the addition of the iterative controller activated after 10 seconds (lower graph). The reference in both cases is set constant equal to 20.

a feedback controller aided by an iterative controller activated after 10 seconds of simulation (lower graph). Shortly after the activation of the iterative controller its effects begin to be seen and the simulated output deviates less from the desired output (blue line) when compared to the standard feedback approach.
In Figure 4.15 the control errors are compared. In blue the error obtained using a single feedback controller, in red the error obtained using the feedback controller supported by the iterative controller. In Figure 4.16 the respective control actions in the two cases are shown.

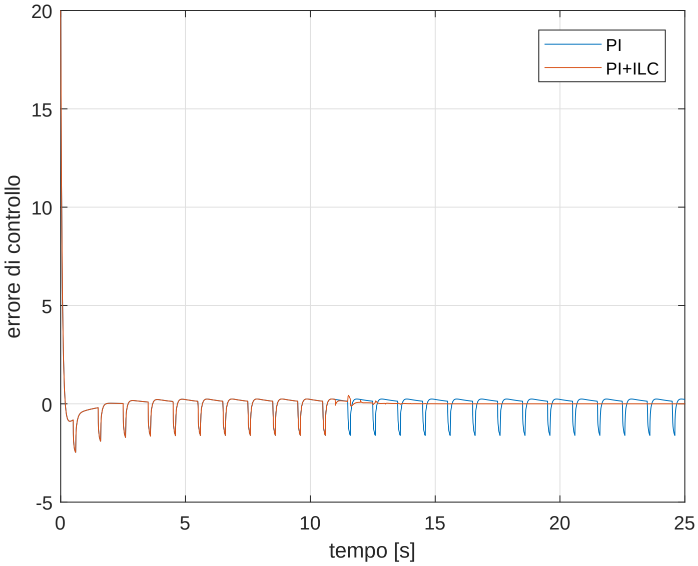


Figure 4.15: Comparison of the control error obtained with a PI controller (blue line) and the control error obtained with a PI controller and the addition of the iterative controller activated after 10 seconds (red line).

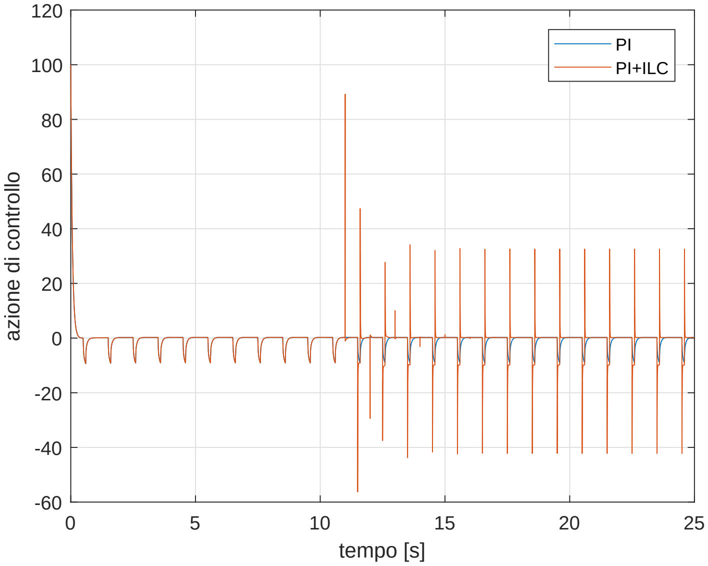


Figure 4.16: Comparison between the control action calculated with a PI controller (blue line) and the control action calculated with a PI controller and the addition of the iterative controller activated after 10 seconds (red line).

4.3.2 HIL validation

For the HIL testing of ILC controllers, the HIL hardware in Figure 1.6 has been used. The parameters of both the system and the feedback controller are the same ones as in the case of the HIL validation of the RC method described in Chapter 4.2.2.

The parameters of the iterative controller used are:

- control period of the iterative controller $T_s = 0.005$ [s];
- iterative controller gain $K_l = 0.00002$;
- cutoff frequency of the stability filter $\omega_c^f = 50$ [Hz];
- delay adjustment parameter $d = -2$.

The simulated disturbance, a square wave on the system input, was created with the following characteristics:

- input disturbance period $T_N^i = 1.0$ [s];
- input disturbance duty cycle $dC_d^i = 50.0$ [%];
- input disturbance amplitude $A_d^i = -2.0$ [Nm].

The transfer function of the Q block, according to the data listed above, is

$$Q(s) = \frac{1}{\tau_f s + 1} = \frac{1}{\frac{1}{50 \cdot 2 \cdot \pi} s + 1} \tag{4.25}$$

while the transfer function of the learning function L_f is

$$L_f(s) = \frac{K_l}{s} = \frac{0.00002}{s}. \tag{4.26}$$

It is important to underline that, in order to use common industrial drives with a reduced memory capability, the sampling period of the iterative controller has been modified in order to obtain memory vectors of a maximum predefined length (in the cases under examination 200 elements). To do this, the ratio between the disturbance period and the control period was kept constant at a value of 200:

$$N = \frac{T_N}{T_s} = 200, \tag{4.27}$$

where T_N is the disturbance period and T_s is the sampling period of the iterative controller.

Figure 4.17 presents a performance comparison between the output of the physical test bench using a single feedback controller (upper graph) and the output of the physical test bench using a feedback controller aided by an iterative controller activated after 5 seconds of acquisition (lower graph).

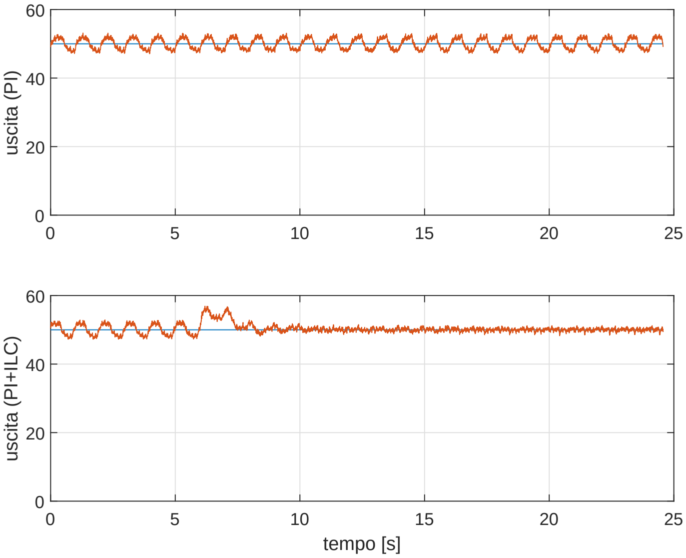


Figure 4.17: Controlled output on the test bench with a PI controller (upper graph) and controlled output with a PI controller with the addition of the iterative controller activated after 5 seconds (lower graph). The reference in both cases is set constant equal to 50.

