UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Head Office: Università degli Studi di Padova

Department of Management and Engineering

Ph.D. COURSE IN: MECHATRONICS AND PRODUCT INNOVATION ENGINEERING
CURRICULUM: MECHATRONICS
SERIES XXXV

MODEL-BASED METHODS FOR PRECISE TRAJECTORY TRACKING IN UNDERACTUATED MULTIBODY SYSTEMS

Coordinator: Ch.mo Prof. Daria Battini
Supervisor: Ch.mo Prof. Dario Richiedei

Ph.D. student: Jason Bettega

ACKNOWLEDGEMENTS

First of all, I want to hugely thank God for giving me the opportunity to take this path and for helping and guiding me along the entire Doctoral degree.

Secondly, I warmly thank my family because they always encouraged me during these years and also because they had a lot of patience in those moments where I did not have too much.

Additionally, a very big “thanks” goes to my supervisor, Prof. Dario Richiedei, who actually was (and he is) more than a supervisor: he is a friend. He allowed me to extend my knowledge in control theory, which is my greatest scientific interest, with a focus on underactuated systems, which represent a very challenging area; at the same time, he was able to understand me and “remain human” even in some difficult moments that occurred in my life during these years.

Furthermore, I want to deeply thank Prof. Alberto Trevisani for all the time that he spent (and he is spending) to allow me to go on with my academic career. In addition, I want to thank him also for all the interesting meeting that we had, regarding the “next research steps”, always allowing me to improve my knowledge in control theory in applied mechanics.

I want to thank all the members of our research group: Prof. Giovanni Boschetti, Prof. Paolo Boscariol and Eng. Aldo Dalla Via. It is always a pleasure to talk with them and the conversations with them always bring also some fun in the Lab.

Furthermore, I want to thank my PhD (and RTD-A) colleagues, Iacopo Tamellin, Riccardo Minto and Giulio Piva, for all the moments that we shared, in particular the most important ones: coffee breaks. It has always been great to share ideas with them, also talking of novel research topics in a very funny way.

Finally, the biggest acknowledgement goes to my father. He taught me a lot in my life. Unfortunately, he passed away on September of this year, so I want to mainly dedicate my thesis to him, which was a father, a friend, a master and a teacher.

ABSTRACT

Underactuation is a property that is characterizing more and more the mechanical and mechatronic systems nowadays, since this aspect can mainly occur in three different ways: intentionally, for example through a cost driven reduction of the number of actuators; accidentally, by the undesired failure of one or several motors; structurally, through the usage of lightweight components. Underactuated systems are therefore showing an increasing interest in the research field and, moreover, they represent a highly challenging area from the control design point of view. Indeed, depending on the desired system output, this kind of system is usually characterized by the presence of unstable internal dynamics which does not allow the exact computation of the model inversion and, as a consequence, the evaluation of the required feedforward control inputs. Moreover, having a number of degrees of freedom which is greater than the number of degrees of actuation, together with the presence of possible undershoot in the system response due to right half plane zeros in their linearized transfer functions, the control design of these systems results to be very challenging and therefore more sophisticated control algorithms have to be used.

Therefore, the goal of this thesis is to propose a set of tools that can be exploited to achieve good performances in the presence of underactuated multibody systems, both in terms of path and trajectory tracking. More precisely, the first part of this thesis focuses on the feedback control design, proposing a novel Model Predictive Control algorithm capable to ensure very low tracking errors by exploiting the dynamics of the desired trajectory. Remaining in the field of feedback control design, the second part shows how it is possible to ensure high performances in the presence of underactuated systems characterized by delay; this topic is important since the presence of delay in a closed-loop system can make the characteristic equation be characterized by latent roots, which can threaten the stability of the system. Therefore, a control algorithm is presented with the goal to achieve the desired performances with a good level of robustness. Finally, the third part focuses the attention on the feedforward control design, showing how model inversion can be accurately performed in the presence of underactuated systems.

All the methods proposed in this thesis are not strictly related to the test cases here presented, but they can be easily exploited to ensure good performances with the majority of the underactuated multibody systems that characterizes the industrial automation environment nowadays.

Keywords: Underactuated systems, nonminimum-phase systems, motion control, Model Predictive Control, feedforward control, motion planning

SOMMARIO

La sotto-attuazione è una proprietà che sta caratterizzando sempre di più i sistemi meccanici e meccatronici oggigiorno, dato che questo aspetto si presenta sostanzialmente in tre modi differenti: intenzionalmente, ad esempio attraverso una riduzione del numero di attuatori dettata da motivi economici; accidentalmente, tramite la rottura indesiderata di uno o più motori; strutturalmente, in seguito all'utilizzo di componenti dotati di flessibilità. I sistemi sotto-attuati stanno pertanto mostrando un interesse sempre più elevato nel mondo della ricercar e, in particolare, essi rappresentano un settore pieno di sfide dal punto di vista della progettazione degli algoritmi di controllo. Infatti, a seconda dell'uscita desiderata del Sistema, questa tipologia di sistemi è solitamente caratterizzata dalla presenza di una dinamica interna instabile che impedisce il calcolo esatto dell'inversione dinamica e, di conseguenza, la valutazione delle forze di controllo richieste. Inoltre, avendo un numero di gradi di libertà maggior del numero di gradi di attuazione, insieme alla possibile presenza di sotto-elongazione nella risposta del sistema data dalla presenza di zeri a parte reale positiva nelle loro funzioni di trasferimento linearizzate, la progettazione del controllo di questi sistemi risulta essere particolarmente difficoltosa e pertanto devono essere utilizzati algoritmi di controllo più sofisticati.

Pertanto, l'obiettivo di questa tesi è quella di proporre un insieme di strumenti che possa essere sfruttato al fine di ottenere delle performance di alto livello in presenza di sistemi sotto-attuati, in termini di inseguimento di una traiettoria sia spaziale sia temporale. Più precisamente, la prima parte di questa tesi si focalizza sulla progettazione di algoritmi di controllo in retroazione, proponendo un nuovo algoritmo di controllo predittivo "model-based" in grado di assicurare errori di inseguimento bassi grazie alla considerazione della dinamica caratterizzante la traiettoria desiderata stessa. Rimanendo nel campo degli algoritmi di controllo in retroazione, la seconda parte mostra com'è possibile assicurare buone performance in presenza di sistemi sotto-attuati affetti da ritardo; questo argomento è molto importante dato che la presenza di ritardo in un sistema a catena chiusa può portare alla presenza di radici trascendentali all'interno dell'equazione caratteristica del sistema, minacciando di conseguenza la stabilità dello stesso. Pertanto, è proposto un algoritmo di controllo con l'obiettivo di ottenere le perfomance dinamiche desiderate, assicurando al contempo un buon livello di robustezza. Infine, la terza parte di questa tesi si focalizza sulla progettazione di algoritmi di feedforward, mostrando quindi com'è possibile svolgere il calcolo di inversione del modello in maniera accurata in presenza di sistema sotto-attuati.

Tutti i metodi proposti in questa tesi non sono relegati solamente agli esempi applicativi qui riportati, ma al contrario possono essere facilmente sfruttati per assicurare buone performance con la maggior parte dei sistemi sotto-attuati caratterizzanti l'ambiente industriale odierno.

Parole chiave: Sistemi sotto-attuati, Sistemi a non-minima fase, Controllo del moto, Controllo Predittivo "Model-Based", Controllo di feedforward, Pianificazione del moto

TABLE OF CONTENTS

1. Introduction 9
1.1 Background and motivation 9
1.2 Aim and structure of the work 10
2. Trajectory Tracking in Underactuated Multibody Systems through Model Predictive Control with Embedded Reference Dynamics 12
2.1. Introduction 12
2.1.1. Motivations and state-of-the-art 12
2.1.2. Contributions of the Chapter 13
2.2. Theoretical concepts 13
2.2.1. Underactuated system modeling 13
2.2.2. MPC with embedded integrator 15
2.2.3. Prediction phase in standard MPC with embedded integrator 15
2.2.4. Definition of the cost function for standard MPC 16
2.2.5. Autonomous state-space model for MPC with Embedded Reference Dynamics 17
2.2.6. Definition of constraints for the MPC-ERD with embedded integrator 20
2.3. Test case description and system model 21
2.4. Numerical validation 25
2.4.1. Simulation results 25
2.4.2. Robustness analysis 30
2.4.3. Computational effort 32
2.5. Experimental validation 32
2.6. Conclusions 35
3. Vibration Control of a Two-Link Flexible Robot Arm with Time Delay through the Robust Receptance Method 37
3.1. Introduction 37
3.2. Definitions 38
3.3. Method description 39
3.3.1. Placement of the Np closed-loop poles 39
3.3.2. Introduction of the robustness condition 40
3.3.3. Numerical implementations details 41
3.4. Test case: a two-link flexible robot arm 42
3.4.1. System description 42
3.4.2. Numerical assessment 43
3.4.3. Experimental application 46
3.5. Conclusions 52
4. Collocated Model Predictive Control with Embedded Integrator for Path Tracking in Cable Suspended Parallel Robots with Flexible Cables 53
4.1. Introduction 53
4.1.1. State-of-the-art and Chapter motivations 53
4.1.2. Chapter contributions 54
4.2. Dynamic model 55
4.2.1. Studied system 55
4.2.2. Dynamic model of the CSPR with stiff cables 55
4.2.3. Dynamic model of the CSPR with axially-flexible cables 56
4.3. Design of the control scheme 58
4.3.1. Underlying idea of the control scheme 58
4.3.2. Control architecture 58
4.3.3. Design of the MPC 59

4.3.4. Computation of the motor torques 62
4.4. Numerical results 63
4.4.1. System description 63
4.4.2. Test cases 64
4.5. Conclusions 75
5. Precise Path and Trajectory Tracking in Cable Driven Parallel Robots through Model Predictive Control with Embedded Reference Dynamics 77
5.1. Introduction 77
5.1.1. Motivations and state-of-the-art 77
5.1.2. Contributions of this Chapter 77
5.2. System model 77
5.3. The control scheme 77
5.3.1. Control overview 77
5.3.2. The state-space model with the difference variables 77
5.3.3. Model for the prediction 77
5.3.4. Solution of a standard MPC problem 77
5.3.5. Embedding of the reference dynamics 77
5.3.6. Autonomous state-space model of the reference trajectory 77
5.3.7. Introduction of constraints 77
5.3.8. Solution of the constrained MPC problem 78
5.3.9. Computation of the motor torques and currents 78
5.4. Numerical validation 78
5.4.1. General description 78
5.4.2. Controller stability margins 78
5.4.3. Test case 1: trajectory tracking with a step reference 78
5.4.4 Test case 2: trajectory tracking with planar circumference 78
5.4.5. Test case 3: trajectory tracking of a spatial spiral 78
5.5. Conclusions 78
6. Stable Inverse Dynamics for Feedforward Control of Nonminimum-phase Underactuated Systems 79
6.1. Introduction 79
6.1.1. Motivations and state-of-the-art 79
6.1.2. Contributions of this Chapter 80
6.2. Method description 81
6.2.1. Definitions 81
6.2.2. Coordinate partitioning 82
6.2.3. Approximation of the internal dynamics through linearly combined output 83
6.2.4. Solution of the algebraic equation for the actuated coordinates 84
6.3. Test case 1: flexible beam with lumped masses 85
6.3.1. Description and system model 85
6.3.2. Stabilization of the internal dynamics 87
6.3.3. Numerical results 89
6.4. Test case 2: overhead cartesian crane 91
6.4.1. Description and system model 91
6.4.2. Stabilization of the internal dynamics 93
6.4.3. Numerical results 94
6.5. Test case 3: underactuated robotic arm with flexible passive joint 98
6.5.1. Description and system model 98
6.5.2. Stabilization of the internal dynamics 100
6.5.3. Numerical application 102

6.5.4. Robustness considerations against model uncertainties 106
6.5.5. Experimental application 108
6.6. Conclusions 109
7. Model Inversion for Precise Path and Trajectory Tracking in a Underactuated, Nonminimum-phase, spatial overhead crane 111
7.1. Introduction 111
7.1.1. Motivations and state-of-the-art 111
7.1.2. Contributions of the Chapter 112
7.2. System model and method description 113
7.2.1. System dynamic model 113
7.2.2. Formulation and stabilization of the internal dynamics 114
7.2.3. Computation of the command values of the actuated coordinates 115
7.2.4. Some comments on the required command forces 116
7.3. Experimental application 116
7.3.1. Setup description 116
7.3.2. Test case 1: planar circular path with 8 s motion time 118
7.3.3. Test case 2: planar circular path with 4 s motion time – experimental results 122
7.3.4. Test case 3: planar rhombus path – experimental results 125
7.3.5. Test case 4: planar spiral path – experimental results 129
7.4. Conclusions 131
8. Conclusion 133
BIBLIOGRAPHY 135

1. Introduction

1.1 Background and motivation

Energy saving is becoming an ever more important topic both in industrial and academic environments. Usually, robot manufacturers try to achieve this goal by lowering the mass of the components, however this aspect inevitably leads to the rise of flexibility issues and therefore vibrational phenomena start becoming more relevant. The main consequence in relationship to the presence of mechanical vibrations is represented by the degradation of the accuracy during the execution of the tasks, leading to final products with lower quality or, in the worst case, to fatal failures during the working processes themselves. These issues represent a huge loss for the industries in terms of money since the quality, and also the quantity, of the final products would not be as the one expected; therefore, a proper control design has to be carried on with the focus to take into account the presence of flexibility, and more in general the underactuation property. If this aspect is not considered, inefficient mechatronic systems would be achieved since the flexible terms would be always treated as external disturbances that the controller should mitigate without any knowledge on them, leading to robustness issues and a general increase of the energy demand. This latter is completely in contrast with the initial statement regarding the energy saving goal, therefore more sophisticated control algorithms must be designed in order to take into account the underactuation property ensuring, at the same time, good performances during the execution of the tasks of the robots as well as low energy consumption. With this objective in mind, a high-fidelity model is needed to properly describe the dynamics of the studied system and consequently exploiting this knowledge for the controller design procedure; due to this reason, the theory concerning multibody systems is exploited.

The core of multibody system dynamics concerns the modeling of mechanical, and in general mechatronic, systems that are subjected to large nonlinear motion. This kind of systems can be divided into two main categories: fully-actuated multibody systems and underactuated ones. The first group is characterized by a number of actuators which is equal to the number of degrees of freedom, while the second one includes all the systems that possess a number of actuators which is lower than the number of independent generalized coordinates. Underactuated multibody systems therefore represent a challenging type of multibody systems from the control design point of view and, consequently, they are covering more and more relevance for the industrial applications. Underactuation can mainly occur in three main ways. Firstly, a weight or cost driven methodology can reduce the number of actuators and therefore leading to an underactuated system in an intentional way. Secondly, one or several actuators can fail during operation, achieving underactuation in an accidentally way; in both cases one or more actuated joints are substituted with passive ones. Finally, underactuation might occur due to the presence of flexibility, and therefore in a structural way. As briefly mentioned before, this last aspect is more and more common nowadays due to the innovative light-weight design of robots in order to reach energy saving objectives; indeed, productivity volumes are growing at a very fast pace and, as a consequence, industrial environments are becoming more and more interested in light-weight manipulators since they show a high payload to self-weight ratio. This last aspect is very attractive because it allows to reduce the energy demand during the execution of the task. Even though this reduction could be relatively small for a singular operation, tasks are usually performed in a repetitive manner, with high operating frequencies, in the industrial automation processes, leading to a significant decrement of the total energy demand. However, as previously stated, the presence of tasks characterized by fast dynamics makes the flexibility property no longer negligible since undesired structural vibrations can rise and, therefore, its contribution must be properly modelled and included into the control design process.

From the control design point of view, underactuated multibody systems represent a highly challenging topic that is covering more and more interest in the research field. More precisely,

different solutions have been recently proposed in the field of feedback control, ranging from the more common PID controllers to more sophisticated control algorithms, such as Sliding Mode Controllers, Artificial Neural Networks and Model Predictive Controllers. Similarly, considering the field of feedforward control design, several solutions have been proposed in the recent years, offering algorithms for the computation of model inversion that in some cases lead to causal solutions and in other cases lead to non-causal ones. Typical examples can range from algorithms that exploit the zeros of the linearized transfer functions, such as the NPZ-Ignore technique, to the more recent ones based on the flatness property of a system, such as the Byrnes-Isidori algorithm.

1.2 Aim and structure of the work

The aim of this work is to offer a set of tools for the control of underactuated multibody systems, extending the current state-of-the-art with novel approaches and formulations, both in terms of feedback control and feedforward one.
More precisely, in Chapter 2 a scheme is proposed for precise tip trajectory tracking of time-varying references in underactuated, nonminimum-phase multibody systems, that exploits a novel formulation of Model Predictive Control (MPC). The proposed approach embeds an autonomous state-space model of the desired reference, with the goal to ensure low tracking errors. The controller is therefore named MPC with Embedded Reference Dynamics, MPC-ERD. The effectiveness of the proposed control algorithm is validated experimentally through a two-link underactuated planar manipulator, and the results are compared with those provided by standard MPC with embedded integrator.
The same underactuated system of Chapter 2 is adopted in Chapter 3 as test case for a different control paradigm, that is Pole Placement. This control technique is commonly used for explicitly imposing natural frequencies and damping of the vibrational modes of linear systems. In particular, a method for active vibration control of a two-link flexible robot arm in the presence of time delay is proposed in Chapter 3, by means of robust pole placement. The controller design is performed through the receptance method and the desired stability margin is achieved by solving an optimization problem based on the Nyquist stability criterion. The method is applied on a laboratory testbed that mimic a typical flexible robotic system employed for pick-and-place applications and the nonlinear terms, due to the finite displacements, uncertainty, disturbances, and the coarse encoder quantization, are effectively handled by embedding the robustness requirement into the design.
In Chapter 4, a Model Predictive Control algorithm, with embedded integrator, is considered for path tracking purposes in the presence of a Cable Suspended Parallel Robot characterized by cables whose elasticity is not negligible. To handle model nonlinearities while ensuring small computational effort, a controller made by two sequential control actions is developed, as the one presented in the previous chapter. The first term is a position-dependent, collocated Model Predictive Control (MPC) with Embedded Integrator (EI), to compute the optimal cable tensions, while the second control term transforms the optimal tensions into the commanded motor torques, and hence currents. Control design is performed through the robot dynamics model, formulated with the assumption of rigid cables, with the goal to assess the robustness of the controller in the presence of highly flexible cables, whose flexibility is not taken into account in the controller design. Reduced excitation of the load oscillations is achieved by penalizing large tension variations in the cost function adopted in the controller design, plus some hard constraints on the maximum tension derivatives. Additionally, a simplified dynamic model of the electrical dynamics and the sensor quantization are included to provide a realistic representation of the real environments and, finally, the results are compared with the ones coming from a benchmark.
Since the outcomes of Chapter 4 highlights that Model Predictive Controllers represent a very good candidate for achieving high performances in Cable Driven Parallel Robots, also in the presence of highly flexible cables, a Cable Suspended Parallel Robot is still considered in Chapter 5, but in this case with rigid cables. Indeed, it is useful to highlight that, when the number of independent cables

is equal to the number of degrees of freedom of the end-effector, a fully-actuated system is present until cable tensions are greater than (or even equal to) zero. But when this condition fails, the system under investigation results to be an underactuated one. With the goal to avoid this undesired condition, Chapter 5 exploits the proposed MPC-ERD with the goal to ensure precise path and trajectory tracking of time-varying references in Cable Driven Parallel Robots (CDPRs). This problem is not straightforward since this kind of systems is usually highly nonlinear, and the positivity of cable tensions must be ensured since cables can only pull and not push. To face the highly nonlinearity of this kind of systems, a two-stage controller is proposed: the first stage exploits the MPC-ERD algorithm, while the second one consider the dynamic model of each actuator. A further novelty of is that the operating limits of the motors in terms of torque and speed, described by their characteristic curve, are included as well into the controller design, acting as time-varying constraints for the cable tensions. The effectiveness of proposed controller is validated numerically through a Cable Suspended Parallel Robot (CSPR) with three degrees of freedom, and the results are compared with those provided by both standard MPC with embedded integrator and classic MPC without embedded integrator.

Unless advanced controllers, such as the proposed MPC-ERD, are considered, feedforward control must be taken into account with the goal to further improve the performance in terms of path and trajectory tracking. The solution of inverse dynamics problem is usually trivial for fully-actuated systems but, unfortunately, it is not straightforward for underactuated ones. Therefore, Chapter 6 proposes an enhanced inverse dynamics approach for feedforward control of underactuated multibody systems and, more precisely, the method exploits the concept of partitioning the independent coordinates into actuated and unactuated ones, to achieve the internal dynamics of the system. After stabilization of the internal dynamics, the exact algebraic model of the actuated sub-system is inverted, by leading to the desired control forces with just little approximations.