4.4 Position domain Repetitive Control (Pos-RC)

The repetitive control in the position domain here presented allows the suppression of disturbances that depend on the position of the system.
The scheme of a repetitive controller in the position domain is similar to that of a repetitive controller in the time domain, and is shown in Figure 4.18.

The internal model of a repetitive controller in the position domain, which corresponds to a periodic signal generator in position, is the scheme within the area surrounded by the green dotted line. This model allows the addition of the controller input to the current position with the controller input corresponding to the same position of the previous instant. Block D_p, in fact, represents a position delay. As in the case of the repetitive controller in the time domain, the function H represents a generic stability filter, which is a low-pass filter which allows reducing any noise in high frequency

$$H(s) = K_r \frac{1}{\tau_f s + 1} \tag{4.28}$$

where τ_f it is the time constant of the filter and K_r is the gain of the repetitive controller in the position domain. The function G, as in the case of the repetitive controller in the time domain, has the task of correctly phase-shifting and amplifying the output signal from the internal model in order to guarantee the suppression of the disturbance on the output in the correct way:

$$G(s) = \frac{1}{F(s)}, \tag{4.29}$$

where

$$F(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.30}$$

represents the transfer function between the repetitive controller input and the system output. $C(s)$ is the transfer function of the controller while $P(s)$ is the transfer function of the system to be controlled.

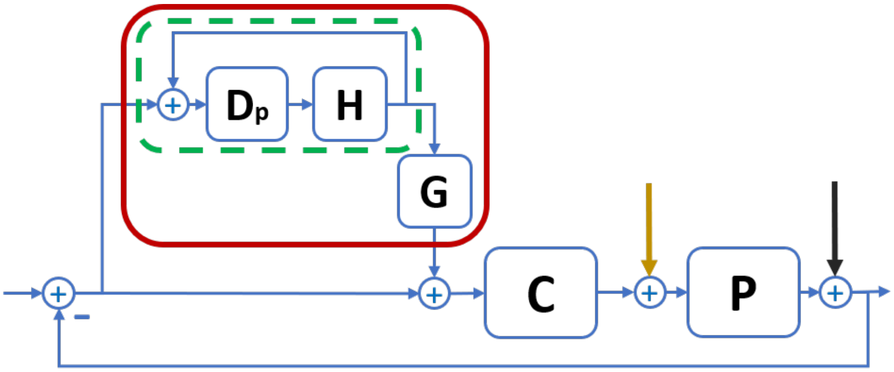


Figure 4.18: Repetitive control scheme. Orange arrow: load disturbance. Black arrow: disturbance on the output.

The block D_p, which as mentioned represents a position delay, can be implemented as a vector of predefined length N_p in which each position interval corresponds to an element of the vector. By analogy with respect to the case of the standard repetitive control, where the sampling period of the repetitive controller can be made to vary by keeping constant the ratio between the period of the disturbance and sampling period of the controller, that is

$$N = \frac{T_N}{T_s} = \text{ constant.} \tag{4.31}$$

in the case of repetitive control in the position domain, the sampling is performed based on the position of the system:

$$N_p = \frac{P_d}{\Delta_{P_d}} = \text{ constant} \tag{4.32}$$

where P_d is the periodicity of the position and Δ_{P_d} represents the sampling interval of the periodicity of the position.

The output of the internal model u of the repetitive controller in the position domain (area surrounded by the dashed green line in Figure 4.18) corresponding to the position interval Δ_{P_d} of the current cycle j, i.e. $u^j_{\Delta_{P_d}}$, depends on the control error on the range of position Δ_{P_d} of the current cycle, i.e. $e^j_{\Delta_{P_d}}$, added to the control error at the position interval Δ_{P_d} of the previous cycle ($e^{j-1}_{\Delta_{P_d}}$) filtered by the stability filter H. Its formulation can therefore be expressed as

$$IM^j_{\Delta_{P_d}} = e^j_{\Delta_{P_d}} + H\, e^{j-1}_{\Delta_{P_d}}. \tag{4.33}$$

The output w of the repetitive controller in the position domain will then be

$$w^j_{\Delta_{P_d}} = G\, IM^j_{\Delta_{P_d}}. \tag{4.34}$$

4.4.1 MIL validation

For the MIL validation of the proposed position repetitive control, the considered mechatronic system and the corresponding feedback controller are the same one described in Chapter 4.2.1.

The parameters of the repetitive controller in the position domain, for both the simulations running on Simulink and for the co-simulation cases, are:

- gain of the repetitive controller in the domain of the position $K_r = 1.0$;
- cutoff frequency of the stability filter $\omega_c^f = 490$ [Hz];
- length of the memory vector $N_p = 500$;
- periodicity of position $P_d = 2\pi$ [rad].

The simulated disturbance, acting on the system output, was created as the sum of three sinusoidal signals in the position domain:

$$d(t) = A_1 \sin(\omega t) + A_2 \sin(2\omega t) + A_3 \sin(4\omega t) \tag{4.35}$$

with the following characteristics:

- first sinusoid amplitude $A_1 = 0.33$ [Nm];
- frequency of the first sinusoid corresponding to the output speed of the system;
- second sinusoid amplitude $A_2 = 0.33$ [Nm];
- frequency of the second sine wave corresponding to twice the system output speed;
- third sinusoid amplitude $A_3 = 0.33$ [Nm];
- frequency of the third sine wave corresponding to four times the system output speed.

The stability filter H used in repetitive control simulations in the position domain is a first order low pass filter with transfer function

$$H(s) = K_r \frac{1}{\tau_f s + 1} = 1.0 \frac{1}{\frac{1}{490 \cdot 2 \cdot \pi} s + 1}. \tag{4.36}$$

The G function, in the simulations of repetitive control in the position domain, is given by

$$G(s) = \frac{1}{F(s)}, \tag{4.37}$$

where

$$F(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.38}$$

represents the transfer function between the repetitive controller input and the system output.

Figure 4.19 shows the Simulink scheme used for the simulations; two sub-schemes simulate in parallel. The sub-diagram in the centre of the Figure 4.19 (blue area) simulates a standard controller in feedback, while the sub-scheme in the bottom of the Figure 4.19 (orange area) simulates a standard controller aided by a controller repetitive in the position domain.

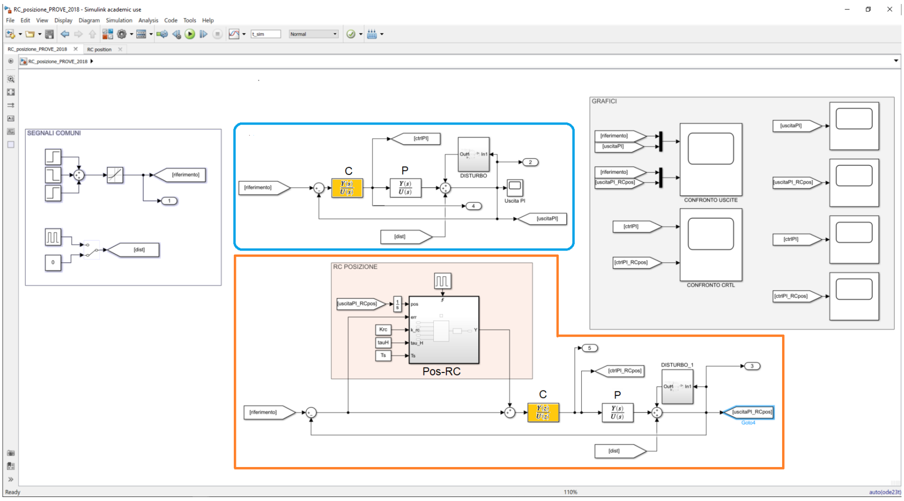


Figure 4.19: Simulink scheme used for MIL tests.

These simulations were carried out to understand if the repetitive controller in the position domain could give benefits on generic mechatronic systems. Figure 4.20 shows a performance comparison between the output of the simulated system using a single feedback controller (top graph) and the output of the simulated system using a feedback controller aided by a repetitive controller in the domain position (lower graph). The effects of the repetitive controller in the position domain are significant. The improved performance brought by the use of the position RC approach are also visible in Figure 4.21, where control errors are compared. In blue the error obtained using a single feedback controller, while in red the error obtained using the feedback controller supported by the repetitive controller in the position domain. In Figure 4.22 the respective control actions in the two cases are shown.

The position repetitive controller has then been validated by means of co-simulation between Simulink and Amesim, as shown by the scheme in Figure 4.23. The sub-scheme in the centre of the Figure 4.23 (blue area)

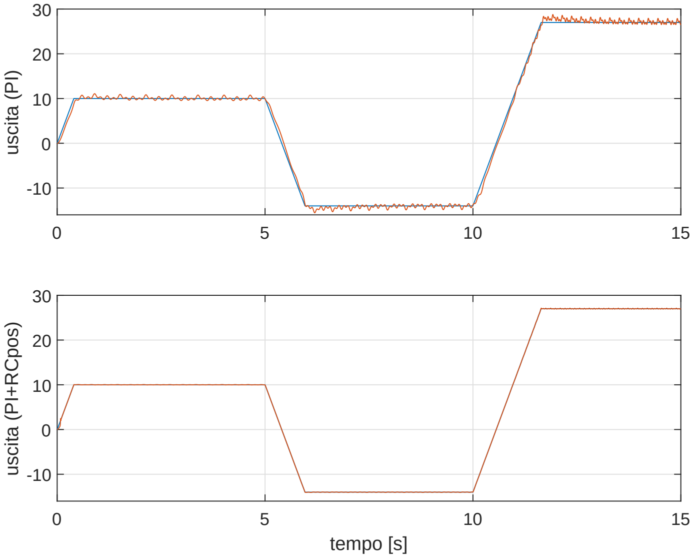


Figure 4.20: Output controlled with a PI controller (upper graph) and output controlled with a PI controller with the addition of the repetitive controller in position (lower graph). The reference in both cases is made to vary specifically to have different periods of disturbance over time.

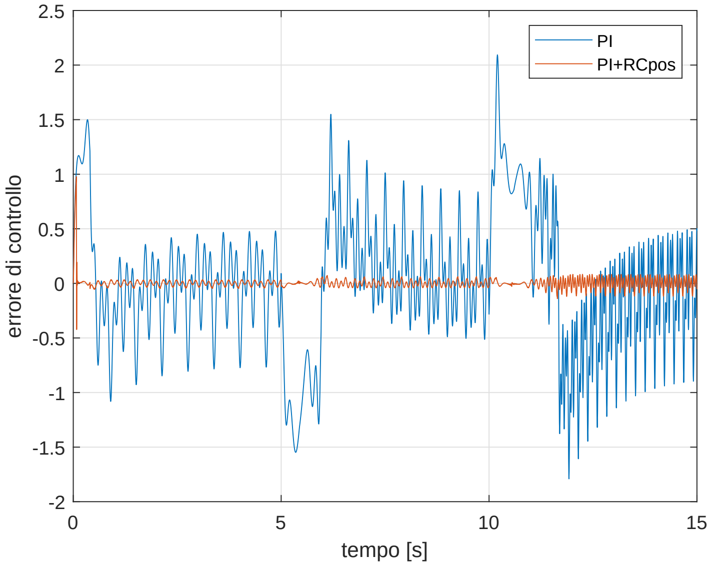


Figure 4.21: Comparison of the control error obtained with a PI controller (blue line) and the control error obtained with a PI controller and the addition of the repetitive controller in position (red line).

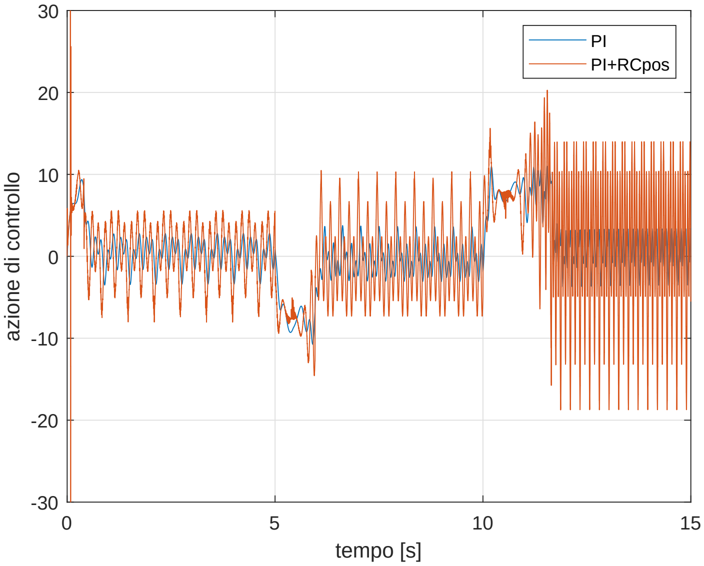


Figure 4.22: Comparison between the control action calculated with a PI controller (blue line) and the control action calculated with a PI controller and the addition of the repetitive controller in position (red line).