Since industrial robots are characterized by co-located controllers and, where there is no direct measurement regarding the load, the only way to achieve good performances in terms of path and trajectory tracking of the load is by implementing a proper motion planning. With this focus, the idea proposed in Chapter 6 is interpreted in a new way and, therefore, it is exploited in Chapter 7, proposing a motion planning technique for precise path and trajectory tracking in an underactuated, nonminimum-phase, spatial overhead crane. Besides having a number of independent actuators that is smaller than the number of degrees of freedom, tip control on this system is characterized by an unstable internal dynamics that leads to divergent solution of the inverse dynamic problem. To overcome this issue, output redefinition is adopted within the internal dynamics to stabilize it, leading to stable and causal reference commands for the platform translations. Comparison with the state-of-the-art input shaping is also proposed.

2. Trajectory Tracking in Underactuated Multibody Systems through Model Predictive Control with Embedded Reference Dynamics

This Chapter proposes a scheme for precise tip trajectory tracking of time-varying references in underactuated, nonminimum-phase multibody systems, that exploits a novel formulation of Model Predictive Control (MPC). This problem is not straightforward, due to the properties of this kind of multibody systems and to the features of traditional MPC algorithms. Indeed, MPC is intrinsically devoted to regulation or to step tracking. To overcome these issues, the proposed approach embeds an autonomous state-space model of the time-varying reference into the constrained optimization process of MPC. The controller is therefore named MPC with Embedded Reference Dynamics, MPC-ERD. By embedding the reference in the controller design, no feedforward control is required, thus overcoming the difficulties in inverting the dynamic model of a nonminimum-phase underactuated systems.
The effectiveness of MPC-ERD is validated experimentally through a two-link underactuated planar manipulator, and the results are compared with those provided by standard MPC with embedded integrator. Besides inheriting the benefits of standard MPC, MPC-ERD ensures precise tracking of the tip time-varying reference, with negligible delay and error.

2.1. Introduction

2.1.1. Motivations and state-of-the-art

Underactuated multibody systems, i.e., with fewer independent control inputs than the number of degrees of freedom, are relevant in mechatronics and robotics and several examples can be found in the literature and in industries: cranes, manipulators with passive joints, flexible joint or flexible links are some meaningful examples. Due to the presence of flexible dynamics to be controlled through just few control forces, precise motion control of underactuated multibody systems is a challenging task especially when they operate with high speeds and accelerations. For these reasons, in the last decades a lot of effort has been spending to find suitable feedforward and feedback controllers to perform effective position control. The difficulties are exacerbated when the goal is ensuring precise trajectory tracking of the end-effector position; this kind of control problems is often known as tip tracking control. Tip tracking control of underactuated flexible multibody systems has, indeed, an additional difficulty in the nonminimum-phase property due to the presence of positive real zeros in the transfer function from the input force to tip position, that cause some unwanted behaviours. First, if the system features an odd number of real right-half plane zeros [1], undershoot is experienced by the tip position in tracking the reference trajectory. Moreover, if a reference trajectory-in-time is specified for the end-effector in lieu of a simple step reference, tracking delay and errors become relevant. Secondly, such zeros, additionally, make difficult computing exact feedforward forces and, for this reason, several techniques have been recently proposed to overcome these limitations, allowing to evaluate the required feedforward inputs that are capable to achieve the desired tip displacement [2–5]. Unfortunately, feedforward techniques are usually not enough to ensure good tracking performances due to unmodeled dynamics, model parameter variations and external disturbances. As a consequence, feedforward techniques are usually implemented together with feedback ones to achieve more robust controlled systems. On the other hand, feedback control alone is often not satisfactory for tracking time-varying reference trajectory, and therefore most of the approaches usually adopted to regulation are not always suitable. State-of-the-art control techniques therefore relies on feedforward terms, sometimes computed through approximate inversion [6], and (or together with) composite feedback control laws or multi-loop schemes (see e.g. [6–12]).

2.1.2. Contributions of the Chapter

In this Chapter, precise tip trajectory control in underactuated multibody systems is solved by exploiting and extending the idea of Model Predictive Control (MPC). Indeed, among the model-based control approaches, MPC has several features that make it very attractive for underactuated multibody systems. The basic idea of MPC is to solve an optimal control problem defined by a cost function over a receding horizon and constrained by the system dynamics and by bounds on some variables. The optimization problem, over the future control variables, is solved at each time step by predicting the future states and outputs of the system. Hence, MPC provides an optimal sequence of the control input, in accordance with some metrics. A great advantage is that it explicitly embeds constraints on input and/or output variables, which is instead difficult to include in most of the classical controllers.

MPC was originally applied to control power plants and chemical processes [13], and several industrial applications can be found too [14]. In the field of motion control of mechanisms and manipulators, MPC has been recently proved to be effective in regulation problems [15,16,25,26,17–24], where one or more system output positions are controlled to become null or to assume prescribed values in a finite time, after a step change of the reference, without imposing the transition from the initial to the final state. In these papers, it is clearly demonstrated the capability of MPC schemes to ensure precise, and sometimes fast too, achievement of the desired position, while damping vibrations when required. Moreover, thanks to the interesting feature of including constraints into the controller design, recently MPC algorithms have also been exploited in the presence of fully-actuated cable robots [27,28] to handle the requirement of positive cable tensions, since cables can only pull and not push.

On the other hand, precise and zero-delay tracking of time-varying position references is not straightforward with MPC, and hence it has been marginally addressed in the literature. Indeed, in the standard MPC formulation, the reference signal is kept constant during the prediction horizon, thus leading to a piecewise-constant approximation of a time varying reference. Clearly, these difficulties are exacerbated in the case of underactuated systems, and even more in tip control where unstable internal dynamics should be faced.

A novel formulation is proposed in this work and experimentally validated through a laboratory prototype of a two-link underactuated system driven by just one actuator. The idea of the proposed approach is to embed the model of the time-varying reference of the tip into the optimization process, performed over the prediction horizon, through an autonomous state-space model. The controller is therefore named MPC-ERD, MPC with embedded reference dynamics. By embedding the dynamics of both the reference and of the system to be controlled, the proposed approach is effective even without using feedforward terms, thus avoiding the difficulties in model inversion of nonminimum-phase systems. To achieve the autonomous state-space representation of the reference, to be embedded in the optimization problem, this Chapter exploits the theory of Dynamic Mode Decomposition (DMD) [29], leading to a control algorithm formulation which is capable to take into account the time-varying nature of the reference directly into the cost function that will be minimized. The concept of changing the predicted reference during the prediction horizon has never been considered so far and, for this reason, it represents a novel contribution of this work. The use of this powerful numerical tool is another novel contribution of this Chapter.

2.2. Theoretical concepts

2.2.1. Underactuated system modeling

Let us consider a multibody system modelled through its n independent coordinates $\boldsymbol{\theta} \in \mathbb{R}^n$ (n is the number of degrees of freedom, DOFs) as follows:

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}(t)+\mathbf{D}\dot{\boldsymbol{\theta}}(t)+\mathbf{K}\boldsymbol{\theta}(t)+\mathbf{d}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})+\mathbf{g}(\boldsymbol{\theta})=\mathbf{P}\mathbf{u}(t) \tag{1}$$

where $\mathbf{M}(\boldsymbol{\theta})$, $\mathbf{D}$, $\mathbf{K}$ $\in \mathbb{R}^{n\times n}$ are the mass, damping and stiffness matrices, respectively, $\mathbf{d}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \in \mathbb{R}^n$ is the vector of gyroscopic and centrifugal terms, $\mathbf{g}(\boldsymbol{\theta}) \in \mathbb{R}^n$ is the vector containing the gravity terms, $\mathbf{u}(t) \in \mathbb{R}^{n_{in}}$ is the control force vector, $\mathbf{P} \in \mathbb{R}^{n\times n_{in}}$ is the control force distribution matrix (n_{in} is the number of independent control inputs). If $n_{in} < n$, the system is said to be underactuated. In this case the equations of motion cannot be always inverted through algebraic manipulation to exactly compute the control forces that leads to the desired state, or output, trajectory [4]. Additionally, feedback control is more challenging as well.

To face all the discussed negative aspects and to ensure a precise and fast trajectory tracking of the end-effector, this Chapter exploits a new and enhanced formulation of Model Predictive Control (MPC). The method is developed through the linearized model, i.e., by assuming small displacements about the equilibrium configuration. Although the dynamics of flexible multibody systems are nonlinear, the use of linearized models and linear controller has been widely proposed in the literature and it has been often proved that it can be effective for finite displacement too [30]. Deviations from linearity are considered as uncertainties that the controller should be able to face. By linearizing the model around its stable equilibrium point, the following model is obtained:

$$\mathbf{M}_\mathbf{e}\ddot{\boldsymbol{\theta}}(t)+\mathbf{D}_\mathbf{e}\dot{\boldsymbol{\theta}}(t)+\mathbf{K}_\mathbf{e}\boldsymbol{\theta}(t)=\mathbf{P}\mathbf{u}(t) \tag{2}$$

where $\mathbf{M}_\mathbf{e}$, $\mathbf{D}_\mathbf{e}$, $\mathbf{K}_\mathbf{e}$ $\in \mathbb{R}^{n\times n}$ arise from the Taylor's expansion of Eq.(1). It should be noted that the contributions of gravitational forces are included within $\mathbf{K}_\mathbf{e}$.

A first-order, state-space representation is required to develop the MPC, by introducing the *2n*-dimensional state vector $\mathbf{x}_\mathbf{c}(t)=\left[\boldsymbol{\theta}(t) \quad \dot{\boldsymbol{\theta}}(t)\right]^T$:

$$\begin{cases}\mathbf{x}_\mathbf{c}(t)=\mathbf{A}_\mathbf{c}\,\mathbf{x}_\mathbf{c}(t)+\mathbf{B}_\mathbf{c}\,\mathbf{u}(t)\\ \mathbf{y}_\mathbf{c}(t)=\mathbf{C}_\mathbf{c}\,\mathbf{x}_\mathbf{c}(t)\end{cases} \tag{3}$$

where $\mathbf{y}_\mathbf{c}(t) \in \mathbb{R}^{n_{out}}$ represents the vector of the system output, and the state-space matrices $\mathbf{A}_\mathbf{c} \in \mathbb{R}^{2n\times 2n}$ and $\mathbf{B}_\mathbf{c} \in \mathbb{R}^{2n\times n_{in}}$ have the following definitions:

$$\mathbf{A}_\mathbf{c}=\begin{bmatrix}\mathbf{0}_n & \mathbf{I}_n\\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C}\end{bmatrix} \quad \mathbf{B}_\mathbf{c}=\begin{bmatrix}\mathbf{0}_n\\ \mathbf{M}^{-1}\mathbf{P}\end{bmatrix} \tag{4}$$

$\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes a *n*-dimensional identity matrix (the same notation will be used for identity matrices of arbitrary dimensions). The output matrix $\mathbf{C}_\mathbf{c} \in \mathbb{R}^{n_{out}\times 2n}$ is obtained by assuming n_{out} entries representing the position of the end-effector as the output.

Following the usual theoretical frame of MPC [31], the discrete version of the state-space model of Eq. (3) is adopted:

$$\begin{cases}\mathbf{x}_\mathbf{d}(k+1)=\mathbf{A}_\mathbf{d}\,\mathbf{x}_\mathbf{d}(k)+\mathbf{B}_\mathbf{d}\,\mathbf{u}(k)\\ \mathbf{y}_\mathbf{d}(k)=\mathbf{C}_\mathbf{d}\,\mathbf{x}_\mathbf{d}(k)\end{cases} \tag{5}$$

where k is the generic time instant and $\mathbf{A}_\mathbf{d} \in \mathbb{R}^{2n\times 2n}$, $\mathbf{B}_\mathbf{d} \in \mathbb{R}^{2n\times n_{in}}$, $\mathbf{C}_\mathbf{d} \in \mathbb{R}^{n_{out}\times 2n}$ are the discrete version of the continuous counterparts in Eq.(3).

2.2.2. MPC with embedded integrator

In this work, to improve tracking performances, an embedded integrator is introduced by exploiting the difference variables [32]:

$$\Delta\mathbf{x}_\mathbf{d}(k) = \mathbf{x}_\mathbf{d}(k) - \mathbf{x}_\mathbf{d}(k-1) \tag{6}$$

$$\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1) \tag{7}$$

$$\mathbf{y}_\mathbf{d}(k+1) - \mathbf{y}_\mathbf{d}(k) = \mathbf{C}_\mathbf{d}\,\Delta\mathbf{x}_\mathbf{d}(k+1) \tag{8}$$

By defining a new augmented state vector $\mathbf{x}(k) \in \mathbb{R}^{n_{aug}}$, where $n_{aug} = 2n + n_{out}$:

$$\mathbf{x}(k) = \begin{bmatrix} \Delta\mathbf{x}_\mathbf{d}(k) \\ \mathbf{y}_\mathbf{d}(k) \end{bmatrix} \tag{9}$$

the augmented state-space can be inferred:

$$\begin{cases} \begin{bmatrix} \Delta\mathbf{x}_\mathbf{d}(k+1) \\ \mathbf{y}_\mathbf{d}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_\mathbf{d} & \mathbf{0}_{2n\times n_{out}} \\ \mathbf{C}_\mathbf{d}\,\mathbf{A}_\mathbf{d} & \mathbf{I}_{n_{out}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_\mathbf{d}(k) \\ \mathbf{y}_\mathbf{d}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_\mathbf{d} \\ \mathbf{C}_\mathbf{d}\,\mathbf{B}_\mathbf{d} \end{bmatrix} \Delta\mathbf{u}(k) \\ \mathbf{y}(k) = \begin{bmatrix} \mathbf{0}_{n_{out}\times 2n} & \mathbf{I}_{n_{out}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_\mathbf{d}(k) \\ \mathbf{y}_\mathbf{d}(k) \end{bmatrix} \end{cases} \tag{10}$$

In a more compact form, Eq.(10) becomes:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\Delta\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) \end{cases} \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{n_{aug}\times n_{aug}}$, $\mathbf{B} \in \mathbb{R}^{n_{aug}\times n_{in}}$ and $\mathbf{C} \in \mathbb{R}^{n_{out}\times n_{aug}}$.

2.2.3. Prediction phase in standard MPC with embedded integrator

In order to understand how the prediction phase works, two relevant parameters have to be introduced: the prediction horizon $N_p \in \mathbb{R}$ and the control horizon $N_c \in \mathbb{R}$, with $N_c \leq N_p$. The former is the number of samples during which the prediction will be considered, and the optimization process will be performed. The latter, on the other hand, is the number of samples in which the optimization outcome will be spread.
Let us consider an arbitrary time step k ($k > 0$); the future time-history of the control forces is written in the difference form of Eq.(7) as:

$$\Delta\mathbf{u}_\mathbf{v} = \begin{bmatrix} \Delta\mathbf{u}(k) & \Delta\mathbf{u}(k+1) & \cdots & \Delta\mathbf{u}(k+N_c-1) \end{bmatrix}^T \tag{12}$$

with $\Delta\mathbf{u}_v \in \mathbb{R}^{n_{in}N_c}$. If the state vector $\mathbf{x}(k)$ is known, either measured or estimated, the predicted state vectors in a horizon of N_p future sampling instants can be obtained:

$$\begin{aligned}
\mathbf{x}(k+1|k) &= \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\Delta\mathbf{u}(k) \\
\mathbf{x}(k+2|k) &= \mathbf{A}\,\mathbf{x}(k+1|k) + \mathbf{B}\,\Delta\mathbf{u}(k+1) \\
&\cdots \\
\mathbf{x}(k+N_p|k) &= \mathbf{A}^{N_p}\,\mathbf{x}(k) + \mathbf{A}^{N_p-1}\,\mathbf{B}\,\Delta\mathbf{u}(k) + \mathbf{A}^{N_p-2}\,\mathbf{B}\,\Delta\mathbf{u}(k+1) + \cdots + \mathbf{A}^{N_p-N_c}\,\mathbf{B}\,\Delta\mathbf{u}(k+N_c-1)
\end{aligned} \tag{13}$$

In a similar way, the predicted output vector, defined as $\mathbf{y}_v = \begin{bmatrix}\mathbf{y}(k+1|k) & \mathbf{y}(k+2|k) & \cdots & \mathbf{y}(k+N_p|k)\end{bmatrix}^T$, where $\mathbf{y}_v \in \mathbb{R}^{n_{out}N_p}$, is expressed as follows:

$$\mathbf{y}_v = \mathbf{F}\,\mathbf{x}(k) + \boldsymbol{\Phi}\,\Delta\mathbf{u}_v \tag{14}$$

Matrices $\mathbf{F} \in \mathbb{R}^{n_{out}N_p \times n_{aug}}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{n_{out}N_p \times n_{in}N_c}$ assume the following meaning:

$$\mathbf{F} = \begin{bmatrix} \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \mathbf{C}\mathbf{A}^3 \\ \mathbf{C}\mathbf{A}^4 \\ \vdots \\ \mathbf{C}\mathbf{A}^{N_p} \end{bmatrix} \tag{15}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{C}\mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{A}^2\mathbf{B} & \mathbf{C}\mathbf{A}\mathbf{B} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{A}^3\mathbf{B} & \mathbf{C}\mathbf{A}^2\mathbf{B} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{C}\mathbf{A}^{N_p-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{N_p-2}\mathbf{B} & \cdots & \mathbf{C}\mathbf{A}^{N_p-N_c}\mathbf{B} \end{bmatrix} \tag{16}$$

2.2.4. Definition of the cost function for standard MPC

Once the matrix equation regarding the prediction phase is achieved, the next step consists in defining the cost function $J \in \mathbb{R}$. In trajectory tracking problems, it usually assumes the following expression:

$$J = \sum_{i=1}^{N_p} \left\| \mathbf{r}(k) - \mathbf{y}(k+i|k) \right\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N_c-1} \left\| \Delta\mathbf{u}(k+i) \right\|_{\mathbf{R}}^2 + \left\| \mathbf{r}(k) - \mathbf{y}(k+N_p|k) \right\|_{\mathbf{S}}^2 \tag{17}$$

that corresponds to the following matrix form:

$$J = \left(\mathbf{r}_{std} - \mathbf{y}_v\right)^T \mathbf{W}_y \left(\mathbf{r}_{std} - \mathbf{y}_v\right) + \Delta\mathbf{u}_v{}^T \mathbf{W}_{\Delta u} \Delta\mathbf{u}_v \tag{18}$$

Matrices $\mathbf{Q} \in \mathbb{R}^{n_{out} \times n_{out}}$ and $\mathbf{R} \in \mathbb{R}^{n_{in} \times n_{in}}$ are introduced to weight the tracking errors and the control input effort, respectively, while $\mathbf{S} \in \mathbb{R}^{n_{out} \times n_{out}}$ is the weighting matrix related to the terminal cost term. By looking at the matrix notation, it can be noticed that $\mathbf{W}_{\mathbf{y}} = diag\left(\mathbf{Q}, \quad \cdots, \quad \mathbf{Q}, \quad \mathbf{S}\right) \in \mathbb{R}^{n_{out}N_p \times n_{out}N_p}$ and $\mathbf{W}_{\Delta \mathbf{u}} = diag\left(\mathbf{R}, \quad \cdots, \quad \mathbf{R}\right) \in \mathbb{R}^{n_{in}N_c \times n_{in}N_c}$ are the weighting matrices related to the tracking error vector and to the control input, respectively, while $\mathbf{r}_{\mathbf{std}} \in \mathbb{R}^{n_{out}N_p}$ is the vector containing the information on the reference trajectory along the prediction horizon.
Standard MPCs [32] usually define vector $\mathbf{r}_{\mathbf{std}}$ as follows:

$$\mathbf{r}_{\mathbf{std}} = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \cdots & \mathbf{I}_{n_{out}} \end{bmatrix}^T \mathbf{r}(k) = \mathbf{N}_{\mathbf{std}}\, \mathbf{r}(k) \tag{19}$$

where $\mathbf{r}(k) \in \mathbb{R}^{n_{out}}$ is the k^{th} sample of the reference signal. However, as it will be shown in Section 2.4, such a formulation is correct just when constant, or piecewise-constant, set-points should be tracked; for this reason, most of the papers in the literature exploit MPC to track step signals. In contrast, when time-changing references should be tracked, such as time-varying motion laws, this formulation is no longer effective to track the reference without delay. Indeed, in the standard MPC formulations, the reference signal is kept constant over the prediction horizon lasting N_p samples by means of matrix $\mathbf{N}_{\mathbf{std}} = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \cdots & \mathbf{I}_{n_{out}} \end{bmatrix}^T \in \mathbb{R}^{n_{out}N_p}$, leading therefore to a piecewise-constant approximation of the actual reference, while, in contrast, the prediction of the system dynamics is performed through matrix $\mathbf{F}$.