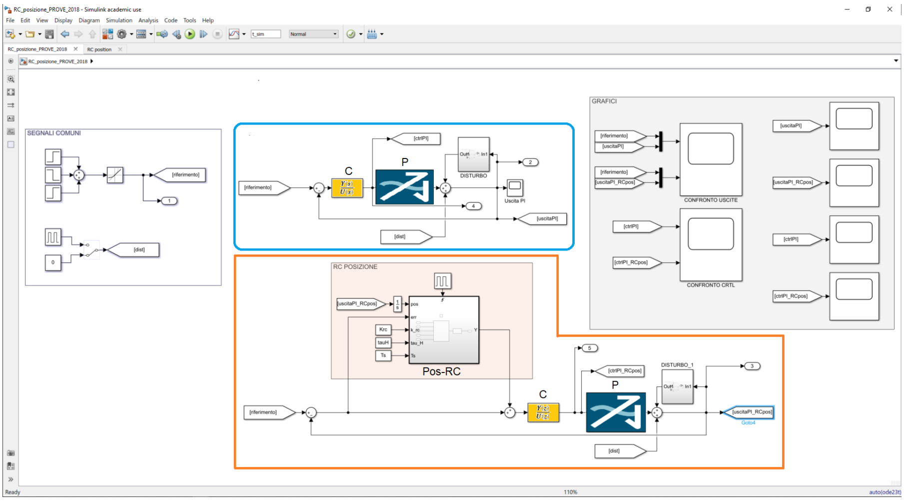


Figure 4.23: Simulink scheme used for MIL tests done in Simulink/Amesim co-simulation.

simulates a standard controller in feedback, while the sub-scheme in the lower part of the Figure 4.23 (orange area) simulates a standard controller aided by a repetitive controller in the position domain.

These co-simulations were carried out to understand if the repetitive controller in the position domain could give benefits on generic mechatronic systems modeled through the Amesim software.

Figure 4.24 shows the performance comparison between the output of the simulated system with Amesim using a single feedback controller (upper graph) and the output of the simulated system using a feedback controller aided by a repetitive controller in the position domain (lower graph). As can also be seen in the co-simulation, the effects of the repetitive controller in the position domain are significant, as shown by Figure 4.25, where the control errors are compared. In blue the error obtained using a single feedback controller, while in red the error obtained using the feedback controller supported by the repetitive controller in the position domain. In Figure

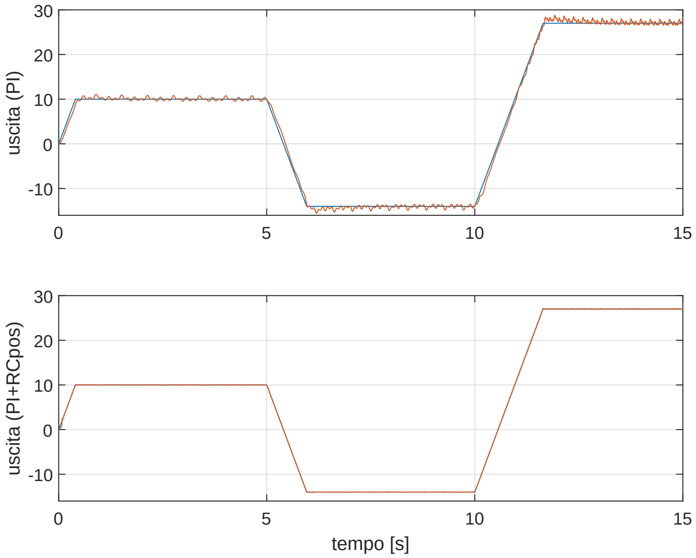


Figure 4.24: Output controlled with a PI controller (upper graph) and output controlled with a PI controller with the addition of the repetitive controller in position (lower graph). The reference in both cases is made to change specifically to have different periods of disturbance over time.

4.26 the respective control actions in the two cases are shown.

4.4.2 HIL validation

The proposed position repetitive control has finally been validated by implementing it on the HIL testbed shown in Figure 1.6. The parameters of both the system and the feedback controller are the same ones as in the case of the HIL validation of the RC method described in Chapter 4.2.2. The parameters of the RC position controller are as follows:

- gain of the repetitive controller in the domain of the position $K_r = 1.0$;

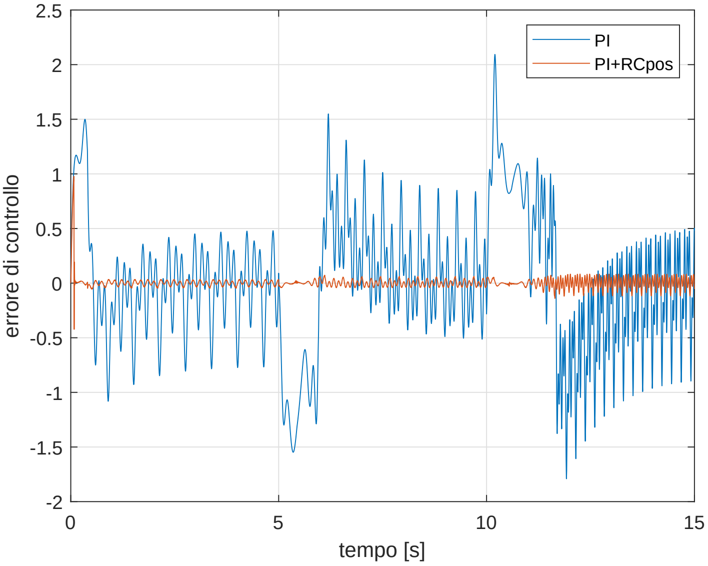


Figure 4.25: Comparison of the control error obtained with a PI controller (blue line) and the control error obtained with a PI controller and the addition of the repetitive controller in position (red line).