2.2.5. Autonomous state-space model for MPC with Embedded Reference Dynamics

The problem of using a piecewise-constant reference in minimizing J is overcome in this work by properly including the reference dynamics within the cost function. The novel approach here proposed is to exploit a representation of the reference, over the prediction horizon, through an autonomous state-space model:

$$\begin{cases} \mathbf{x}_{\mathbf{r}}(k+1) = \mathbf{A}_{\mathbf{r}}\, \mathbf{x}_{\mathbf{r}}(k) \\ \mathbf{r}(k) = \mathbf{C}_{\mathbf{r}}\, \mathbf{x}_{\mathbf{r}}(k) \end{cases} \tag{20}$$

where $\mathbf{r}(k) \in \mathbb{R}^{p_r}$ represents the predicted reference vector, $\mathbf{x}_{\mathbf{r}}(k) \in \mathbb{R}^{n_r}$ is the state vector capable to describe the reference dynamics, $\mathbf{A}_{\mathbf{r}} \in \mathbb{R}^{n_r \times n_r}$ is the dynamic matrix of the autonomous state-space model, $\mathbf{C}_{\mathbf{r}} \in \mathbb{R}^{p_r \times n_r}$ is the output matrix, n_r is the number of entries of $\mathbf{x}_{\mathbf{r}}(k)$ and p_r indicates the number of auto-generated references to be taken into account during the prediction horizon.
To achieve such an autonomous state-space representation of the reference, this Chapter exploits the theory of Dynamic Mode Decomposition (DMD) and applies it to the designed reference trajectory. This theory made its first appearance in the fluid dynamics community as a technique to decompose complex flows into a trivial representation based on spatiotemporal coherent structures [33–35]. Here, DMD is exploited in a different field, by taking advantage of its effectiveness and low computational effort; on the other hand, the evaluation of the matrices $\mathbf{A}_{\mathbf{r}}$ and $\mathbf{C}_{\mathbf{r}}$ can be also computed offline and stored in the controller memory.
Let us define $\mathbf{p}(t)$, $\mathbf{s}(t)$, $\mathbf{a}(t) \in \mathbb{R}^{n_{out}}$ as the reference position, speed and acceleration, respectively, in the continuous-time domain, while the discretized counterparts will be indicated as $\mathbf{p}(k)$, $\mathbf{s}(k)$, $\mathbf{a}(k) \in \mathbb{R}^{n_{out}}$. Two approaches can be adopted to define the state vector $\mathbf{x}_{\mathbf{r}}(k)$ in tip tracking problem. The first, and simpler approach, is to include the position and the speed references:

$$\mathbf{x}_\mathbf{r}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{s}(k) \end{bmatrix} \tag{21}$$

leading to the following matrices $\mathbf{X}_\mathbf{r} \in \mathbb{R}^{n_r \times (N_s-1)}$ and $\mathbf{X}_\mathbf{r}^{\mathrm{I}} \in \mathbb{R}^{n_r \times (N_s-1)}$:

$$\mathbf{X}_\mathbf{r} = \begin{bmatrix} \mathbf{p}(k=1) & \mathbf{p}(k=2) & \mathbf{p}(k=3) & \cdots & \mathbf{p}(k=N_s-1) \\ \mathbf{s}(k=1) & \mathbf{s}(k=2) & \mathbf{s}(k=3) & \cdots & \mathbf{s}(k=N_s-1) \end{bmatrix} \tag{22}$$

$$\mathbf{X}_\mathbf{r}^{\mathrm{I}} = \begin{bmatrix} \mathbf{p}(k=2) & \mathbf{p}(k=3) & \mathbf{p}(k=4) & \cdots & \mathbf{p}(k=N_s) \\ \mathbf{s}(k=2) & \mathbf{s}(k=3) & \mathbf{s}(k=4) & \cdots & \mathbf{s}(k=N_s) \end{bmatrix} \tag{23}$$

The search for the dynamic state-space matrix $\mathbf{A}_\mathbf{r}$, as defined in Eq.(20), can be solved through DMD as finding best-fit linear operator that allows writing the following relationship:

$$\mathbf{X}_\mathbf{r}^{\mathrm{I}} = \mathbf{A}_\mathbf{r}\,\mathbf{X}_\mathbf{r} \tag{24}$$

that leads to the following result, where † denotes the matrix pseudoinverse:

$$\mathbf{A}_\mathbf{r} = \mathbf{X}_\mathbf{r}^{\mathrm{I}}\mathbf{X}_\mathbf{r}^{\dagger} \tag{25}$$

Finally, the output matrix $\mathbf{C}_\mathbf{r}$ should be chosen to map one or more entries of $\mathbf{x}_\mathbf{r}(k)$ in the output with the focus to be consistent with the definition of $\mathbf{y}_\mathbf{d}$ in Eq.(5), by ensuring at the same time that the couple $(\mathbf{A}_\mathbf{r}, \mathbf{C}_\mathbf{r})$ is observable. In this Chapter and with this definition of $\mathbf{x}_\mathbf{r}(k)$, since it has been assumed that the output is the position vector (see e.g. Eq.(3)), $\mathbf{C}_\mathbf{r}$ assumes the following form:

$$\mathbf{C}_\mathbf{r} = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{0}_{n_{out}} \end{bmatrix} \tag{26}$$

The second possible definition of $\mathbf{x}_\mathbf{r}(k)$ includes the reference acceleration $\mathbf{a}(k)$ together with position and speed, to better capture the reference dynamics:

$$\mathbf{x}_\mathbf{r}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{s}(k) \\ \mathbf{a}(k) \end{bmatrix} \tag{27}$$

leading to the following matrices $\mathbf{X}_\mathbf{r}$ and $\mathbf{X}_\mathbf{r}^{'}$:

$$\mathbf{X}_\mathbf{r} = \begin{bmatrix} \mathbf{p}(k=1) & \mathbf{p}(k=2) & \mathbf{p}(k=3) & \cdots & \mathbf{p}(k=N_s-1) \\ \mathbf{s}(k=1) & \mathbf{s}(k=2) & \mathbf{s}(k=3) & \cdots & \mathbf{s}(k=N_s-1) \\ \mathbf{a}(k=1) & \mathbf{a}(k=2) & \mathbf{a}(k=3) & \cdots & \mathbf{a}(k=N_s-1) \end{bmatrix} \tag{28}$$

$$\mathbf{X}_\mathbf{r}^{\mathbf{l}} = \begin{bmatrix} \mathbf{p}(k=2) & \mathbf{p}(k=3) & \mathbf{p}(k=4) & \cdots & \mathbf{p}(k=N_s) \\ \mathbf{s}(k=2) & \mathbf{s}(k=3) & \mathbf{s}(k=4) & \cdots & \mathbf{s}(k=N_s) \\ \mathbf{a}(k=2) & \mathbf{a}(k=3) & \mathbf{a}(k=4) & \cdots & \mathbf{a}(k=N_s) \end{bmatrix} \tag{29}$$

The dynamic matrix of the state-space, $\mathbf{A}_\mathbf{r}$, is computed, again, through Eq.(25) and the output matrix $\mathbf{C}_\mathbf{r}$ is chosen accordingly.

It should be noted that the computation of $\mathbf{A}_\mathbf{r}$ through Eq.(25) might be numerically ill-conditioned for some pairs n_r and N_s. In this case, Singular Value Decomposition (SVD) can be exploited to improve the solution by decomposing $\mathbf{X}_\mathbf{r}$ and $\mathbf{X}_\mathbf{r}^{\mathbf{l}}$ as follows:

$$\mathbf{X}_\mathbf{r} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^* \tag{30}$$

$$\mathbf{X}_\mathbf{r}^{\mathbf{l}} = \mathbf{U}^{\mathbf{l}}\,\boldsymbol{\Sigma}^{\mathbf{l}}\,\mathbf{V}^{\mathbf{l}*} = \mathbf{A}_\mathbf{r}\,\mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^* \tag{31}$$

Matrix $\mathbf{U}$ contains the spatial structures and its columns corresponds to the proper orthogonal decomposition modes, which are ranked hierarchically from the most to the least important in terms of capturing the variance of $\mathbf{X}_\mathbf{r}$; matrix $\boldsymbol{\Sigma}$ contains the singular values, while matrix $\mathbf{V}$ describes the temporal structures [33]. Therefore, by choosing a subset of the first most meaningful columns of $\mathbf{U}$ and $\mathbf{U}^{\mathbf{l}}$, the dominant contributions to the system dynamics are accurately captured through a reduced number of data while ensuring good numerical conditioning of the computation. Finally, $\mathbf{A}_\mathbf{r}$ is computed by applying the pseudo-inverse operation to the SVD quantities of $\mathbf{X}_\mathbf{r}$ and the solution will have the following expression:

$$\mathbf{A}_\mathbf{r} = \mathbf{X}_\mathbf{r}^{\mathbf{l}}\,\mathbf{V}\,\boldsymbol{\Sigma}^{-1}\,\mathbf{U}^* \tag{32}$$

Once $\mathbf{A}_\mathbf{r}$ is computed, either through Eq.(25) or Eq.(32), and since that the reference state vector $\mathbf{x}_\mathbf{r}(k_i)$ is available, the future reference trajectory can be computed as follows:

$$\begin{aligned} \mathbf{r}(k+1\,|\,k) &= \mathbf{C}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k+1\,|\,k) = \mathbf{C}_\mathbf{r}\,\mathbf{A}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k) \\ \mathbf{r}(k+2\,|\,k) &= \mathbf{C}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k+2\,|\,k) = \mathbf{C}_\mathbf{r}\,\mathbf{A}_\mathbf{r}^{\,2}\,\mathbf{x}_\mathbf{r}(k) \\ &\cdots \\ \mathbf{r}(k+N_p\,|\,k) &= \mathbf{C}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k+N_p\,|\,k) = \mathbf{C}_\mathbf{r}\,\mathbf{A}_\mathbf{r}^{\,N_p}\,\mathbf{x}_\mathbf{r}(k) \end{aligned} \tag{33}$$

The predicted reference through the autonomous state-space matrices can be written in vector form:

$$\mathbf{r}_{\mathbf{erd}} = \begin{bmatrix} \mathbf{r}(k+1\,|\,k) & \mathbf{r}(k+2\,|\,k) & \cdots & \mathbf{r}(k+N_p\,|\,k) \end{bmatrix}^T \tag{34}$$

and the following matrix equation can therefore be stated:

$$\mathbf{r}_{\mathbf{erd}} = \mathbf{N}_{\mathbf{erd}}\,\mathbf{x}_\mathbf{r}(k) \tag{35}$$

where $\mathbf{r}_{\mathbf{erd}} \in \mathbb{R}^{p_r N_p}$ and matrix $\mathbf{N}_{\mathbf{erd}} \in \mathbb{R}^{p_r N_p \times n_r}$ is defined as follows:

$$\mathbf{N}_{\mathbf{erd}} = \begin{bmatrix} \mathbf{C}_{\mathbf{r}}\,\mathbf{A}_{\mathbf{r}} \\ \mathbf{C}_{\mathbf{r}}\,\mathbf{A}_{\mathbf{r}}^{2} \\ \mathbf{C}_{\mathbf{r}}\,\mathbf{A}_{\mathbf{r}}^{3} \\ \mathbf{C}_{\mathbf{r}}\,\mathbf{A}_{\mathbf{r}}^{4} \\ \vdots \\ \mathbf{C}_{\mathbf{r}}\,\mathbf{A}_{\mathbf{r}}^{N_p} \end{bmatrix} \tag{36}$$

The temporal evolution of the autonomous state-space model defined in Eq.(35) can be easily included as the reference in the MPC algorithm adopted in the cost function J, which is still defined as in Eq.(18), by substituting $\mathbf{r}_{\mathbf{std}}$ with $\mathbf{r}_{\mathbf{erd}}$. It should be noted that the proposed approach, once $\mathbf{r}_{\mathbf{erd}}$ is computed, allows using standard algorithms and routines for the minimization of J, thus exploiting any commercial (or open source) software that provides effective and reliable numerical solutions. By considering the simple calculations required to obtain $\mathbf{r}_{\mathbf{erd}}$, this is, to the author's opinion, a relevant benefit of the proposed approach.

Once the cost function J is defined, solving $\partial J / \partial \Delta \mathbf{U} = 0$ leads to the optimal control signal. In the case of the unconstrained problem, the following analytical solution is obtained:

$$\Delta \mathbf{u}_{\mathbf{v}} = \left(\mathbf{\Phi}^{T} \mathbf{\Phi} + \mathbf{W}_{\Delta \mathbf{u}} \right)^{-1} \mathbf{\Phi}^{T} \left(\mathbf{N}_{\mathbf{erd}}\, \mathbf{x}_{\mathbf{r}}(k) - \mathbf{F}\, \mathbf{x}(k) \right) \tag{37}$$

Among all the entries of $\Delta \mathbf{u}_{\mathbf{v}}$, just the first one, denoted as $\Delta \mathbf{u}(k)$, is considered while the remaining ones are discarded, by adopting the Receding Horizon approach.

2.2.6. Definition of constraints for the MPC-ERD with embedded integrator

Constraints due to limitations on the feasible input (control forces) can be incorporated within the MPC formulation, leading to a constrained control design where $\mathbf{u}(k)$ is bounded by the maximum and the minimum values allowed for the control signal, named $\mathbf{u}_{\mathbf{min}} \in \mathbb{R}^{n_{in}}$ and $\mathbf{u}_{\mathbf{max}} \in \mathbb{R}^{n_{in}}$ respectively (the following inequality is element-wise):

$$\mathbf{u}_{\mathbf{min}} \leq \mathbf{u}(k) \leq \mathbf{u}_{\mathbf{max}} \tag{38}$$

In the case of MPC formulation with embedded integrator, where the difference control $\Delta \mathbf{u}(k)$ is the input, the previous inequalities have to be rewritten as constraints on $\Delta \mathbf{u}(k)$:

$$\Delta \mathbf{u}_{\mathbf{min}} \leq \Delta \mathbf{u}(k) \leq \Delta \mathbf{u}_{\mathbf{max}} \tag{39}$$

The following relationship holds between $\mathbf{u}_{\mathbf{v}}$, $\Delta \mathbf{u}_{\mathbf{v}}$ and $\mathbf{u}(k-1)$:

$$\begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \mathbf{u}(k+2) \\ \mathbf{u}(k+3) \\ \cdots \\ \mathbf{u}(k+N_c-1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{in}} \\ \mathbf{I}_{n_{in}} \\ \mathbf{I}_{n_{in}} \\ \mathbf{I}_{n_{in}} \\ \cdots \\ \mathbf{I}_{n_{in}} \end{bmatrix} \mathbf{u}(k-1) + \begin{bmatrix} \mathbf{I}_{n_{in}} & \mathbf{0}_{n_{in}} & \mathbf{0}_{n_{in}} & \cdots & \mathbf{0}_{n_{in}} \\ \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{0}_{n_{in}} & \cdots & \mathbf{0}_{n_{in}} \\ \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \cdots & \mathbf{0}_{n_{in}} \\ \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \cdots & \mathbf{0}_{n_{in}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \cdots & \mathbf{I}_{n_{in}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \Delta\mathbf{u}(k+2) \\ \Delta\mathbf{u}(k+3) \\ \cdots \\ \Delta\mathbf{u}(k+N_c-1) \end{bmatrix} \tag{40}$$

which can be rewritten in a more compact form as follows (with obvious meaning of $\mathbf{C}_1 \in \mathbb{R}^{n_{in}N_c \times n_{in}}$ and $\mathbf{C}_2 \in \mathbb{R}^{n_{in}N_c \times n_{in}}$):

$$\mathbf{u}_\mathrm{v} = \mathbf{C}_1 \mathbf{u}(k-1) + \mathbf{C}_2 \Delta\mathbf{u}_\mathrm{v} \tag{41}$$

Therefore, defining the input constraints along the entire control horizon through the vectors $\mathbf{u}_{\mathbf{lower}} \in \mathbb{R}^{n_{in}N_c}$ and $\mathbf{u}_{\mathbf{upper}} \in \mathbb{R}^{n_{in}N_c}$:

$$\begin{aligned} \mathbf{u}_{\mathbf{lower}} &= \begin{bmatrix} \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \cdots & \mathbf{I}_{n_{in}} \end{bmatrix}^T \mathbf{u}_{\mathbf{min}} \\ \mathbf{u}_{\mathbf{upper}} &= \begin{bmatrix} \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \mathbf{I}_{n_{in}} & \cdots & \mathbf{I}_{n_{in}} \end{bmatrix}^T \mathbf{u}_{\mathbf{max}} \end{aligned} \tag{42}$$

constraints on all the entries of $\Delta\mathbf{u}_\mathrm{v}$ can be, finally, defined as follows:

$$\mathbf{u}_{\mathbf{lower}} \leq \mathbf{C}_1 \mathbf{u}(k-1) + \mathbf{C}_2 \Delta\mathbf{u}_\mathrm{v} \leq \mathbf{u}_{\mathbf{upper}} \tag{43}$$

Since the cost function J is a quadratic and the constraints are linear inequalities, the constrained optimization problem becomes a standard quadratic programming problem. Therefore, as previously stated, standard solution algorithms can be adopted.

2.3. Test case description and system model

In this Chapter, a test case with $n = 2$ and $n_{in} = 1$ is considered. The studied system is shown in Figure 1, through a picture of the experimental setup and a simplified kinematic scheme. The system is a planar two-link robot arm with two revolute joints: joint A is actuated by a DC motor, while joint B is passive with a torsional spring (made through two linear springs). Such a system is an example of underactuated system and mimics a flexible arm; for this reason it is often assumed as the base for many studies on vibration control [36].

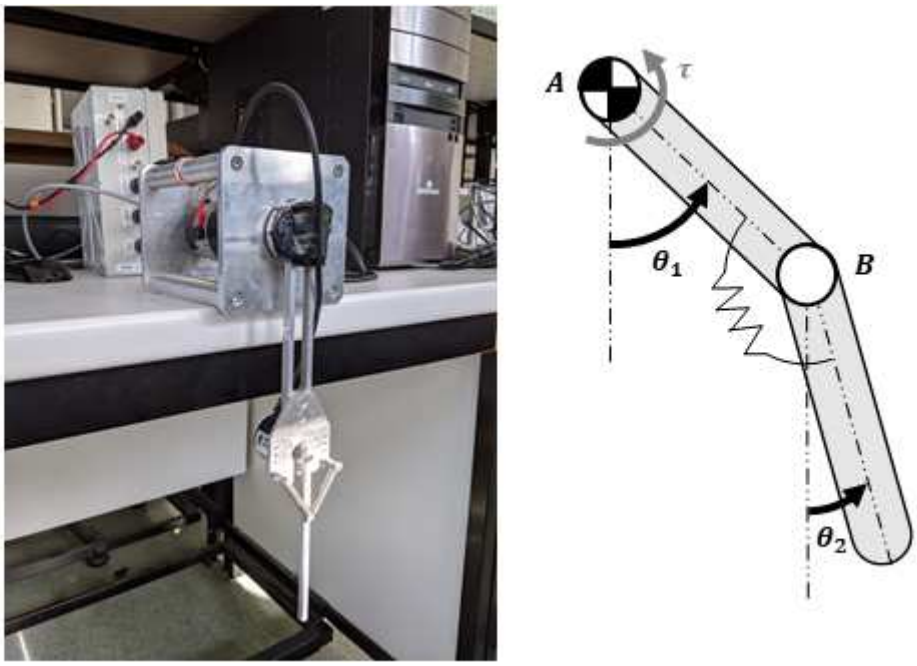


Figure 1. Picture and kinematic scheme of the underactuated robotic arm.