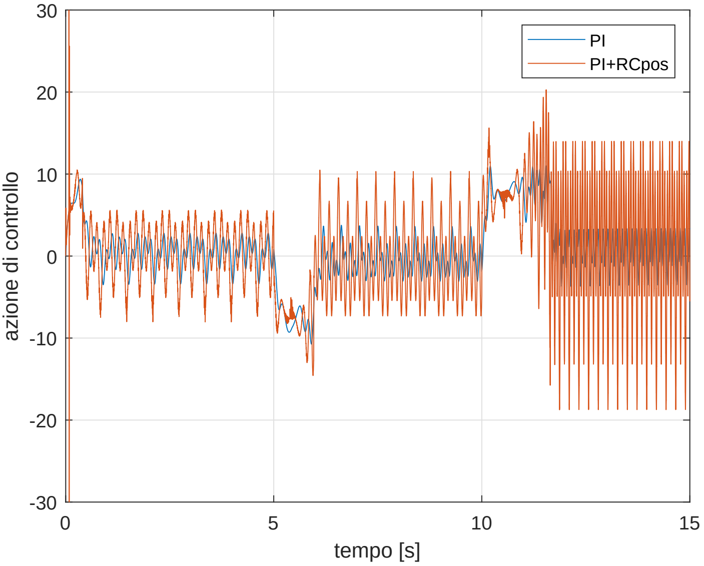


Figure 4.26: Comparison between the control action calculated with a PI controller (blue line) and the control action calculated with a PI controller and the addition of the repetitive controller in position (red line).

- cutoff frequency of the stability filter $\omega_c^f = 490$ [Hz];
- length of the memory vector $N_p = 500$;
- periodicity of position $P_d = 2\pi$ [rad].

The stability filter is the same one described in (4.36).

The introduced disturbance on the load side is defined by function (4.35) and the values for A_1, A_2 and A_3 have been choose as in Chapter 4.4.1.

The comparison between the tracking performance between with the closed-loop PI approach with and without the use of the RC position controller are shown in Figure 4.27. Due to the presence of external disturbances, the use of a PI controller alone cannot suppress the effect of disturbances, while the use of the RC position controller increase the performance at both the setpoint velocities. On the contrary, the RC position controller introduces an overshoot when the velocity reverence is changed. This overshoot can be avoided by turning off the RC controller during drastic changes in the velocity reference.

The improvement in performance is underlined by Figure 4.28, where the velocity error is shown.

On the other hand, while the improvement in terms of performance are evident, the inclusion of the RC controller in the control scheme entails a more aggressive approach, as shown by the comparison of the control actions of the two control schemes in Figure 4.29.

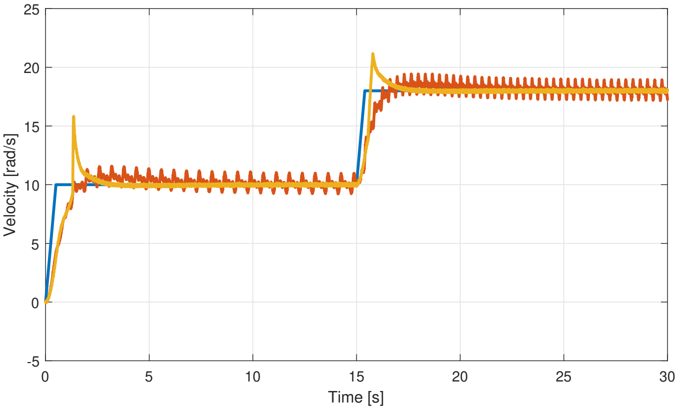


Figure 4.27: Comparison between PI control (orange line) and PI plus position RC control (yellow line). The reference is shown in blue line.

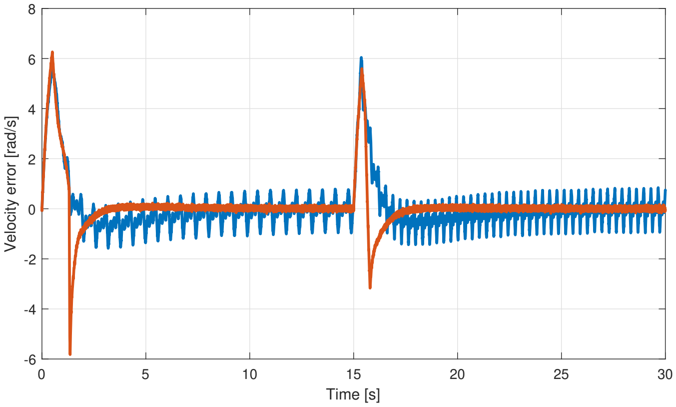


Figure 4.28: Velocity error with PI (blue line) and PI plus position RC control (orange line).

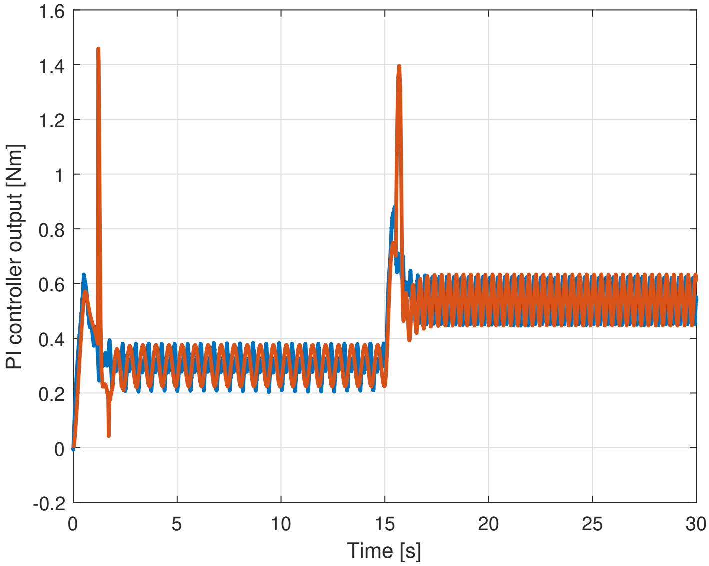


Figure 4.29: Control action with PI (blue line) and PI plus position RC control (orange line).

4.5 Conclusions and contributions

In conclusion, in this chapter various approaches to the control of mechatronic systems subject to repetitive disturbances have been analyzed. With a special focus on their implementability on industrial hardware, the techniques presented have firstly been implemented on Matlab/Simulink environment and tested. All the proposed techniques have been also implemented on a standard industrial drive (ADV200S series by Gefran S.p.a.) and validated on the HIL testbed described in Chapter 1.4.4, demonstrating their ability in compensating repetitive disturbance, both in time and position domain.

The reduced memory necessary for the implementation of the proposed techniques, which require storing vectors of only 200 elements, is motivated by the need of implementing the techniques directly into the drives, without the need for an external PLC. Therefore, the techniques can be implemented also in industrial hardware with reduced memory capability.