Incremental encoders are used to measure the link rotations, whose number of pulses-per-revolute is denoted as $N_{ppr} \in \mathbb{R}$. The link connected to the actuated joint A is characterized by length $l_1 \in \mathbb{R}$, mass $m_1 \in \mathbb{R}$ and moment of inertia $J_1 \in \mathbb{R}$ (defined with respect of point A). The other link is characterized by length $l_2 \in \mathbb{R}$, mass $m_2 \in \mathbb{R}$ and a moment of inertia $J_2 \in \mathbb{R}$ (defined with respect of point B). The moment of inertia of the rotor of the DC motor driving link 1 is $J_m \in \mathbb{R}$, to be summed with the moment of inertia of the rigid coupling $J_d \in \mathbb{R}$. Both the encoders have mass $m_{enc} \in \mathbb{R}$; the encoder of the non-actuated link is mounted at the ending part of the actuated link through a mounting bracket with mass $m_b \in \mathbb{R}$. The linear springs that couple the two links behave as a rotational spring, whose equivalent rotational stiffness $k_s \in \mathbb{R}$ has been identified through analysis of the system free responses. The two independent coordinates that have been chosen are the absolute positions of the two links with respect to the vertical configuration, denoted $\theta_1 \in \mathbb{R}$ and $\theta_2 \in \mathbb{R}$ as shown in Figure 1, leading to the following vectors:

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix} \quad \dot{\boldsymbol{\theta}}(t) = \begin{bmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \end{bmatrix} \quad \ddot{\boldsymbol{\theta}}(t) = \begin{bmatrix} \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} \tag{44}$$

By applying the Lagrangian approach, the nonlinear model of the system is formulated, and it is omitted here for brevity. Then, by linearizing it around the stable equilibrium point defined by the vertical position of both links in the low configuration ($\theta_1 = \theta_2 = 0$), the system dynamics is described by the following linear matrices:

$$\mathbf{M}_\mathbf{e} = \begin{bmatrix} J_m + J_d + J_1 + (m_2 + m_{enc} + m_b) l_1^2 & \frac{1}{2} m_2 l_1 l_2 \\ \frac{1}{2} m_2 l_1 l_2 & J_2 \end{bmatrix} \tag{45}$$

$$\mathbf{D_e} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{46}$$

$$\mathbf{K_e} = \begin{bmatrix} k_s + \left(\frac{1}{2}m_1 + m_{enc} + m_b + m_2\right) g\, l_1 & -k_s \\ -k_s & k_s + \frac{1}{2} m_2\, g\, l_2 \end{bmatrix} \tag{47}$$

$$\mathbf{P} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{48}$$

$\mathbf{D_e} \in \mathbb{R}^{2\times 2}$ is the linear damping matrix, whose entries $d_{11}, d_{12}, d_{21}, d_{22} \in \mathbb{R}$ have been obtained through the Rayleigh model and identified through experimental model updating [37]. Such a model exactly holds in an infinitesimal neighbourhood about the equilibrium configuration. Nonetheless, it will be adopted for finite displacements about it and, therefore, the controller should be robust enough to handle such an uncertainty source.

The test consists of a sort of pick-and-place repetitive cycle, from the pick-position (x_{pick}, y_{pick}) to the place one (x_{place}, y_{place}), and then back again to the pick-location. Indeed, pick-and-place tasks have a great importance for robots employed in manufacturing plants [38] and hence the end-effector motion should be properly controlled.

The controlled output, and hence the output of the state-space model, is the end-effector displacement along the horizontal axis; as it is common in controlling underactuated systems, the number of controlled outputs is equal to the number of independent actuators [2,3]. On the other hand, the displacement along the vertical axis is negligible in the presence of small displacements about the assumed configuration.

This choice of the output leads to a nonminimum-phase system, since the transfer function from the control torque to the tip horizontal position features one zero with positive real part (see Figure 2).

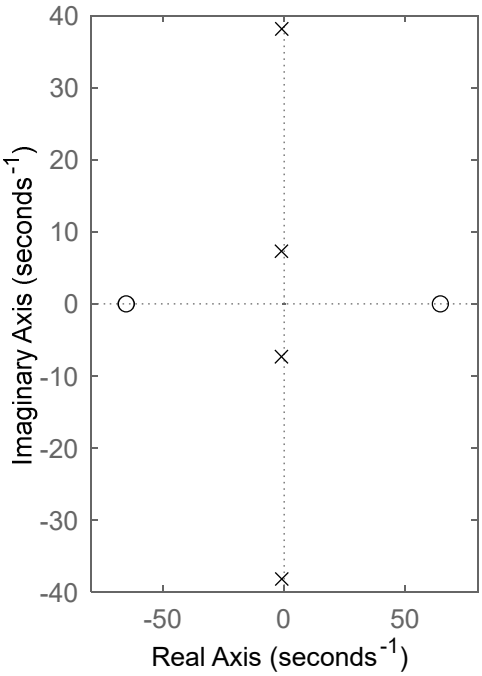


Figure 2. Poles and zeros of the open loop receptance from the control torque to the tip horizontal position.

The coordinates of the pick-and-place locations are defined by fictitiously introducing a rigid behaviour of the mechanism, that should behave as a single-DOF rigid link whose length is $l_{tot} = l_1 + l_2$, leading to an equivalent pick-angle θ_{pick} and a place-angle θ_{place} (the both are defined with the same reference frame of θ_1):

$$\begin{aligned} x_{pick} &= l_{tot} \sin\left(\theta_{pick}\right) \\ x_{place} &= l_{tot} \sin\left(\theta_{place}\right) \end{aligned} \tag{49}$$

The following values have been assumed, to comply with the torque limitation of the DC motor of the experimental setup as $\theta_{pick} = -20°$ and $\theta_{place} = 20°$. A fifth-degree polynomial motion law is assumed to interpolate the boundary positions, for two different motion durations T_{task}, where this latter indicates the time to move from a pick-position to a place one (and vice versa).
All the results contained in this work (both simulative and experimental) have been achieved through *Matlab/Simulink*® software. The Hildreth's quadratic programming algorithm [39,40] is adopted to solve the constrained optimization problem. In the experimental applications, the control schemes have been implemented in a PC thanks to the *Simulink Real-Time* package.

Table 1. System parameters.

System parameters	Values
m_1	0.05 kg
m_2	0.021 kg
l_1	0.16 m
l_2	0.15 m
J_1	4.27×10^{-4} kgm^2
J_2	1.58×10^{-4} kgm^2

m_{enc}	0.10 kg
m_b	0.025 kg
g	9.81 m/s^2
J_m	2.70×10^{-5} kgm^2
J_d	1.08×10^{-5} kgm^2
k_s	0.1772 Nm/rad
d_{11}	6.30×10^{-3} Nms/rad
d_{12}	4.00×10^{-4} Nms/rad
d_{21}	4.00×10^{-4} Nms/rad
d_{22}	3.00×10^{-4} Nms/rad

2.4. Numerical validation

2.4.1. Simulation results

Firstly, simulation results are proposed, also extending the features of the experimental setup (e.g., assuming different encoder resolutions). Secondly, experimental validations will be provided in Section 2.5 to corroborate the numerical results. Simulations are performed through the nonlinear model to assess the robustness of the linear controllers against the presence of nonlinearities. The simulator includes the model of the quantization and of the speed estimation (that is simply performed through a filtered backward Euler numerical derivation), to provide a meaningful representation of the real device. The simulator runs with a sampling time $T_s = 1\ ms$. Three controllers are compared to provide evidence of the benefits of the proposed idea:

- The standard MPC with embedded integrator.
- The MPC-ERD exploiting position and speed of the reference.
- The MPC-ERD exploiting position, speed and acceleration of the reference.

To perform a fair comparison, the same prediction horizon N_p, control horizon N_c and weighting matrices $\mathbf{W_y}$, $\mathbf{W_{\Delta u}}$ have been considered, together with the same control constraints $u_{\min}$ and $u_{\max}$, that are consistent with the actual constraints of the real setup. The values of all these parameters are reported in Table 2 (where $\mathbf{I}_{n_{out}N_p}$ and $\mathbf{I}_{n_{in}N_c}$ are identity matrices with proper dimensions).

The basic idea of the proposed MPC-ERD, which embeds the autonomous representation of the time-varying reference in the prediction horizon, can be inferred from Figure 3, where the predicted reference of the standard MPC is compared with the one of the MPC-ERD.

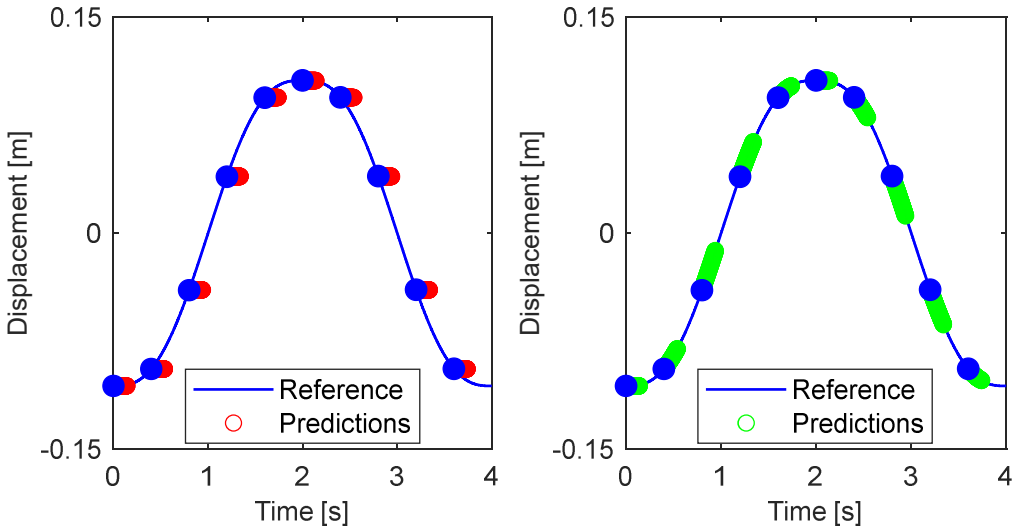


Figure 3. Predicted reference: standard MPC (on the left) and MPC-ERD (on the right).

Table 2. Controller parameters.

Controller parameter	Values
N_p	140
N_c	1
$\mathbf{W_y}$	$1\times\mathbf{I}_{n_{out}N_p}$
$\mathbf{W_{\Delta u}}$	$50\times\mathbf{I}_{n_{in}N_c}$
$u_{\min}$	–0.40 Nm
$u_{\max}$	0.40 Nm

In the simulations, an optical incremental encoder with $N_{ppr}=500$ is assumed in the model for both links, since this value represents the features of the experimental test rig adopted in Section 2.5.
The trajectory tracking responses of the three controllers under investigation are reported in Figure 4, together with their relative tracking errors, for the motion duration equal to $T_{task}=3$ s. The system responses with the remaining motion duration ($T_{task}=1$ s) are not here reported for the sake of brevity: the results regarding both motion timings will be shown in Section 2.5 through the experimental validation, which represents the most interesting part.
Figure 4 clearly highlights that standard MPC is not able to ensure a good trajectory tracking in the presence of time-varying references and nonminimum-phase systems mainly due to two reasons: firstly, the way of modelling the reference in the control synthesis, which is kept constant during the prediction horizon, and secondly the presence of RHP zeros in the plant model. In contrast, it can be noticed that MPC-ERD ensures fast and precise trajectory tacking with quasi-zero tracking delay, thanks to the inclusion of the reference dynamics in its formulation. Additionally, no undershoot is experienced due to the inclusion of the reference dynamics within the controller. It should be noted that no feedforward term has been implemented, and just the feedback provided by the MPC-ERD is adopted. Hence, the contribution of embedding the reference dynamics has an effect similar to the one of feedforward. On the other hand, the MPC-ERD overcomes the difficulties in inverting dynamic models of underactuated, nonminimum-phase systems, that usually impose approximate [2,3,6] or even non-causal [41] solutions of the model inversion problem.

Among the two formulations of MPC-ERD, smaller tracking errors are obtained by the one employing position, speed and acceleration in the autonomous representation of the reference, that provide a more precise model of the reference up to its second derivative. Figure 4 reveals that the standard MPC formulation leads to a tracking error whose maximum value is 15.5 mm and whose Root-Mean Square (RMS) value is 9.5 mm, while on the other hand MPC-ERD formulations with and without the acceleration contribute show tracking errors whose maximum values are 3.3 mm (78.7% reduction) and 2.8 mm (81.9% reduction), respectively, and whose RMS values are 0.53 mm (94.4% reduction) and 0.61 mm (93.6% reduction), respectively.

Using encoders with $N_{ppr} = 500$ is a severe case when feedback control design has to be carried on; indeed, off-the-shelf encoders with higher resolutions are quite common. Clearly, lower values of N_{ppr} increase the noise in the angular positions, which in turn degrade the numerical estimation of speeds; therefore, the bandwidths of the numerical derivative filters should be lowered by reducing the stability margins or the tracking capabilities. To show the role of the encoder resolution within this closed-loop system, simulations have been performed also assuming both the encoders with $N_{ppr} = 4000$. All the three tested controllers have been implemented in this new scenario and the related trajectory tracking errors are shown in Figure 5. Thanks to this comparison, it is evident that encoders with low resolution slightly degrade the performance of all the controllers since worsening the resolution introduces some small harmonic components whose frequency matches the second vibrational modes of the system. If higher resolution encoders are adopted, just a small error is sported by the proposed MPC-ERD.

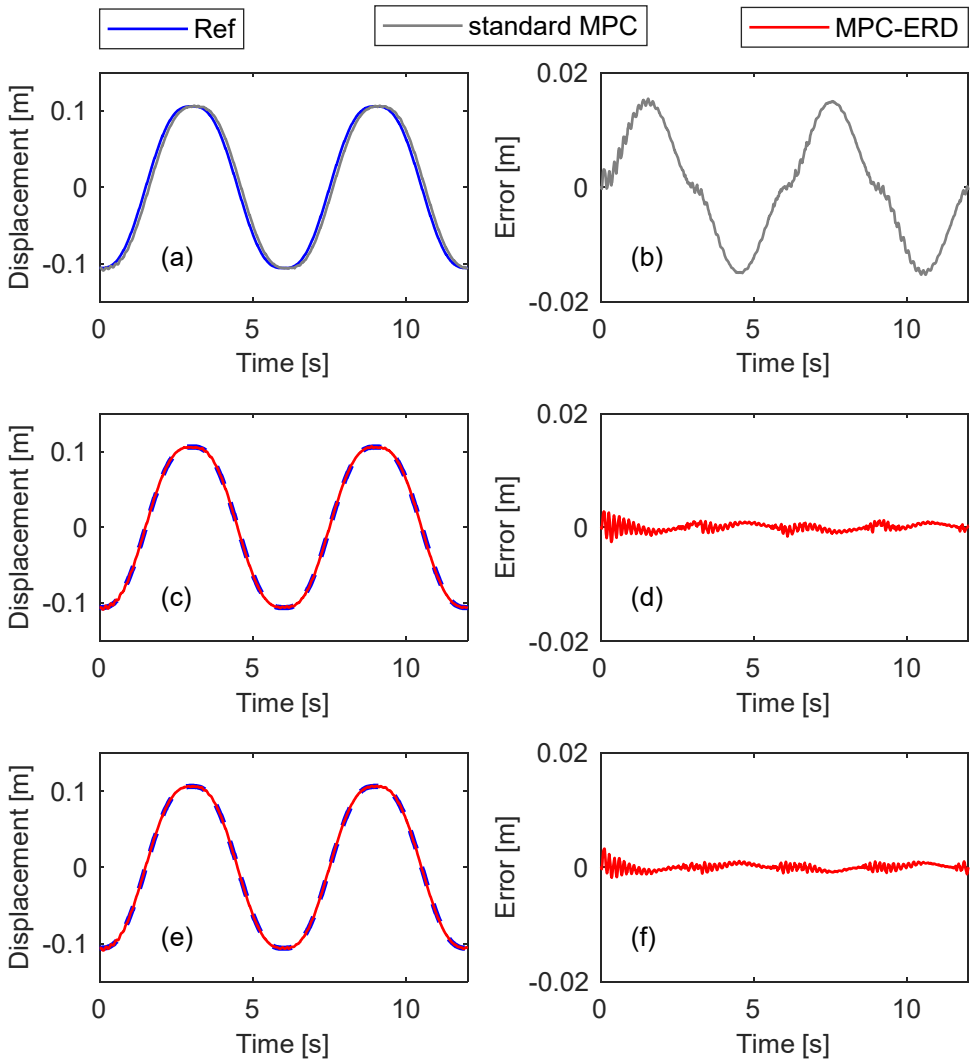


Figure 4. Numerical results with $T_{task} = 3\ s$: tracking responses and errors with standard MPC (a,b), MPC-ERD exploiting position and speed information (c,d) and MPC-ERD exploiting position, speed and acceleration knowledge (e,f).

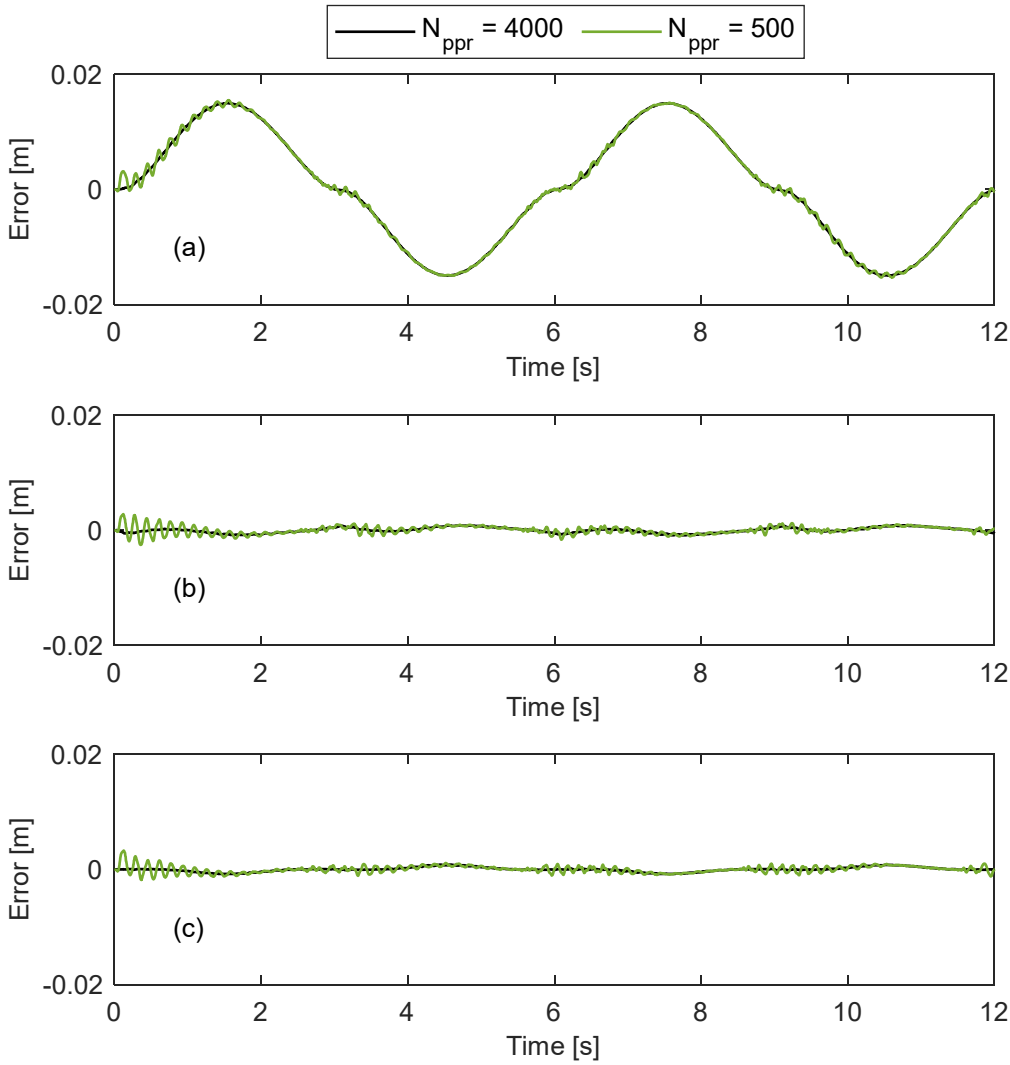


Figure 5. Numerical results with $T_{task} = 3\ s$: tracking errors considering different encoder resolutions with standard MPC (a), MPC-ERD exploiting position and speed information (b) and MPC-ERD exploiting position, speed and acceleration knowledge (c).