In addition, an innovative repetitive controller methodology was developed and tested in simulation and co-simulation in the domain of the position suitable for application in common industrial controllers (Section 4.4).

The tangible result of this chapter is a library, written in IEC 61131-3 Structured Text language, containing plug-and-play function blocks which implement the presented ILC, RC and position-based RC algorithms. As the function blocks are written with a IEC 61131-3 coding language, they are easily exported into virtually every industrial control system.

Conclusions

This thesis has presented practical solutions for the most common problems affecting industrial mechatronic systems.

An in-depth study of the available control approaches for highly underdamped mechatronic systems (such as industrial cranes) and the study of tailored control techniques in Chapter 2 has yield to the implementation and testing of various approaches. Both studies regarding the robustness and, in general, the performance of the single techniques as well as comparative studies have been performed, in order to outline a guideline for the choice of the correct approach for the control of a specific industrial application. In particular, a novel approach for the control of cranes which exhibit double-pendulum dynamics has been proposed, implemented and tested. This novel approach differs from input shaping and notch filtering approaches as it is characterized by only one tuning parameter, with a clear physical meaning (the total manoeuvre time τ), which can be directly set and changed by the operator depending on level of robustness required by the operator.

Different autotuning procedures have been devised in Chapter 3. Depending on the specific application, the devised autotuning approaches can cope with open or closed-loop identification phases and parameters tuning procedures are available both for velocity-driven and position-driven applications. The novelties of the proposed approaches are the explicit use of the system constraints (in terms of maximum torque, velocity and position) during the definition of the trajectory for the identification phase, and the automatic tuning of bi-quadratic filters for the compensation of oscillations generated by elastic transmissions between motor and load. These features allow an increase in the achievable bandwidth, an increase in the performance as oscillations are reduced, and the active control of the achievable constraints.

Solutions for the compensation of repetitive disturbances on mechatronic systems have been studied and implemented. In addition to well-known solutions for the control of repetitive disturbances in the frequency domain, a new approach for the suppression of disturbances that depend on the position of the system has been validated.

All the studied solutions have been tested and validated following the rapid prototyping paradigm, and even when the implementation of the available HIL testbed has not be achieved (as in the case of MPC approaches for the control of industrial cranes), the result has been an indicator of the specific limits of modern industrial hardware which should be pushed further in the next generations.

Moreover, the techniques have been implemented and collected in libraries coded with languages from the IEC 61131-3 standard, so that the solutions can be easily implemented over a wide range of industrial applications.

The approach of testing and implementing on standard industrial hardware both techniques already present in literature and new is online with the purpose of the I-MECH project, which is to bring novel solutions to industrial applications by bridging the gap between latest research results and industrial practice. The validation of the presented techniques on industrial pilots finally demonstrates the soundness of the approach for its application in standard industrial mechatronic systems.

Bibliography

[1] E. M. Abdel-Rahman, A. H. Nayfeh, and Z. N. Masoud. Dynamics and control of cranes: A review. *Journal of Vibration and control*, 9 (7):863–908, 2003. (Cited on pages 17 and 38.)

[2] H. S. Ahn, Y. Chen, and K. L. Moore. Iterative learning control: Brief survey and categorization. *Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007. (Cited on page 187.)

[3] J. D. Álvarez, L. J. Yebra, and M. Berenguel. Repetitive control of tubular heat exchangers. *Journal of Process Control*, 17(9):689–701, 2007. (Cited on page 186.)

[4] K. J. Åström and T. Hägglund. *Advanced PID Control.* ISA Press, Research Triangle Park, USA, 2006. (Cited on page 135.)

[5] S. Beineke, F. Schutte, H. Wertz, and H. Grotstollen. Comparison of parameter identification schemes for self-commissioning drive control of nonlinear two-mass systems. In *IEEE Industry Applications Society*, pages 493–500, New Orleans, LA, 1997. (Cited on page 136.)

[6] D. Blackburn, J. Lawrence, J. Danielson, W. Singhose, T. Kamoi, and A. Taura. Radial-motion assisted command shapers for nonlinear tower crane rotational slewing. *Control Engineering Practice*, 18(5):523–531, 2010. (Cited on page 38.)

[7] D. Blackburn, W. Singhose, J. Kitchen, V. Patrangenaru, J. Lawrence, T. Kamoi, and A. Taura. Command shaping for nonlinear crane dynamics. *Journal of Vibration and Control*, 16(4):477–501, 2010. (Cited on page 38.)

[8] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *control systems magazine*, 26(3):96–114, 2006. (Cited on pages 186 and 203.)

[9] J. F. Calvert and D. J. Gimpel. Method and apparatus for control of system output in response to system input, July 30 1957. US Patent 2,801,351. (Cited on page 43.)

[10] M. Calvini, M. Carpita, A. Formentini, and M. Marchesoni. PSO-based self-commissioning of electrical motor drives. *Transactions on Industrial Electronics*, 62(2):768–776, 2015. (Cited on page 137.)

[11] Z. Cao and G. F. Ledwich. Adaptive repetitive control to track variable periodic signals with fixed sampling rate. *IEEE/ASME transactions on mechatronics*, 7(3):378–384, 2002. (Cited on page 186.)

[12] M. Čech, A. J. Beltman, and K. Ozols. I-MECH, Smart System Integration for Mechatronic Applications. In *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA*, Zaragoza, E, 2019. (Cited on page 135.)

[13] M. Čech, A. J. Beltman, and K. Ozols. I-mech–smart system integration for mechatronic applications. In *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 843–850, 2019. (Cited on page 3.)

[14] K. K. Chew and M. Tomizuka. Digital control of repetitive errors in disk drive systems. In *American Control Conference*, pages 540–548. IEEE, 1989. (Cited on page 186.)

[15] L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling mpc of nonlinear systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 13(3-4):295–308, 2003. (Cited on page 120.)

[16] R. Costa-Castelló, J. Nebot, and R. Grinó. Demonstration of the internal model principle by digital repetitive control of an educational laboratory plant. *Transactions on Education*, 48(1):73–80, 2005. (Cited on pages 185 and 187.)

[17] G. Ellis and R. D. Lorenz. Resonant load control methods for industrial servo drives. In *Conference Record of the 2000 IEEE Industry Applications Conference. Thirty-Fifth IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy (Cat. No. 00CH37129)*, volume 3, pages 1438–1445, 2000. (Cited on page 137.)

[18] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35(7):1215–1241, 1999. (Cited on page 165.)

[19] B. A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976. (Cited on page 187.)

[20] S. Garrido, M. Abderrahim, A. Gimenez, R. Diez, and C. Balaguer. Anti-swinging input shaping control of an automatic construction crane. *IEEE Transactions on Automation Science and Engineering*, 5: 549–557, 2008. (Cited on page 38.)

[21] M. Giacomelli, M. Beschi, L. Simoni, and A. Visioli. A software tool to make primary school students aware of control systems. In *2020 IFAC World Congress*. (Cited on page 42.)

[22] M. Giacomelli, D. Colombo, L. Simoni, G. Finzi, and A. Visioli. A fast autotuning method for velocity control of mechatronic systems. *IFAC-PapersOnLine*, 51(4):208–213, 2018. (Cited on page 158.)

[23] M. Giacomelli, M. Faroni, D. Gorni, A. Marini, L. Simoni, and A. Visioli. Model predictive control for operator-in-the-loop overhead cranes. In *23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 589–596, 2018. (Cited on page 110.)

[24] M. Giacomelli, M. Faroni, D. Gorni, A. Marini, L. Simoni, and A. Visioli. MPC-PID control of operator-in-the-loop overhead cranes: A practical approach. In *7th International Conference on Systems and Control (ICSC)*, pages 321–326, 2018. (Cited on pages 115 and 123.)

[25] M. Giacomelli, F. Padula, L. Simoni, and A. Visioli. Simplified input-output inversion control of a double pendulum overhead crane for residual oscillations reduction. *Mechatronics*, 56:37–47, 2018. (Cited on pages 55 and 57.)

[26] M. Giacomelli, D. Colombo, G. Finzi, V. Šetka, L. Simoni, and A. Visioli. An autotuning procedure for motion control of oscillatory mechatronic systems. In *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 829–835, 2019. (Cited on page 158.)

[27] M. Giacomelli, D. Colombo, G. Finzi, V. Šetka, L. Simoni, and A. Visioli. A closed-loop automatic tuning method for velocity control of oscillatory mechatronic systems. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 3245–3250, 2019. (Cited on page 175.)

[28] H. Giberti, S. Cinquemani, and G. Legnani. A practical approach to the selection of the motor-reducer unit in electric drive systems. *Mechanics based design of structures and machines*, 39(3):303–319, 2011. (Cited on page 160.)

[29] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics.* Addison-Wesley, jun 2001. (Cited on page 22.)

[30] M. Goubej. Kalman filter based observer design for real-time frequency identification in motion control systems. In *Proceedings 20th International Conference on Process Control*, Strbske Pleso, SK, 2015. (Cited on page 136.)

[31] M. Goubej and M. Schlegel. Robust PID control of electrical drive with compliant load. In *Proceedings 19th IFAC World Congress*, pages 11781–11786, Cape Town, SA, 2014. (Cited on page 137.)

[32] M. Goubej, A. Krejčí, and M. Schlegel. Robust frequency identification of oscillatory electromechanical systems. In *2013 International Conference on Process Control (PC)*, pages 79–84, 2013. (Cited on pages 136, 150, 155 and 165.)

[33] R. D. Hanson and T. C. Tsao. Periodic sampling interval repetitive control and its application to variable spindle speed noncircular turning process. *J. Dyn. Sys., Meas., Control*, 122(3):560–566, 2000. (Cited on page 186.)

[34] G. Hillerström and K. Walgama. Repetitive control theory and applications-a survey. *IFAC Proceedings Volumes*, 29(1):1446–1451, 1996. (Cited on pages 186 and 187.)

[35] B. Houska, H. J. Ferreau, and M. Diehl. Acado toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011. (Cited on page 129.)

[36] C. Hsu and Y. Lai. Novel online optimal bandwidth search and autotuning techniques for servo motor drives. *Transactions on Industry Applications*, 53(4):3635–3642, 2017. (Cited on page 136.)

[37] J. S. Hu. Variable structure digital repetitive controller. In *American Control Conference*, pages 2686–2690. IEEE, 1992. (Cited on page 186.)

[38] Y. S. Kim, K. S. Hong, and S. K. Sul. Anti-sway control of container cranes: inclinometer, observer, and state feedback. *International Journal of Control, Automation, and Systems*, 2(4):435–449, 2004. (Cited on page 40.)

[39] J. Lawrence and D. Frakes. Human operator performance testing using an input-shaped bridge crane. *Journal of Dynamic Systems, Measurement, and Control*, 128:835, 2006. (Cited on page 17.)

[40] J. A. Ledin. Hardware-in-the-loop simulation. *Embedded Systems Programming*, 12:42–62, 1999. (Cited on page 12.)

[41] 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*, 36(5):641–657, 2000. (Cited on page 186.)

[42] D. Lewis, G. G. Parker, B. Driessen, and R. D. Robinett. Command shaping control of an operator-in-the-loop boom crane. In *American Control Conference*, volume 5, pages 2643–2647. IEEE, 1998. (Cited on page 39.)

[43] D. Lewis, G. Parker, B. Driessen, and R. D. Robinett. Comparison of command shaping controllers for suppressing payload sway in a rotary boom crane. In *IEEE International Conference on Control Applications*, volume 1, pages 719–724, 1999. (Cited on page 66.)

[44] R. W. Longman. Iterative learning control and repetitive control for engineering practice. *International journal of control*, 73(10):930–954, 2000. (Cited on page 187.)

[45] M. J. Maghsoudi, Z. Mohamed, S. Sudin, S. Buyamin, H. I. Jaafar, and S. M. Ahmad. An improved input shaping design for an efficient sway control of a nonlinear 3d overhead crane with friction. *Mechanical Systems and Signal Processing*, 92:364–378, 2017. (Cited on page 38.)

[46] K. L. Moore. *Iterative learning control for deterministic systems.* Springer Science & Business Media, 1993. (Cited on page 203.)

[47] K. L. Moore. Iterative learning control: An expository overview. In *Applied and computational control, signals, and circuits*, pages 151–214. Springer, 1999. (Cited on page 186.)

[48] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978. (Cited on pages 149 and 165.)

[49] H. M. Omar and A. H. Nayfeh. Anti-swing control of gantry and tower cranes using fuzzy and time-delayed feedback with friction compensation. *Shock and Vibration*, 12(2):73–89, 2005. (Cited on page 85.)

[50] L. F. Osborne, J. Brummond, R. Hart, M. Zarean, S. M. Conger, et al. Clarus: Concept of operations. Technical report, United States. Federal Highway Administration, 2005. (Cited on page 6.)