A further benchmark to assess the proposed MPC-ERD is here provided, by exploiting the capability of standard MPC algorithms to include both input and output constraints in the optimization process. This benchmark consists of MPC, with embedded integrator and with constraints on the control input, where constraints on the output term are imposed. Since Figure 4 reveals a maximum absolute tracking error equal to 15.5 mm when the standard MPC formulation is considered, an admissible error band of ± 12 mm (-20% respect to the case with only input constraints) has been considered as a sample value of output constraints. A new controller tuning has been adopted, compared to the MPC-ERD and the MPC without output constraints, by looking for the "best" solution sported by the output constrained MPC through a trial-and-error search procedure. The results are shown in Figure 6: the tracking error resides within the admissible band. However, relevant residual oscillations, introduced by the controller itself, appear. Additionally, the control effort (and its time-derivative) increases tremendously, and constraints are not satisfied, thus making its implementation unfeasible

for the real setup. Requiring smaller values of the output error, to match that sported by the MPC-ERD, make the controlled system unstable.

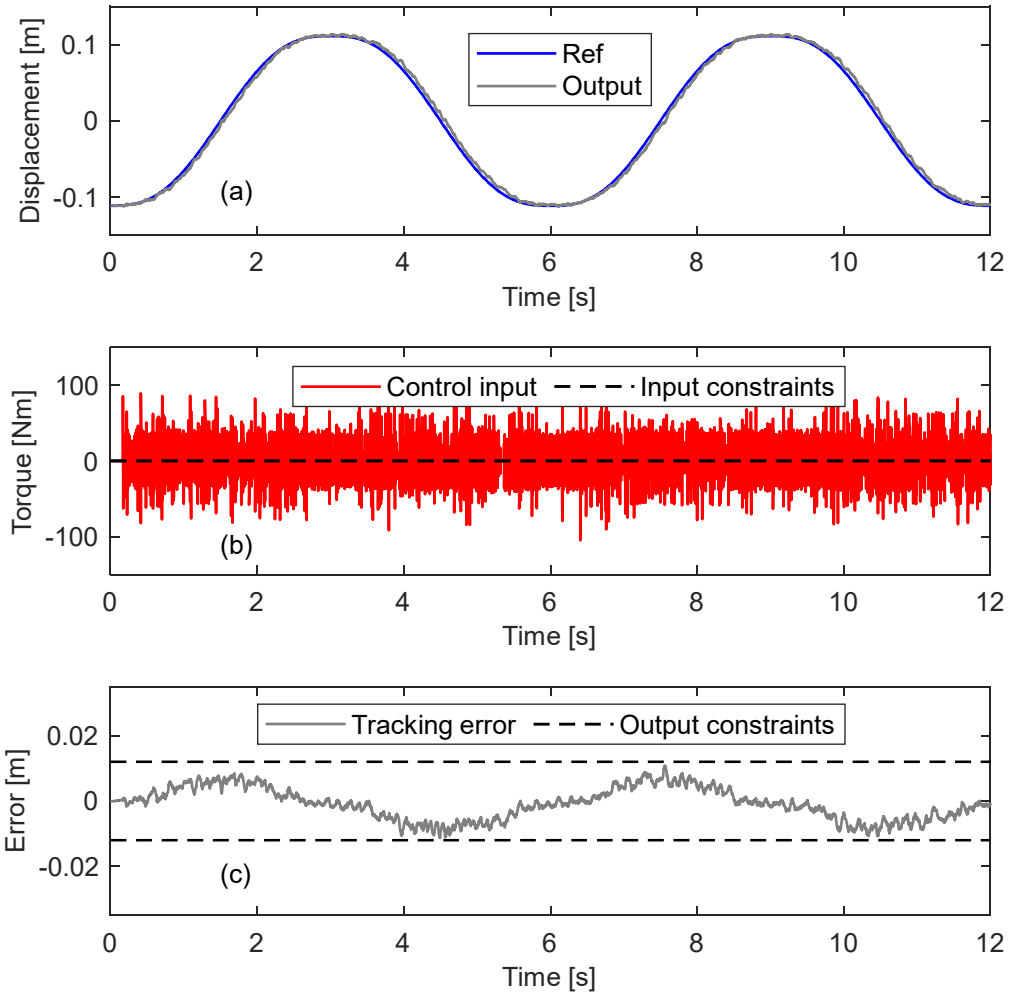


Figure 6. Numerical results in the presence of standard MPC with both input and output constraints, considering $T_{task} = 3\ s$: tracking response (a), control input (b) and tracking error (c).

2.4.2. Robustness analysis

To get a complete picture regarding the controller here proposed, the robustness properties of the closed-loop system are evaluated in this Section. Usually, this evaluation is carried on through classical gain and phase margins, by considering their contribution in an independent manner. However, this kind of evaluation could lead to misleading results that are not conservative and, therefore, combinations of small gain and phase perturbations could make the closed-loop system unstable. Indeed, "real systems differ from their mathematical models in both magnitude and phase" [42] and, consequently, it is more reliable to consider their effects simultaneously. With this goal in mind, the disk margin approach [42] is here exploited by considering perturbations applied to both angular displacements and speeds. In the evaluation of such margins, the system model has been

augmented by including the dynamics of the inner current controller, modelled through a low-pass filter with a bandwidth of 1 kHz.

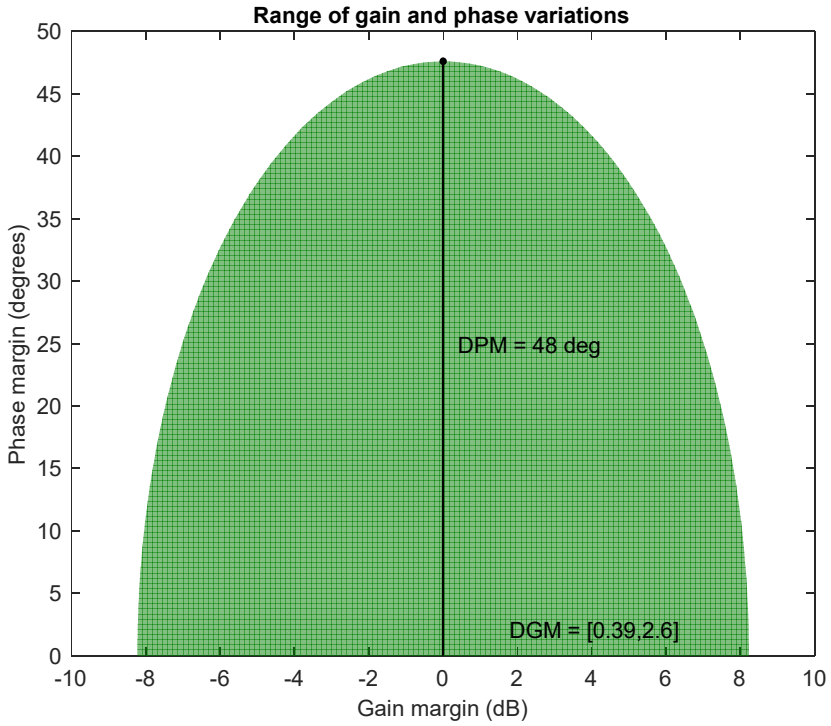


Figure 7. Stability margins of closed-loop system with MPC-ERD.

A "multiloop" analysis is performed to evaluate the robustness of the closed-loop system against perturbations acting on the angular positions and velocities, such as nonlinearities introduced by the quantization of the encoders and delay caused by the speed estimation algorithms. Differently from the "loop-at-a-time" analysis, where each feedback channel is perturbed singularly, the "multiloop" one evaluates the robustness performance by perturbing all feedback channels simultaneously, which represents a more common situation in a real case scenario and a more severe evaluation. The outcome of the disk margin approach applied to MPC-ERD, with the controller parameters reported in Table 2, is shown in Figure 7. It can be noticed that, even in the presence of simultaneous perturbations of angular displacements and speeds, the closed-loop system shows high margins of robustness. In particular, a gain margin equal to ± 8.2 dB is ensured in the presence of no phase variations, while a phase margin equal to $48°$ is obtained when there are no gain variations; all the other intermediate values of robustness, given by the simultaneous variations of gain and phase, are highlighted in Figure 7 by the green area. The same robustness outcome is obtained if the same assessment is applied to the standard MPC formulation, with the same tuning parameters, and therefore it is not reported. This aspect can be also understood by looking at the optimal control input expressed in Eq.(37): for both standard MPC and MPC-ERD the matrices $\mathbf{F}$ and $\mathbf{\Phi}$ assume the same expression and therefore the closed-loop gain is the same. What makes MPC-ERD so attractive is that it is capable to include the reference dynamics into the optimization process ensuring the same stability and robustness properties related to the standard MPC formulation while, at the same time, achieving trajectory tracking with negligible tracking delay and error.

2.4.3. Computational effort

Since the proposed controller has to run on the real setup shown in Figure 1, the computational effort is here discussed and assessed by analysing the required CPU time at each time step. The numerical assessment has been performed using a laptop PC with a 16.0 GB RAM and an octa core processor (11th Gen Intel(R) Core (TM) i7-11800H) characterized by a clock frequency of 2.30 GHz; no parallel computing has been considered and therefore the evaluation is performed by just one core. By looking at Figure 8 it can be noticed that the required CPU time is hugely lower than the sampling time: the Root-Mean Square value of the computational temporal burden is 1.87 μs, which is two orders of magnitude smaller than the sampling time considered for the experimental validation, which is equal to $T_s = 1\ ms$, as stated at the beginning of Section 2.4. This outcome further validates the proposed controller, which is therefore an interesting candidate for trajectory tracking problems in the presence of underactuated multibody systems.

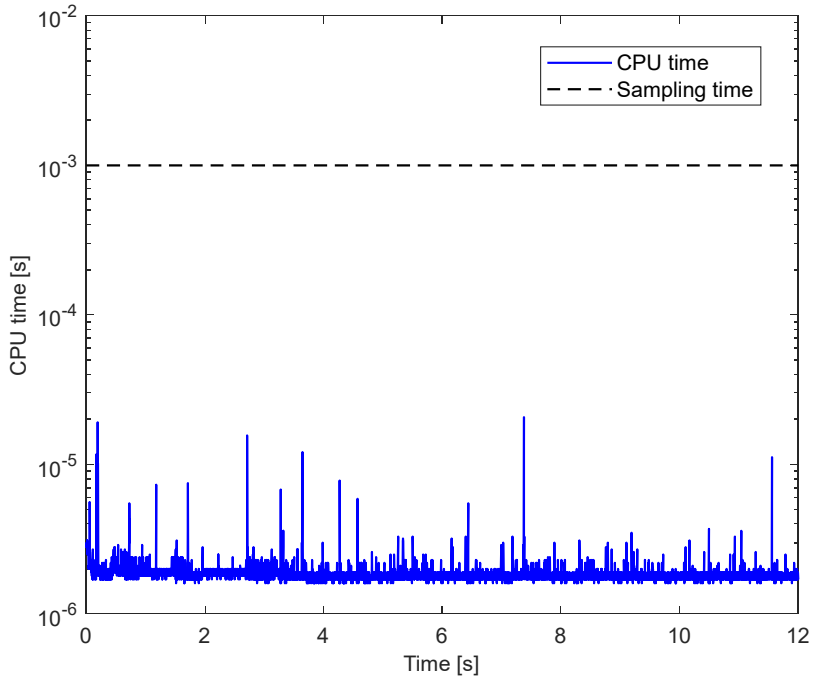


Figure 8. Computational effort required by the proposed MPC-ERD algorithm.

2.5. Experimental validation

The experimental validation exploits the same values of the parameters , N_c, $\mathbf{W}_{\Delta \mathbf{u}}$, $\mathbf{W}_{\mathbf{y}}$, $u_{\min}$ and $u_{\max}$ used in the simulation analysis. The same tests have been also performed. It should be noted that $N_{ppr} = 500$, while higher resolution sensors, as the one simulated in Section 2.4, cannot be evaluated since they are not available in the setup. Speeds have been estimated through a simple numerical derivative with a filtered backward Euler scheme. Clearly, adopting more advanced speed estimation schemes can improve the controller results; nonetheless it is beyond the scopes of this Chapter. The *Simulink Real-Time* package has been used, with $T_s = 1\ ms$, to perform real-time control.

The trajectory tracking responses, together with their relative tracking errors, are shown in Figure 9 and Figure 10 for two different execution times, $T_{task} = 3$ s and $T_{task} = 1$ s, respectively. To have a

clearer image of the benefits coming from the exploitation of the proposed controller, Figure 9 reveals that the standard MPC algorithm leads to a tracking error whose maximum value is 16.9 mm and whose Root-Mean Square value is 9.6 mm, while MPC-ERD formulations with and without the information on the acceleration term show tracking errors whose maximum values are 3.2 mm (81.1% reduction) and 2.9 mm (82.8% reduction), respectively, and whose RMS values are 0.67 mm (93.0% reduction) and 0.74 mm (92.3% reduction), respectively. Moreover, considering $T_{task} = 1$ s, Figure 10 shows that the standard MPC algorithm leads to a tracking error with a maximum value of 47.4 mm and a RMS value of 28.8 mm, while MPC-ERD algorithms with and without the knowledge of the acceleration term show tracking errors whose maximum values are 4.0 mm (91.6% reduction) and 5.5 mm (88.4% reduction), respectively, and whose RMS values are 1.2 mm (95.8% reduction) and 1.9 mm (93.4% reduction), respectively. The benefits coming from the exploitation of the proposed control technique are huge in terms of tracking errors, allowing in this way to achieve high precision tasks, even in the presence of underactuated systems. Comparing Figure 9 with its simulative counterpart proposed in Section 2.4, it can be observed that experimental results and simulative ones are very close to each other. Just a slightly greater oscillation remains uncompensated in the tracking error. Hence, it is confirmed that the proposed MPC has huge benefits in terms of tracking error, despite the presence of optical encoders with a rough resolution and the unavoidable presence of unmodeled nonlinearities and uncertainties.

Finally, the experimental analysis proves the possibility to implement the proposed control scheme in hard real-time controllers, thus allowing for its implementation in real robotic and manufacturing systems.

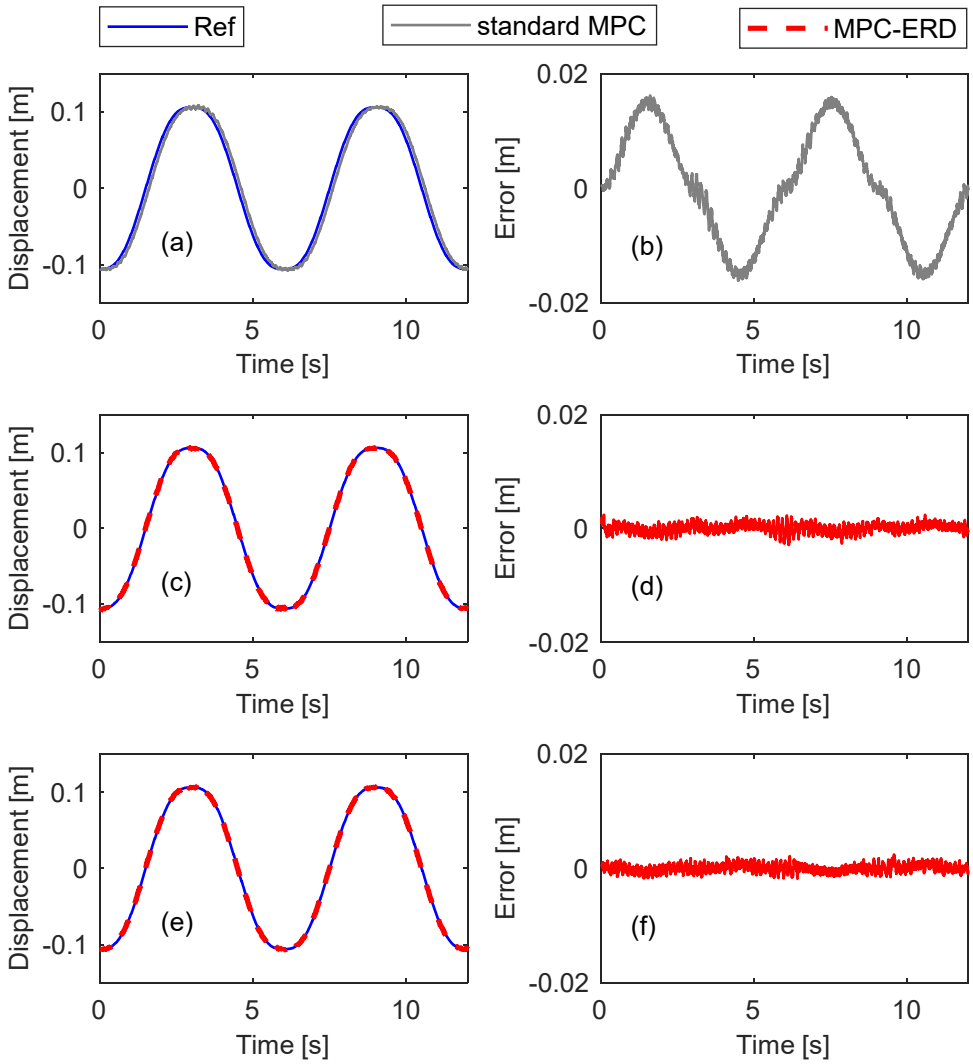


Figure 9. Experimental results with $T_{task} = 3\ s$: tracking responses and errors with standard MPC (a,b), MPC-ERD exploiting position and speed information (c,d) and MPC-ERD exploiting position, speed and acceleration knowledge (e,f).

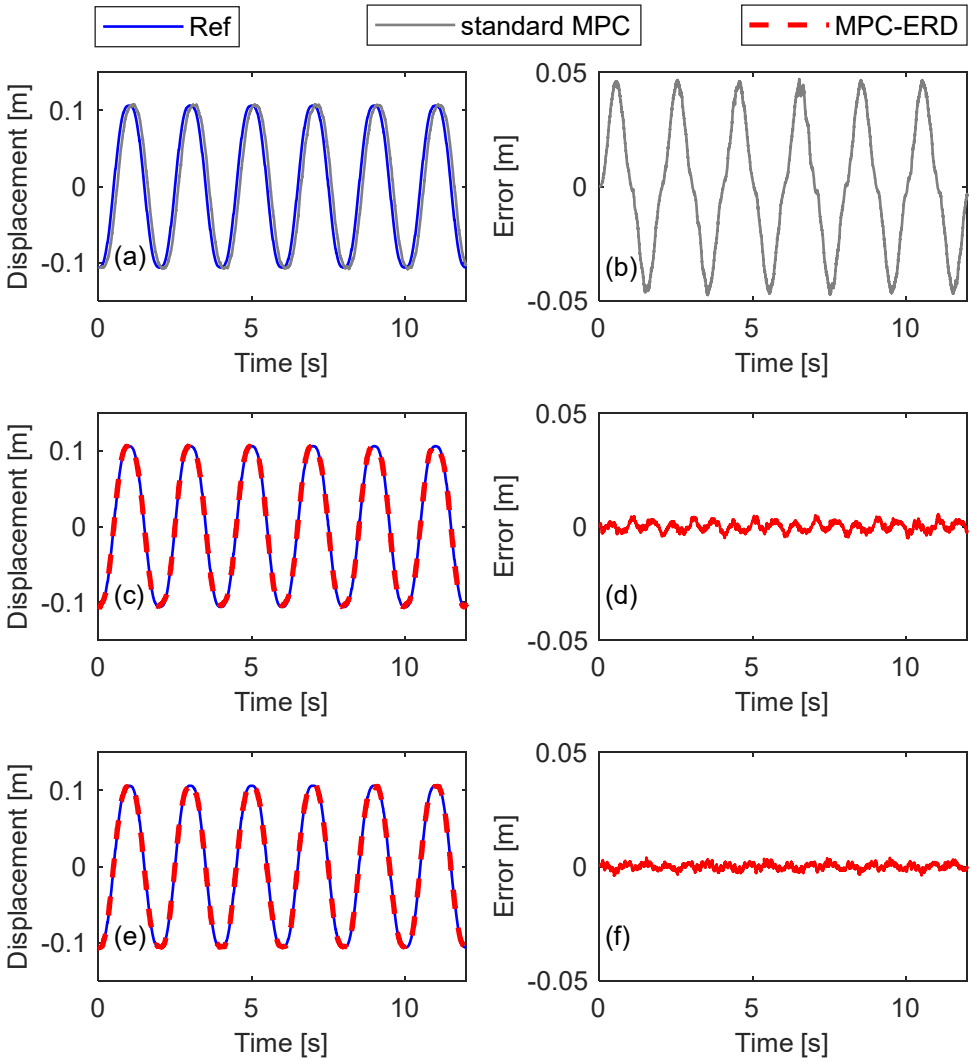


Figure 10. Experimental results with $T_{task} = 1\ s$: tracking responses and errors with standard MPC (a,b), MPC-ERD exploiting position and speed information (c,d) and MPC-ERD exploiting position, speed and acceleration knowledge (e,f).

2.6. Conclusions

This Chapter proposes a new approach to precise tip tracking in underactuated, nonminimum-phase multibody systems, by means of a new architecture of Model Predictive Control (MPC). Besides inheriting the benefits of standard MPC, this new formulation overcomes a relevant limitation of standard MPC and allows tracking time-varying references with negligible error and delay. Additionally, due to the reference embedding, it does not require implementing feedforward controllers, that are usually difficult to design in the case of underactuated, nonminimum-phase systems. Such a result is ensured by exploiting an autonomous state-space model of the tip reference in the prediction phase of MPC and by including it in the controller. For this reason, the controller is named MPC with embedded reference dynamics, MPC-ERD.

The theory of Dynamic Mode Decomposition is exploited to obtain such an autonomous model and allows for real-time control. Two variants of embedding the reference are proposed and investigated: the autonomous system model of the reference described through position and speed; the autonomous system model of the reference described through position, speed and acceleration too.

The effectiveness of MPC-ERD is experimentally validated through a laboratory setup of a two-link underactuated, nonminimum-phase planar manipulator, showing the capability to ensure precise and zero-delay tracking of the tip time-varying reference, even in the presence of position measurement with a rough discretization. The comparison with the "standard" MPC with embedded integrator (also by imposing constraints on the tracking error in the controller design) shows that the novel MPC-ERD outperforms the benchmarks. The experimental implementation and result also corroborate the system stability and the controller feasibility due to a small computational effort

3. Vibration Control of a Two-Link Flexible Robot Arm with Time Delay through the Robust Receptance Method

The same underactuated system of the previous Chapter is here adopted as test case for a different control paradigm, that is Pole Placement. This control technique is commonly used for explicitly imposing natural frequencies and damping of the vibrational modes of linear systems, especially in the presence of flexible unactuated dynamics: indeed, these properties can be easily linked to the achievable performances, such as of settling and rise time. In this Chapter proposes a method for active vibration control to a two-link flexible robot arm in the presence of time delay, by means of robust pole placement. The issue is of practical and theoretical interest as time delay in vibration control can cause instability if not properly taken into account in the controller design. The controller design is performed through the receptance method to exactly assign a pair of pole and to achieve a given stability margin for ensuring robustness to uncertainty. The desired stability margin is achieved by solving an optimization problem based on the Nyquist stability criterion. The method is applied on a laboratory testbed that mimic a typical flexible robotic system employed for pick-and-place applications. The linearization assumption about an equilibrium configuration leads to the identification of the local receptances, holding for infinitesimal displacements about it, and hence applying the proposed control design technique. Nonlinear terms, due to the finite displacements, uncertainty, disturbances, and the coarse encoder quantization, are effectively handled by embedding the robustness requirement into the design. The experimental results, and the consistence with the numerical expectations, demonstrate the method effectiveness and ease of application.

3.1. Introduction

The presence of time delay in controlled systems degrades the closed-loop performance if it is not taken into account in the controller design, and in the worst case, might lead to instability. For example, time delay is due to the physical and operational characteristics of the system, e.g., due to friction [43,44] or due to nature of some manufacturing processes as milling [45] or metal cutting [46]. Delays are also caused by the mechatronic instrumentation employed in experimental real-time systems. In this case, the primary sources of delays are sensors, actuators, and communication networks [47].

Over the years, the most eminent researchers tackled this problem through several control solutions, for example, integer and fractional order PID control [48], model predictive control [49], Smith predictor [47,50,51], communication disturbance observer [52,53], sliding mode control [54], and switching control [55].

A technique attracting an ever-growing interest for active vibration control of vibrating linear systems with time delay is pole placement, borrowed by the traditional approaches for systems without time delay [56,57]. The seminal receptance method for pole placement [58] has been extended in [59] accounting also for time delay. The extension to partial pole placement requires that a subset of the system poles is assigned while the remaining unassigned poles are kept unchanged with respect to the open-loop configuration. This problem has been tackled in [60]. The same problem has been solved using the system matrices instead of the measured receptances in [61] for the single input case and later extended for multi-input control in [62].

The papers previously quoted require evaluating a posteriori the stability of the closed- loop system. Recently, a two-stage method that embeds an a priori stability condition has been developed by Belotti and Richiedei in [63]. It relies on the powerful theory of Linear Matrix Inequalities (LMI) and ensures the placement of the dominant poles of interest while imposing stability of the remaining unassigned poles, either those due to the mechanical resonances and those induced by the time delay.

This method uses both the measured receptances to assign the dominant poles, and the system matrices, that are required by the LMIs.
Inspired by the controller parametrization proposed in the paper of Belotti and Richiedei, a method that only exploits the measured receptances has been proposed by Araujo, Dantas, and Dorea in [64]. The state feedback control gains are computed to assign the dominant poles and simultaneously impose the closed-loop system stability and robustness through the generalized Nyquist criterion. Robustness is achieved using the sensitivity function of the loop gain as an index and the problem is solved through a genetic algorithm.
In this Chapter, such a method is experimentally applied to control a flexible robot arm that mimics a typical system for pick-and-place applications. The arm flexibility is due to the passive joint torsional spring, that is an approach commonly used to represent flexibility of robots through a lumped model (see, e.g., the milestone paper in [2]). Time delays in this kind of system usually arise due to the instrumentation employed for real-time control. The proposed method is implemented by means of local linearization of the nonlinear dynamic model of the flexible robot and nonlinearities, as well as other uncertainty sources, are handled by imposing adequate robustness in the controller design.

3.2. Definitions

Let us consider a N-DOF (degree of freedom) linear, time-invariant, vibrating system. Its mass, damping, and stiffness matrices are respectively denoted by $\mathbf{M},\mathbf{C},\mathbf{K}\in\mathbb{R}^{N\times N}$ and its equations of motion are therefore

$$\mathbf{M}\ddot{\mathbf{q}}(t)+\mathbf{C}\dot{\mathbf{q}}(t)+\mathbf{K}\mathbf{q}(t)=\mathbf{B}u(t), \tag{50}$$

where $\mathbf{q}$ is the generalized displacement vector and $\dot{\mathbf{q}},\ddot{\mathbf{q}}\in\mathbb{R}^N$ its derivatives with respect to the time t. $\mathbf{B}\in\mathbb{R}^{N\times N_B}$ is the force influence matrix and $u\in\mathbb{R}$ is the independent external control force.
The rank-one control law for a regulation problem in the case of state feedback control, and by assuming that delay affects the measurements, is defined as follows:

$$u(t)=\mathbf{f}^T\dot{\mathbf{x}}(t-\tau_f)+\mathbf{g}^T\mathbf{x}(t-\tau_g), \tag{51}$$

where $\mathbf{f},\mathbf{g}\in\mathbb{R}^N$ are the velocity and displacement gain vectors and τ_f and τ_g the respective time delays. State references are omitted in Eq.(51) since do not affect the eigenstructure; their inclusion is, however, trivial.
The closed-loop controlled system in the Laplace domain denoted by s is inferred from Eq.(50), leading to

$$\left(s^2\mathbf{M}+s\left(\mathbf{C}-\mathbf{B}\mathbf{f}^Te^{-s\tau_f}\right)+\left(\mathbf{K}-\mathbf{B}\mathbf{g}^Te^{-s\tau_g}\right)\right)\mathbf{q}(s)=\mathbf{0}, \tag{52}$$

The left-hand side of Eq.(52) is the transcendental characteristic equation of the closed-loop system, $P_c(s)$, whose i^{th} solution p_i is the i^{th} closed-loop pole of the system. If $\tau_f=\tau_g=0$ then $P_c(s)$ is a polynomial and therefore the system features $2N$ eigenpairs that completely describe the system dynamics. Conversely, as studied in this work, if the time delays are not null, the characteristic equation has an infinite number of roots: $2N$ roots are the "primary roots", while an infinite number of "secondary roots" (often denoted as the "latent roots") arise [63,65].

3.3. Method description

3.3.1. Placement of the N_p closed-loop poles

In this work, the problem of robust pole placement in delayed systems with single- input control is tackled: given a set of desired $N_p < 2N$ closed-loop poles, the goal is to compute the control gain vectors $\mathbf{f}$ and $\mathbf{g}$ such that the poles are assigned at the prescribed locations and the controlled system satisfy a certain robustness condition. Additionally, it is assumed that the system matrices $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are not available and therefore the proposed method should just rely on the measured receptances. Indeed, the knowledge of the system receptances suffices to describe the system properties without the need of knowing the system matrices and therefore allows for designing the controller [58,64].
The open-loop receptance matrix of the system is defined as [58]

$$\mathbf{H}(s) = \left(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K}\right)^{-1}, \tag{53}$$

its generic pq^{th} entry $h_{pq}(s)$ is the transfer function from the force applied to the q^{th} coordinate to the displacement of the p^{th} coordinate.
The closed-loop receptance matrix is simply inferred from Eq.(52), leading to

$$\tilde{\mathbf{H}}(s) = \left(s^2\mathbf{M} + s\left(\mathbf{C} - \mathbf{B}\mathbf{f}^T e^{-s\tau_f}\right) + \left(\mathbf{K} - \mathbf{B}\mathbf{g}^T e^{-s\tau_g}\right)\right)^{-1}. \tag{54}$$

The application of the Sherman-Morrison formula [66] to Eq.(54) yields [59]

$$\tilde{\mathbf{H}}(s) = \mathbf{H}(s) + \frac{\mathbf{H}(s)\mathbf{B}\left(\mathbf{g}e^{-s\tau_g} + s\mathbf{f}e^{-s\tau_f}\right)^T \mathbf{H}(s)}{1 - \left(\mathbf{g}e^{-s\tau_g} + s\mathbf{f}e^{-s\tau_f}\right)^T \mathbf{H}(s)\mathbf{B}}. \tag{55}$$

The closed-loop poles are characterized by those complex values of s that set to zero the denominator of Eq.(55):

$$1 - \left(\mathbf{g}e^{-s\tau_g} + s\mathbf{f}e^{-s\tau_f}\right)^T \mathbf{H}(s)\mathbf{B} = 0. \tag{56}$$

The problem of finding the control gains $\mathbf{f}$ and $\mathbf{g}$ that assign the desired N_p closed-loop poles $\tilde{p}_1, \ldots, \tilde{p}_{N_p}$ can be written as follows [17,22]:

$$\begin{bmatrix} \tilde{p}_1\mathbf{r_1}^T e^{-\tilde{p}_1\tau_f} & \mathbf{r_1}^T e^{-\tilde{p}_1\tau_g} \\ \tilde{p}_2\mathbf{r_2}^T e^{-\tilde{p}_2\tau_f} & \mathbf{r_2}^T e^{-\tilde{p}_2\tau_g} \\ \vdots & \vdots \\ \tilde{p}_{N_p}\mathbf{r_{N_p}}^T e^{-\tilde{p}_{N_p}\tau_f} & \mathbf{r_{N_p}}^T e^{-\tilde{p}_{N_p}\tau_g} \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{57}$$

where $\mathbf{r}_i = \mathbf{H}(\tilde{p}_i)\mathbf{B}$, with $i = 1, \ldots, N_p$. The system can be formulated, with a more compact notation as the usual form of a linear system, $\mathbf{G}\mathbf{k} = \mathbf{y}$, with obvious meaning of the notation from Eq.(57).
In the case of the complete assignment of the closed-loop poles, i.e., $N_p = 2N$, if the matrix on the left-hand side of Eq.(57) is invertible, the solution of the linear system is: $\mathbf{k} = \mathbf{G}^{-1}\mathbf{y}$. Therefore, the solution to the complete pole placement problem is unique. Conversely, in the case of partial pole placement $N_p < 2N$ desired closed-loop poles are assigned. In this scenario the linear system in Eq.(57) has infinite solutions $\mathbf{k}$.

In particular, if $N_p < 2N$, the solution of $\mathbf{Gk} = \mathbf{y}$ is

$$\mathbf{k} = \mathbf{k_0} + \mathbf{k_h}, \tag{58}$$

where $\mathbf{k_0}$ is the particular solution of the non-homogeneous equation. While, $\mathbf{k_h}$ is the solution of the homogeneous equation $\mathbf{Gk_h} = \mathbf{0}$. Finally, the solution of Eq.(57) is more conveniently formulated as follows [63,67,68]:

$$\mathbf{k} = \mathbf{k_0} + \mathbf{V}\mathbf{k_r}, \tag{59}$$

where $\mathbf{V} \in \mathbb{R}^{2N \times (2N - N_p)}$ is a matrix whose columns span the null space of $\mathbf{G}$, i.e., $\mathbf{V} \in \text{null}(\mathbf{G})$, while $\mathbf{k_r} \in \mathbb{R}^{2N - N_p}$ is an arbitrary vector. Any choice of vector $\mathbf{k_r}$ does not perturb the assigned N_p poles. Therefore, $\mathbf{k_r}$ can be wisely chosen to accomplish secondary tasks, such as assigning other poles, stabilizing the systems or obtaining the desired robustness. In the following Section a strategy to compute $\mathbf{k_r}$ ensuring the desired robustness will be discussed, by taking advantage of the receptance-based method proposed by Araujo, Dantas, and Dorea in [64].

3.3.2. Introduction of the robustness condition

Due to the strong influence of time delays, satisfaction of Eqs. (56) and (57), does not guarantee that the set of desired closed-loop poles $\tilde{p}_1, \ldots, \tilde{p}_{N_p}$ are "primary roots". The search for a solution with stability and performance/robustness certificate must be carried out with focus on Eq.(59). Frequency domain techniques can successfully deal with rational and transcending transfer functions, including those resulting from time delay in linear systems [69,70]. In particular, the Nyquist stability criterion [71], a cornerstone of classical control theory, can be straightforwardly applied to the characteristic Eq.(56), in conjunction with the robustness margins approach [42,70] by taking the loop-gain transfer function as

$$L(s) = -\left(\mathbf{g}e^{-s\tau_g} + \mathbf{f}e^{-s\tau_f}\right)^T \mathbf{H}(s)\mathbf{B}. \tag{60}$$

A search strategy based on the maximum peak of the sensitivity function (M_s) associated with this loop-gain can offer a trade-off between robustness and performance for the closed-loop controlled system [70]. As the system parameters are usually uncertain, robustness is a significant issue in the controller design [72]. The peak M_s is related with a disk with centre in the critical abscissa of instability $(-1,0)$ and radius equal to M_s^{-1}. This disk establishes an acceptable distance from the Nyquist curve of $L(s)$ from the point $(-1,0)$. The Nyquist criterion states that, for an open-loop system with all poles on the left half-plane, the closed-loop system will be stable if the Nyquist curve of $L(s)$ does not encircle the point $(-1,0)$. The larger the disk radius, the more robust the system is with respect to perturbations on the nominal loop-gain $L(s)$. Then, the design problem can be formulated as that of computing the feedback gains of the parametrized family, $\mathbf{k_r}$ in Eq.(56), with a simultaneous guarantee that the Nyquist curve of the loop-gain lies in a safe distance of M_s^{-1} to the critical point. This problem can be approached through the following minimization formulation:

$$\min_{k_r} \ \min_{\omega_i} \left(\left| -\left(\mathbf{g} e^{-j\omega_i \tau_g} + j\omega_i \mathbf{f} e^{-j\omega_i \tau_f} \right)^T \mathbf{H}(j\omega_i)\mathbf{B} + 1 \right| - M_s^{-1} \right)^2$$

$$\text{s.t.} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} = \mathbf{k_0} + \mathbf{V}\mathbf{k_r} \tag{61}$$

$$\text{Re}\left\{ -\left(\mathbf{g} e^{-j\omega_i \tau_g} + j\omega_i \mathbf{f} e^{-j\omega_i \tau_f} \right)^T \mathbf{H}(j\omega_i)\mathbf{B} \right\} \geq -1 + M_s^{-1} \ \forall \ \omega_i : \text{Im}\{L(j\omega_i)\} = 0.$$

The second constraint in Eq.(61) assures that every cross point on the real axis lies on the right of the M_s disk, avoiding then encirclement of the critical point $(-1,0)$. Notice that the frequency ω_i must belong to a frequency range $[\omega_{\min}, \omega_{\max}]$, suitable to a representative Nyquist plot. This range can be, as an instance, the same used in the experimental identification of the system receptance. Moreover, for underactuated systems, only the partial information $\mathbf{H}(j\omega)\mathbf{B}$ must be known to compute the gains.

3.3.3. Numerical implementations details

The objective function and the stability constraint in Eq.(61) are hard to approach with gradient-based methods. It is well known that genetic algorithm-based search (GA) can be more accurate to solve optimization problems of moderate complexity, as non- convex ones [73]. A GA implementation was developed, tailored to find a solution for Eq.(61).

Given the particular solution $\mathbf{k_0}$ and the null space basis $\mathbf{V}$, the solution for the optimization problem in Eq.(61) is achieved by following some simple steps. First, randomly define a set of $\mathbf{k_r}$ vectors. Then, evaluate each one individual in this set for fitness and constraints. The procedure is summarized in the flowchart displayed in Figure 11. In the step devoted to update the population, functions to execute crossover and mutation can be chosen and adjusted following the theory of Genetic Algorithms [74]. In the step of individuals combination, first, it is ensured that the best rated individual in the actual generation is saved for the next one. Furthermore, it is selected some possible solutions to gives rise to a new population. Those are called parents, and they are chosen in a draw where the best-rated individuals have a greater chance of being selected. Once the parents are chosen, the combination is done, and this could be achieved using any kind of crossover methods available in the GA theory. Finally, a percentage of individuals is slightly randomly modified in the mutation process. In the test cases of the following Sections, GA was programmed for (i) a maximum number of generations of one-hundred and (ii) a fitness function tolerance of 10^{-12}.

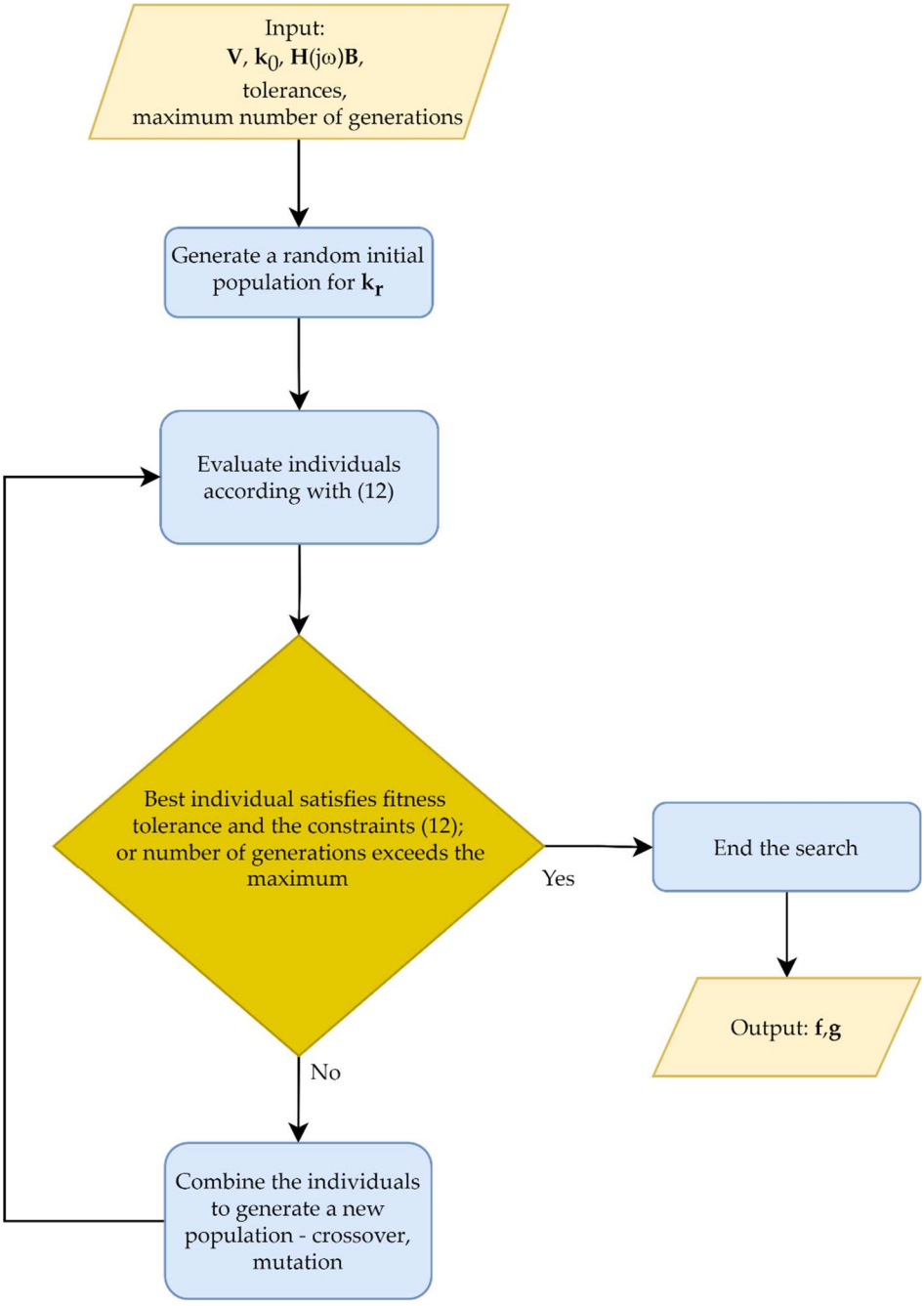


Figure 11. The flowchart of the numerical procedure to feedback gains design.