[51] F. Padula, A. Visioli, D. Facchinetti, and A. Saleri. A dynamic inversion approach for oscillation-free control of overhead cranes. In *IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–6, 2015. (Cited on pages 39 and 55.)

[52] G. G. Parker, B. Petterson, C. Dohrmann, and R. D. Robinett. Command shaping for residual vibration free crane maneuvers. In *American Control Conference*, volume 1, pages 934–938. IEEE, 1995. (Cited on page 39.)

[53] G. G. Parker, R. D. Robinett III, B. J. Driessen, and C. R. Dohrmann. Operator in-the-loop control of rotary cranes. In *Smart Structures and Materials 1996: Industrial and Commercial Applications of Smart Structures Technologies*, volume 2721, pages 364–372. International Society for Optics and Photonics, 1996. (Cited on page 39.)

[54] A. Piazzi and A. Visioli. Optimal dynamic-inversion-based control of an overhead crane. *IEE Proceedings-Control Theory and Applications*, 149(5):405–411, 2002. (Cited on page 39.)

[55] A. Piazzi and A. Visioli. Using stable input–output inversion for minimum-time feedforward constrained regulation of scalar systems. *Automatica*, 41(2):305–313, 2005. (Cited on page 39.)

[56] J. J. Potter and W. Singhose. Improving manual tracking of systems with oscillatory dynamics. *IEEE Transactions on Human-Machine Systems*, 43:46–52, 2013. (Cited on page 38.)

[57] L. Ramli, Z. Mohamed, A. M. Abdullahi, H. I. Jaafar, and I. M. Lazim. Control strategies for crane systems: A comprehensive review. *Mechanical Systems and Signal Processing*, 95:1–23, 2017. (Cited on pages 17, 38 and 86.)

[58] J. A. Rossiter. *Model-based predictive control: a practical approach.* CRC press, 2003. (Cited on page 97.)

[59] M. Schlegel and M. Goubej. Feature-based parametrization of input shaping filters with time delays. *IFAC Proceedings Volumes*, 43(2): 247–252, 2010. (Cited on page 38.)

[60] P. Schmidt and T. Rehm. Notch filter tuning for resonant frequency reduction in dual inertia systems. In *IEEE Industry Applications Conference*, volume 3, pages 1730–1734, 1999. (Cited on page 66.)

[61] O. Severa and M. Čech. Rex—rapid development tool for automation and robotics. In *8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 184–189, 2012. (Cited on page 13.)

[62] L. Simoni, M. Beschi, D. Colombo, A. Visioli, and R. Adamini. A hardware-in-the-loop setup for rapid control prototyping of mechatronic systems. In *20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4, 2015. (Cited on page 12.)

[63] N. Singer and W. Seering. Preshaping command inputs to reduce system vibration. *Journal of Dynamic Systems, Measurement, and Control*, 112(1):76–82, Mar. 1990. ISSN 1109-2777. (Cited on pages 38 and 43.)

[64] T. Singh and G. R. Heppler. Shaped input control of a system with multiple modes. *ASME Journal of Dynamic Systems, Measurement, and Control*, 115, 1993. (Cited on page 38.)

[65] W. Singhose, W. Seering, and N. Singer. Residual vibration reduction using vectors diagrams to generate shaped inputs. *Journal of Mechanical Design*, 116(2):654–659, 1994. (Cited on page 38.)

[66] W. Singhose, E. Crain, and W. Seering. Convolved and simultaneous two-mode input shapers. *IEE Proceedings - Control Theory and Applications*, 144(6):515–520, 1997. (Cited on page 38.)

[67] W. Singhose, L. J. Porter, T. D. Tuttle, and N. Singer. Vibration reduction using multi-hump input shapers. *ASME Journal of Dynamic Systems, Measurement, and Control*, 119:320–326, 1997. (Cited on page 38.)

[68] W. Singhose, W. Seering, and N. Singer. Time-optimal negative input shapers. *Journal of Dynamics, Systems, Measurement, and Control*, pages 198–205, 1997. (Cited on page 49.)

[69] O. Smith. Posicast control of damped oscillatory systems. *Proceedings of the IRE*, 45(9):1249–1255, 1957. (Cited on page 43.)

[70] M. Steinbuch. Repetitive control for systems with uncertain period-time, with application to a compact disc drive. *IFAC Proceedings Volumes*, 33(26):395–400, 2000. (Cited on page 186.)

[71] M. Steinbuch, S. Weiland, and T. Singh. Design of noise and period-time robust high-order repetitive control, with application to optical storage. *Automatica*, 43(12):2086–2095, 2007. (Cited on page 186.)

[72] M. Tomizuka. Dealing with periodic disturbances in controls of mechanical systems. *IFAC Proceedings Volumes*, 40(14):187–195, 2007. (Cited on pages 185 and 187.)

[73] T. C. Tsao, Y. X. Qian, and M. Nemani. Repetitive control for asymptotic tracking of periodic signals with an unknown period. *J. Dyn. Sys., Meas., Control*, 122(2):364–369, 2000. (Cited on page 186.)

[74] J. Vaughan, A. Yano, and W. Singhose. Comparison of robust input shapers. *Journal of Sound and Vibration*, 315:797–815, 2008. (Cited on pages 38, 47 and 48.)

[75] J. Vaughan, D. Kim, and W. Singhose. Control of tower cranes with double-pendulum payload dynamics. *Transactions on Control Systems Technology*, 18(6):1345–1358, 2010. (Cited on page 85.)

[76] S. Villwock and M. Pacas. Application of the welch-method for the identification of two-and three-mass-systems. *Transactions on Industrial Electronics*, 55(1):457–466, 2008. (Cited on pages 136 and 149.)

[77] Y. Wang, F. Gao, and F. J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19(10):1589–1600, 2009. (Cited on pages 185 and 187.)

[78] J. Weissbacher, E. Grunbacher, and M. Horn. Automatic tuning of a servo drive speed controller for industrial applications. In *IEEE International Conference on Mechatronics*, pages 700–705, Vicenza, I, 2013. (Cited on page 136.)

[79] S. Yang and K. Lin. Automatic control loop tuning for permanent-magnet ac servo motor drives. *Transactions on Industrial Electronics*, 63(3):1499–1506, 2016. (Cited on page 136.)

[80] G. Zhang and J. Furusho. Speed control of two-inertia system by PI/PID control. *Transactions on Industrial Electronics*, 41(3):603–609, 2000. (Cited on page 137.)