3.4. Test case: a two-link flexible robot arm

3.4.1. System description

The proposed control strategy is applied to the two-link flexible robot arm shown in Figure 1, already employed in [36] as a benchmark for vibration control. The meaning of all the relevant system parameters is the same that has been given in the previous Chapter and, therefore, the values of all the parameters of interest are the ones defined in Table 1.
The link rotations are measured through two incremental encoders, with 500 pulses per revolute. This low resolution perturbs the sensed displacements and the estimated speeds, that are obtained

through numerical derivatives, and therefore attention to the robustness issue should be paid in the controller design.

The nonlinear model of the system is obtained through the Lagrangian approach, leading to the following equation of motion (g is the gravitational acceleration):

$$\begin{bmatrix} J_m + J_d + J_1 + m_{2,eq} l_1^2 & \frac{1}{2} m_2 l_1 l_2 \cos(\theta_1 - \theta_2) \\ \frac{1}{2} m_2 l_1 l_2 \cos(\theta_1 - \theta_2) & J_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} m_2 l_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2^2 \\ -\frac{1}{2} m_2 l_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta}_1^2 \end{bmatrix}$$
$$+ \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{2} m_1 + m_{2,eq}\right) g l_1 \sin\theta_1 \\ \frac{1}{2} m_2 g l_2 \sin\theta_2 \end{bmatrix} + \begin{bmatrix} k_s & -k_s \\ -k_s & k_s \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} T_m, \tag{62}$$

In the presence of small displacements about an equilibrium configuration, the system in Eq.(62) can be locally linearized to apply the theory of linear control. Therefore, local receptances can be experimentally identified to apply the theory proposed in this Chapter. To show the features of the linearized model, although it is not used by the receptance-based approach here proposed to controller design, the linearized model around the vertical equilibrium position, i.e., $\theta_1 = \theta_2 = 0$, is described by the mass matrix $\mathbf{M_e}$, the damping matrix $\mathbf{D_e}$, the stiffness matrix $\mathbf{K_e}$ and the input distribution matrix $\mathbf{P}$ [36], whose expressions have been reported in the previous Chapter through Eqs. (45), (46), (47) and (48).

The deviations between the nonlinear model and the linearized one, and thus between the nonlinear model and the local receptances, that arise with finite link rotations about the equilibrium, should be considered as uncertainty sources that the controller should get rid through adequate robustness. Other perturbations affect the controller design: the aforementioned low resolution of the encoders, errors, and uncertainty in the receptance measurements, non-viscous friction terms, and some unpredictable dynamics that will be discussed in Section 3.4.2.1.

In all the tests, both numerical and experimental, it is assumed that $\tau_f = 0.04$ s and $\tau_g = 0.05$ s. These severe values are chosen by means of example to highlight some benefits of the proposed method. The delays have been introduced in the real-time control scheme, developed through Real Time Simulink (running with a sample time $T_s = 1$ ms).

Two different assignment tasks are performed to highlight different features of the proposed method and then are experimentally applied to provide further evidences.

3.4.2. Numerical assessment

3.4.2.1. Test case 1: increasing damping of the second vibrational mode

The state-feedback controller is designed such that the second pole pair is assigned to $-5.000 \pm 38.177\,j$, i.e., damping is increased to $\xi_{3,4} = 0.130$ while the damped natural frequency is kept unchanged. The maximum peak of the sensitivity function is imposed to $M_s = 1.75$ in order to ensure adequate robustness.

Pole placement is performed by computing the control gains $\mathbf{k_0}$ through the receptance method proposed in Section 3.3.1, leading to $\mathbf{f_0} = \begin{bmatrix} -0.1346 & -0.0129 \end{bmatrix}^T$ and $\mathbf{g_0} = \begin{bmatrix} 0.0060 & -0.0422 \end{bmatrix}^T$. The closed-loop poles are computed through the collection of Matlab functions released under the package DDE-Biftool [75] and are shown in Figure 12. The dominant closed-loop poles are summarized in

Table 3, while the latent roots are omitted for brevity. The desired closed-loop poles are correctly assigned. Conversely, the closed-loop system Nyquist curve for the Loop Gain function $L(j\omega)$ shown in Figure 13 highlights that $\mathbf{k_0}$ does not satisfy the prescribed robustness condition, since the Nyquist curve belongs to the M_s circle.

Table 3. Test case 1: open-loop and closed-loop poles.

Open-Loop Poles	Desired Poles	Poles with k_0	Poles with k
$-1.197 \pm 7.122\,j$	–	$-2.461 \pm 36.167\,j$	$-13.635 \pm 11.480\,j$
$-1.037 \pm 38.177\,j$	$-5.000 \pm 38.177\,j$	$-5.000 \pm 38.177\,j$	$-5.000 \pm 38.177\,j$

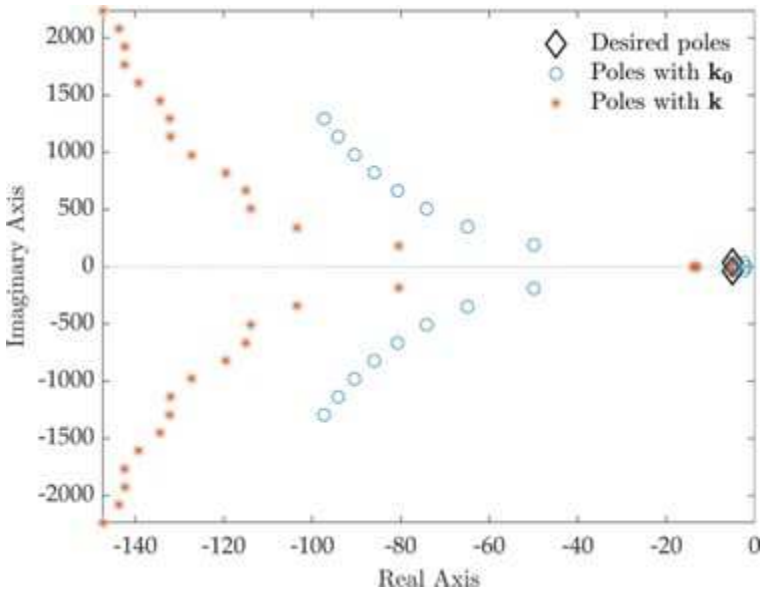


Figure 12. Test case 1: closed-loop poles.

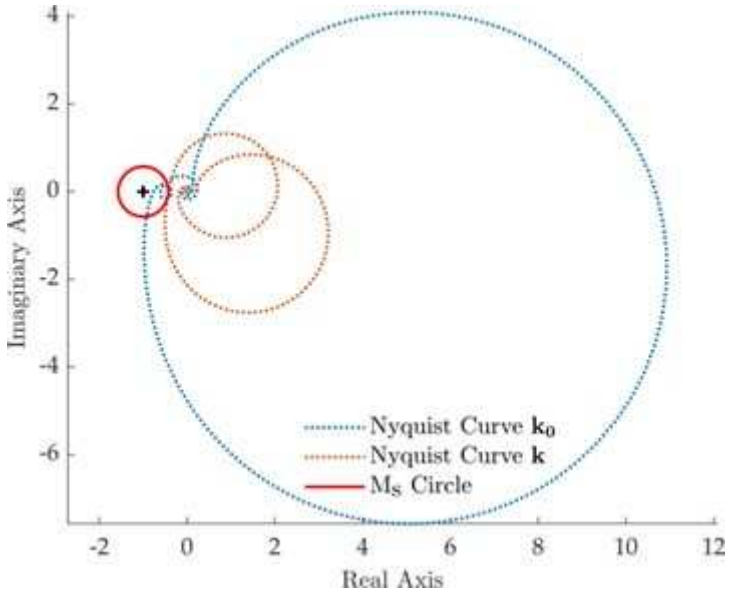


Figure 13. Test case 1: Nyquist curve of $L(j\omega)$.

The prescribed robustness is, in contrast, achieved by the method described in Section 3.3.2, that has led to the gain vector $\mathbf{k} = \mathbf{k_0} + \mathbf{V}\mathbf{k_r}$ with $\mathbf{f} = \begin{bmatrix} -0.0388 & -0.0071 \end{bmatrix}^T$ and $\mathbf{g} = \begin{bmatrix} 0.2311 & -0.3180 \end{bmatrix}^T$. The resulting closed-loop system features a Nyquist curve that does not belong to the M_s circle for any complex value, as evidenced by Figure 13, and the desired poles are correctly assigned.

Note that neglecting the time delay in the synthesis of the controller, i.e., setting $\tau_f = \tau_g = 0$ in Eq.(57) and requiring the same two pairs of the primary closed-loop poles obtained through $\mathbf{k}$, leads to a control gain vector that destabilize the closed-loop system if these gains are applied to the time delayed system, due to the primary roots crossing the imaginary axis.

3.4.2.2. Test case 2: improving the speed of response

In the second example, the state feedback controller is tuned to modify the poles associated to the first vibrational mode. It is wanted to simultaneously increase its damping and its natural frequency, and hence the absolute values of both the real and the imaginary parts, to speed up the transient response. By means of example, the pair of poles is assigned to $-10 \pm 20\,j$. Robustness is obtained by imposing $M_s = 2$.

The pole placement task is correctly satisfied through $\mathbf{k_0}$, with $\mathbf{f_0} = \begin{bmatrix} -0.0883 & 0.0114 \end{bmatrix}^T$ and $\mathbf{g_0} = \begin{bmatrix} -0.0011 & -0.0098 \end{bmatrix}^T$, obtained by solving Eq.(57) through the *pinv* function. In this case, as corroborated by the closed-loop poles listed in Table 4 and shown in Figure 14, the closed-loop system is unstable due to the spillover on the second pole pair that lies in the right-hand half of the complex plane.

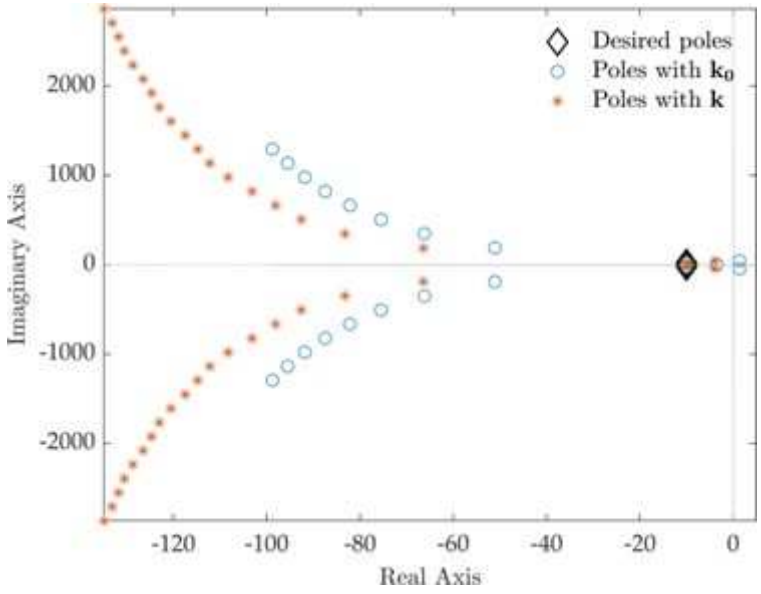


Figure 14. Test case 2: closed-loop poles.

Table 4. Test case 2: open-loop and closed-loop poles.

Open-Loop Poles	Desired Poles	Poles with k_0	Poles with k
$-1.197 \pm 7.122\,j$	$-10 \pm 20\,j$	$-10 \pm 20\,j$	$-10 \pm 20\,j$
$-1.037 \pm 38.177\,j$	–	$-1.370 \pm 45.847\,j$	$-3.930 \pm 40.452\,j$

The method proposed in this work tackles this problem by leading to $\mathbf{k}$, with $\mathbf{f}=\begin{bmatrix}-0.0641 & -0.0049\end{bmatrix}^T$ and $\mathbf{g}=\begin{bmatrix}0.0041 & -0.2466\end{bmatrix}^T$, that simultaneously satisfies the pole placement requirement and stabilizes the closed-loop system additionally ensuring the required robustness. The Nyquist curves of the loop gain $L(j\omega)$ of the closed-loop systems with $\mathbf{k_0}$ and $\mathbf{k}$ are shown in Figure 15 and clearly prove these aspects.

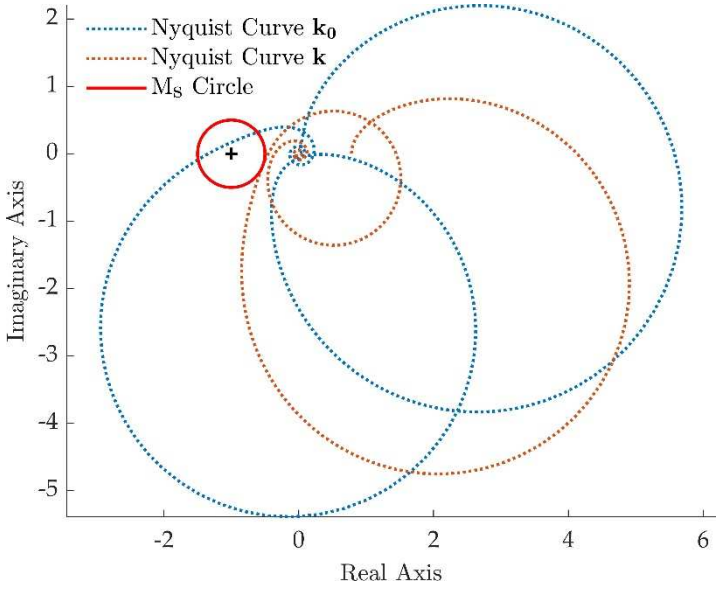


Figure 15. Test case 2: Nyquist curve of $L(j\omega)$.

3.4.3. Experimental application

3.4.3.1. Application of the controller of test case 1

The controller proposed in Section 3.4.2.1 is applied to the experimental two-link flexible robot arm shown in Figure 1. An impulse-like torque disturbance is applied through the DC motor, the experimental responses of the open-loop and closed-loop system with $\mathbf{k}$ are shown in Figure 16 through the relative angle of the second link $\theta_2^r=\theta_2-\theta_1$.

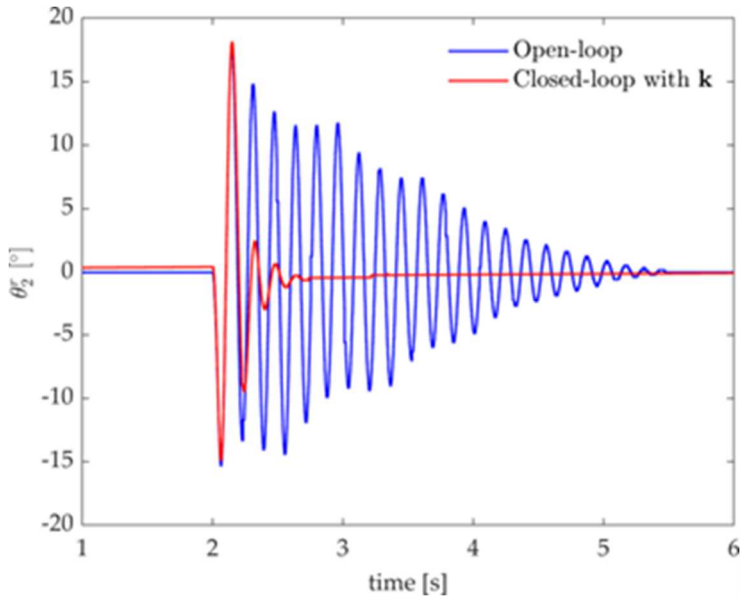


Figure 16. Test case 1: experimental impulse responses, relative angle of the second-link.

The modal parameters of the open-loop and closed-loop system are identified through the *modalfit* function and compared with the expected numerical counterpart in Table 5, where f_n denotes the natural frequency and ξ_n the related damping ratio. Additionally, the expected and experimental inertances for the open-loop and closed-loop system are shown in Figure 17. The fulfilment of the assignment task is evidenced by the agreement between the numerical data and the experimental measurements.

Table 5. Test case 1: numerical and experimental modal parameters.

Open-Loop	**Mode 1**		**Mode 2**	
Modal Parameter	**Numerical**	**Experimental**	**Numerical**	**Experimental**
$f_n\,[\mathrm{Hz}]$	1.15	1.29	6.08	6.16
ξ_n	0.166	0.161	0.027	0.025
Closed-Loop				
$f_n\,[\mathrm{Hz}]$	2.84	3.08	6.13	6.14
ξ_n	0.770	0.749	0.130	0.190

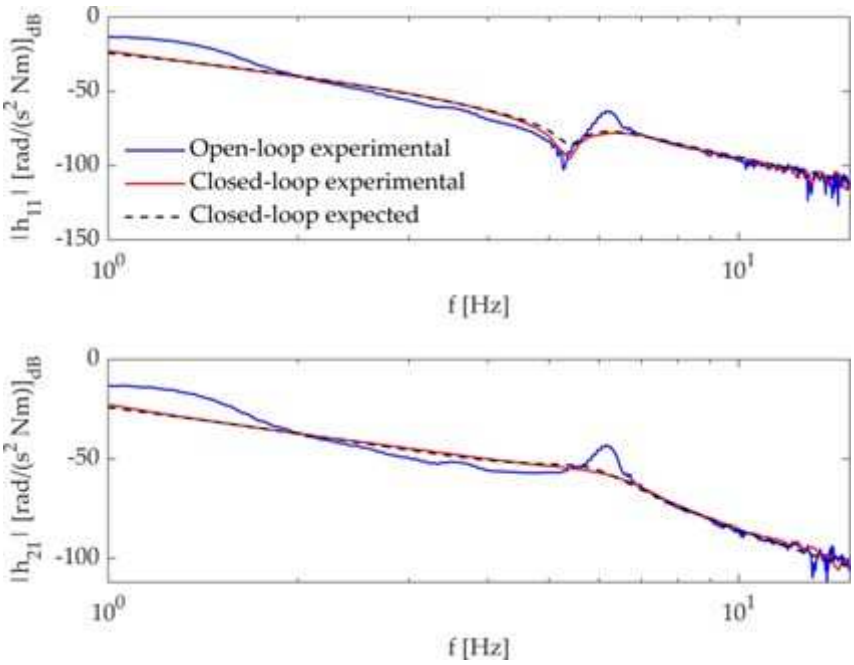


Figure 17. Test case 1: expected and experimental inertances.

The agreement between the numerical expectations and the actual experimental results corroborates the controller robustness and its effectiveness to handle perturbations. Besides the linearization and the unavoidable, although small, errors on the measured receptances, the system is affected by the warp of the wire of encoder B that slightly perturbs the system mass distribution during the motion and also the equilibrium configuration. This effect is evident in Figure 18 where a train of pulses with different amplitudes is applied to obtain different nonlinear effects due to the geometrical non-linearity of the system (see Eq.(62)). The motion of the wire causes slightly different equilibrium configurations once the response settles. Nonetheless, the controller ensures stable, fast, and repeatable vibration control, thus proving robustness.

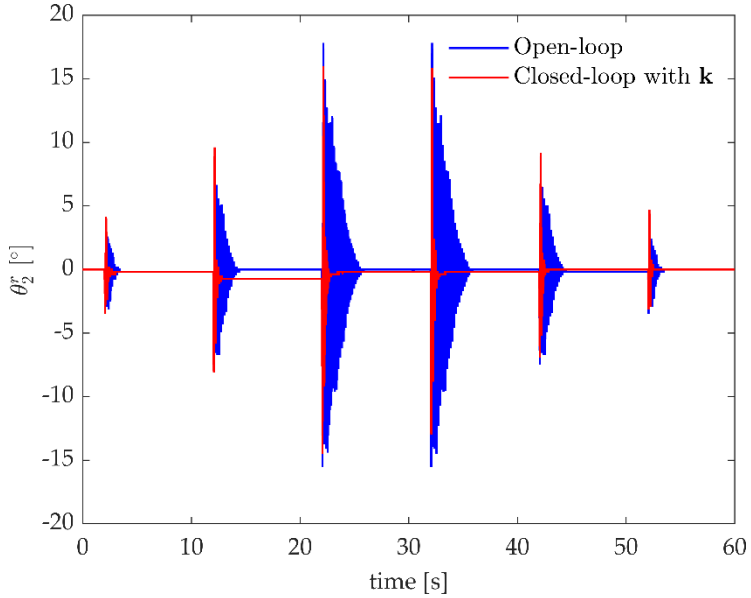


Figure 18. Test case 1: experimental response to a train of impulses.

If no adequate robustness in obtained, the nominally stabilizing controller might result unstable or, at least, remarkably downgrade the actual performances. The robustness of the controller synthesized

through the proposed method, i.e., the one with gains $\mathbf{k}$, is even more evident if it is compared with the non-robust controller tuned through $\mathbf{k}_0$. The experimental impulsive response of $\mathbf{k}_0$ is shown in Figure 19.

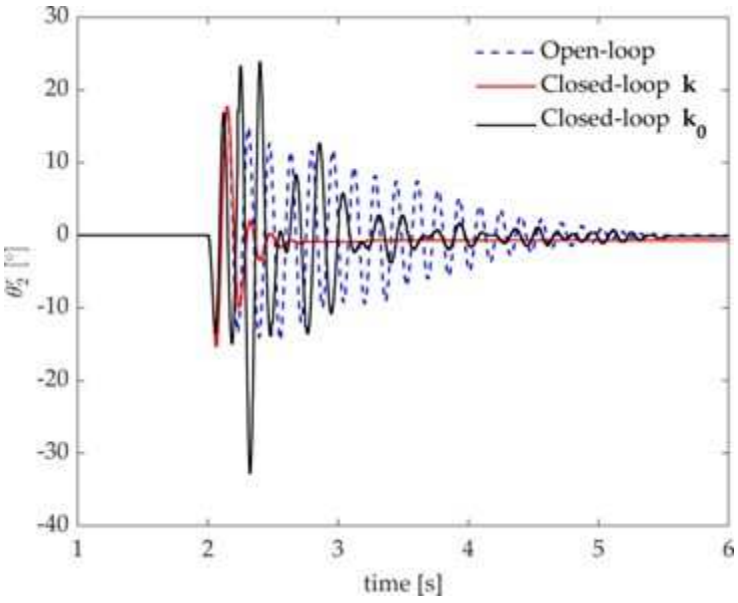


Figure 19. Test case 1: experimental impulse response with the non-robust controller $\mathbf{k}_0$.

The numerical and experimental responses of the open-loop and closed-loop systems with $\mathbf{k}$ and $\mathbf{k}_0$ are compared in Figure 20. These provide a satisfying agreement for the open-loop and closed-loop system with $\mathbf{k}$, while in case of $\mathbf{k}_0$ the experimental results do not match with the expected numerical results. It is evident that improving robustness is mandatory to enable for practical applications with adequate and reliable performances.

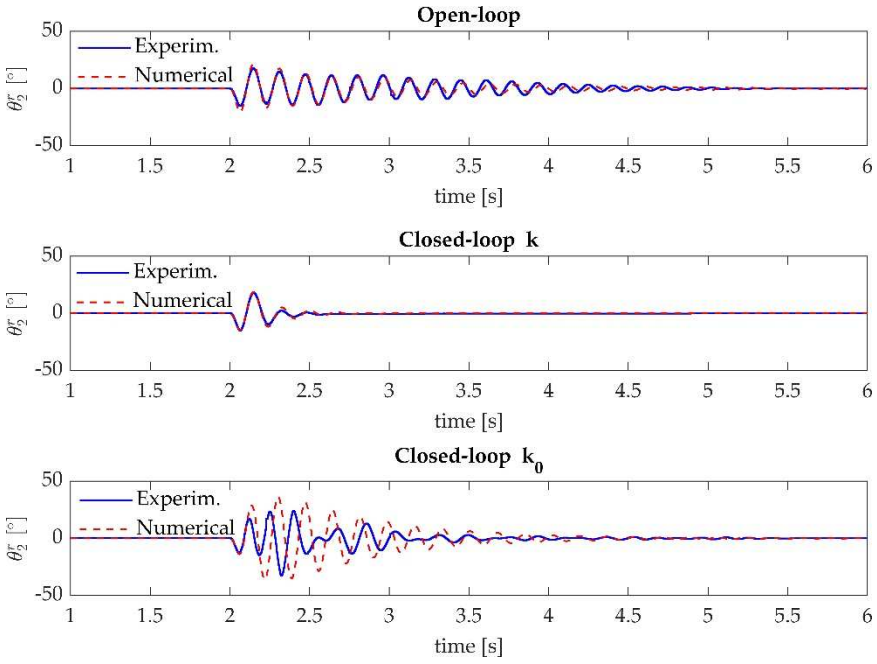


Figure 20. Test case 1: comparison of the numerical and experimental open-loop and closed-loop impulse responses.

Last, Figure 21 shows the application of the controller to a step-tracking problem, that represents a typical pick-and-place task [76,77]. A sequence of step references to move the end-effector to the

target value $x_{ee} = 0.1$ m is applied. The state-feedback controller is supported by a feedforward controller to compensate for gravity and friction and ensure no (or negligible) steady state error. Such feedforward control, that is the same for both the open and closed-loop schemes, consists of a square-wave, and does not affect the assigned closed-loop system eigenstructure. The comparison of the time responses, by denoting T_r the $10-90\%$ rise time and the 2% settling time is T_s, shows that the controlled system fulfils the theoretical expectations. A faster response is exhibited ($T_r = 0.186$ s and $T_s = 0.476$ s) with respect to the open loop system ($T_r = 0.125$ s and $T_s = 1.490$ s), with just a negligible overshoot (4%, compared to the 75% of the open loop system).

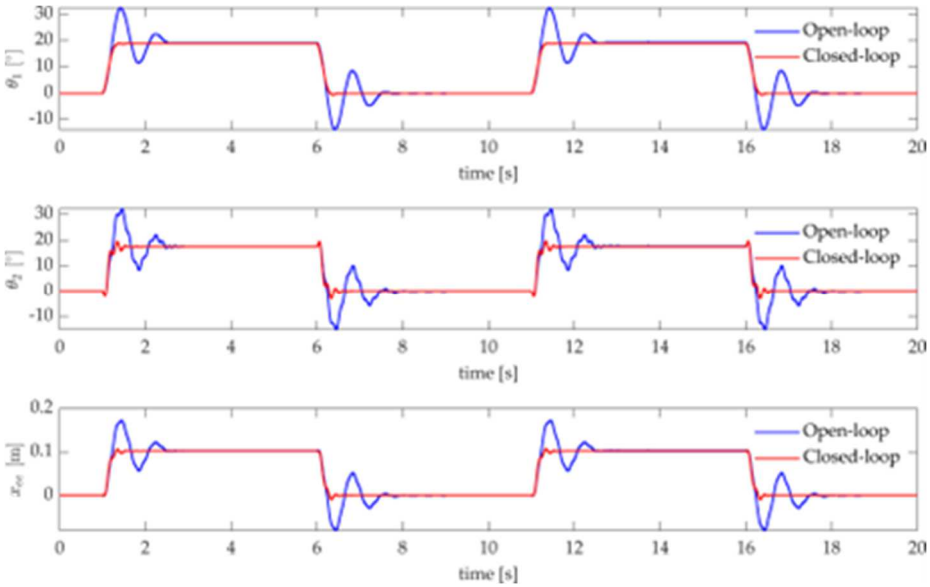


Figure 21. Test case 1: experimental pick-and-place application.

3.4.3.2. Application of the controller of test case 2

The controller designed in Section 3.4.2.2 is experimentally tested to assess its performances and robustness. The response to an impulse-like excitation exert by the DC motor is tested for the open-loop and the closed-loop systems with $\mathbf{k_0}$ and $\mathbf{k}$. Figure 22 confirms that the controller tuned by solving Eq.(57) (i.e., $\mathbf{k_0}$) does not ensure asymptotic stability. Although a divergent behaviour in the case of $\mathbf{k_0}$ was expected, θ_2^r oscillates between two bounded values due to the saturation of the enforceable actuator torque (0.4 Nm). In contrast, $\mathbf{k}$ is stable, as expected, and ensures quick settling of the unwanted vibration. The natural frequency and damping of the closed-loop primary poles obtained with $\mathbf{k}$ are compared with the expected numerical results in Table 6, and corroborate the correct fulfilment of the assignment task.

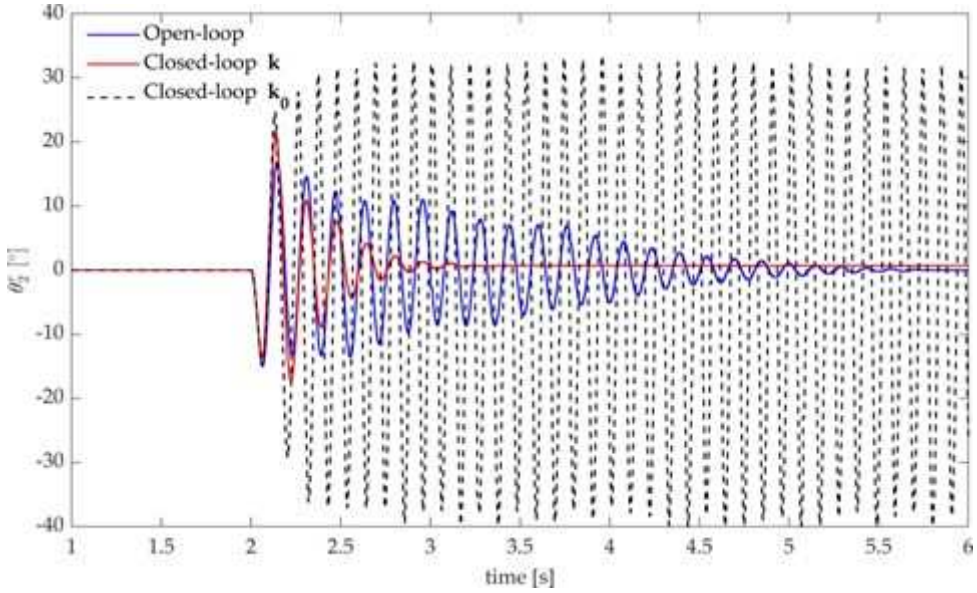


Figure 22. Test case 2: experimental impulse response.

Table 6. Test case 2: closed-loop numerical and experimental modal parameters.

	Mode 1		Mode 2	
Modal Parameter	**Numerical**	**Experimental**	**Numerical**	**Experimental**
$f_n\,[\mathrm{Hz}]$	3.55	3.80	6.48	6.47
ξ_n	0.440	0.375	0.097	0.093

The same controller is applied in the pick-and-place application discussed in Section 3.4.3.2. The experimental open-loop and closed-loop responses of the system are shown in Figure 23. The benefits of speeding up the first mode as done in test-case 2 (see Section 3.4.2.2) are evident. The second controller enables to simultaneously reduce the rise time and the settling time ($T_r = 0.094$ s and $T_s = 0.458$ s), with respect to both the open-loop and closed-loop system when the controller tuned to tackle the second mode is employed (see Sections 3.4.2.1 and 3.4.3.2).

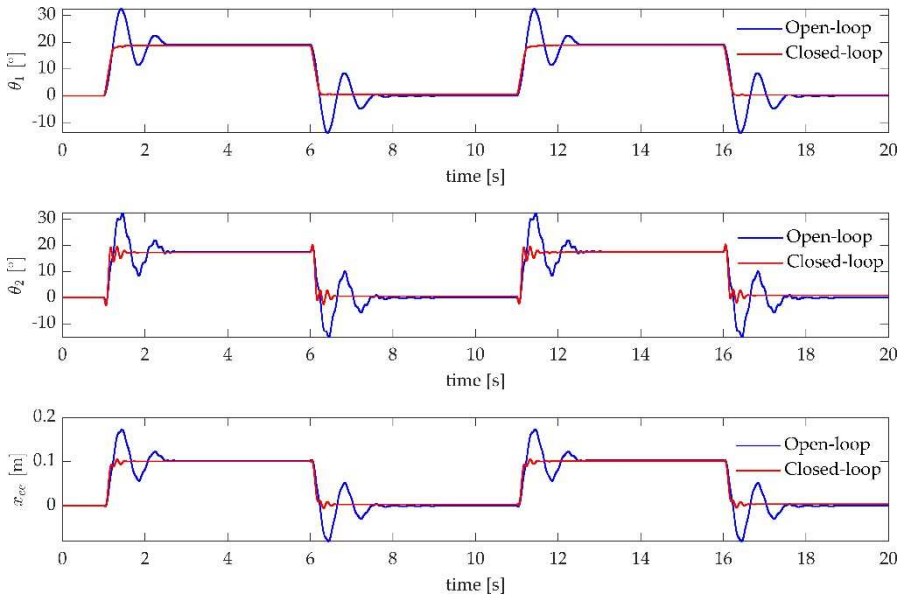


Figure 23. Test case 2: experimental pick-and-place application.

3.5. Conclusions

This Chapter provides the experimental application of the method proposed by Araujo, Dantas, and Dorea for pole placement in flexible linear systems with time delay. The method exploits state feedback control to perform the partial pole placement of the desired system poles. The degrees of freedom in the choice of the control gains assigning such dominant poles is the leverage to stabilize all the remaining poles, including the infinite number of secondary roots due to the time delay, and to achieve the desired robustness of the closed-loop system. Robustness is quantified through the sensitivity function of the loop gain transfer function. The proposed technique exploits only the measured receptances, i.e., the system matrices $\mathbf{M}_e, \mathbf{C}_e$ and $\mathbf{K}_e$ are not needed to design the controller.

The effectiveness and usefulness of the proposed method is experimentally assessed against a challenging nonlinear and uncertain two-link flexible robot arm, whose flexibility arises due to the passive joint. Two controllers are tuned and experimentally applied to show different features of the method. The effectiveness of both controllers is evaluated by applying impulse disturbances as well a square wave reference (i.e., a sequence of steps) mimicking a pick-and-place robotic application. In both cases, the experimental application of the controllers provides excellent results and agreement with the expected numerical results. Besides effectively handling the severe time delays assumed, the imposed robustness allows the controller to get rid of the unavoidable uncertainties, e.g., due to the warp of the encoder wire, to the coarse encoder quantization, and to the nonlinear terms neglected in the controller design.

Due to the effectiveness of the proposed approach together with its simplicity, as a consequence of the use of the experimental receptances without the need of accurate system model, it is well suited for more complicated delayed systems such as mechatronic systems employed for manufacturing processes (see, e.g., in [45]), as well as robotic systems performing remote-operations [78] or teleoperations [79].

4. Collocated Model Predictive Control with Embedded Integrator for Path Tracking in Cable Suspended Parallel Robots with Flexible Cables

This Chapter proposes a novel control scheme for precise path tracking control in kinematically-determined, non-redundant Cable Suspended Parallel Robots (CSPR). To handle model nonlinearities while ensuring small computational effort, a controller made by two sequential control actions is developed. The first term is a position-dependent, collocated Model Predictive Control (MPC) with Embedded Integrator (EI), to compute the optimal cable tensions ensuring accurate path tracking and fulfilling the feasibility constraints. The second control term transforms the optimal tensions into the commanded motor torques, and hence currents, that are evaluated through the kinetostatic model of the electric motors used for winding and unwinding the cables. Control design is performed through the robot dynamics model, formulated with the assumption of rigid cables. While the imposition of positive tensions is embedded and treated as hard constraints in the controller design, reduced excitation of the load oscillations is achieved by penalizing large tension variations in the cost function adopted in the controller design, plus some hard constraints on the maximum tension derivatives. These features, together with the embedding of the integrator within the MPC formulation, ensure smooth control tensions that allows handling the axial flexibility of the cables, although it is not explicitly considered in the controller design.
To assess the performances of the proposed control algorithm, a kinematically-determined CSPR with lumped end-effector is simulated, by also adopting very flexible cables. Additionally, a simplified dynamic model of the electrical dynamics and the sensor quantization are included to provide a realistic representation of the real environments. The results, together with the fair comparison with a benchmark, corroborate the effectiveness of the proposed approach, its robustness and its feasibility in real-time controllers due to the wise reduction of the computational effort.

4.1. Introduction

4.1.1. State-of-the-art and Chapter motivations

Cable Driven Parallel Robots (CDPR) have been attracting an even greater attention of researchers in the fields of robotics and multibody system dynamics, because of their benefits, that include large workspaces and large payloads, which result in a lightweight and cheap design [80–82]. On the other hand, motion planning control of such a robot is challenging since cables can pull but not push the end-effector, leading to a unilateral actuation (i.e., just tensile cable forces can be exerted). This limitation is sometimes overcome at the design stage through cable redundancy or by placing cables under the moving end-effector, thus resulting in a more expensive construction. A more ambitious approach is developing Cable Suspended Parallel Robots (CSPRs), i.e., when all cables lie above the moving end-effector, without redundant actuation forces: this is the occurrence investigated in this Chapter.
In non-redundant CSPRs, ensuring positive cable tensions along the path relies therefore on advanced motion planning and control techniques. Optimal motion planning has been widely investigated, by designing trajectories that a-priori ensure positive and bounded cable tensions, in an open-loop manner, by exploiting the robot dynamic model and in the presence of some boundary conditions (e.g. [80,83] and the references therein). As far as closed-loop control is concerned, standard industrial controllers, such as PID, have been mainly applied to this kind of robotic systems in the last decades ([84,85]). The drawback of such standard controllers is that that the positiveness constraints on the cable tensions, as well as bounds on the feasible maximum tensions, cannot be embedded in the control design and, hence, a-posteriori verification or control saturation should be adopted. However, a-posteriori saturating in the presence of integral control is often ineffective and

might lead to relevant overshoot, due to windup. Thus, these controllers are effective in the presence of optimized trajectories.
The recent advances in the field of motion control of multibody systems suggest that CSPRs can benefit of some advanced control techniques to improve their performances. In this work, precise path tracking control in a CSPR is solved by exploiting and extending the idea of Model Predictive Control (MPC). Indeed, among the several model-based control techniques used for multibody systems, MPC has several features that make it very attractive for Cable-Driven Parallel Robots (CDPRs) in general and, in particular, for the more challenging case of CSPRs.
The underlying idea of MPC is to compute the optimal control forces by solving an optimal control problem, defined through a cost function. The optimization problem over the future control variables is solved at each time step by predicting the future system states and outputs over a receding horizon. Hence, MPC provides an optimal sequence of the control input, in accordance with some metrics. Prediction is performed through a first-order representation of the dynamic model and bounds on the control variables can be embedded in the optimization problem. These features make MPC attractive for motion control of multibody systems (see e.g. [17,86,87]), and in particular to CDPRs: constraints on input and output variables can be effectively included without requiring a-posteriori saturations.
In the very last years, just a few researchers in the field of CDPRs have applied concepts of MPC ([88–91]); as a matter of fact, the greatest efforts have been devoted to ensure positive tensions through optimal motion planning (see e.g. [80,92,93] and the references therein). The idea of MPC has been also adopted to plan dynamic transition trajectories for a fully-actuated cable suspended robot in [27].

4.1.2. Chapter contributions

In this work, a novel MPC architecture is proposed for precise path tracking control, i.e., for contour error control, in kinematically-determined, non-redundant CDPRs. The controller is based on two terms, to allow for a simpler implementation and for handling the nonlinearities of the dynamic model in an effective way. The control scheme is collocated, since it just uses the measurement on the motor side, i.e., positions and estimated speeds of the motors used for winding and unwinding the cables driving the end-effector, and the controller is designed through the dynamic model of the considered CSPR, by assuming that cables are perfectly stiff. Smoothness of the control action is included by means of a suitable cost function, and the effectiveness of the proposed control scheme to handle axially-flexible cables is provided by using it to control a flexible CSPR. To improve the path tracking capability, without increasing excessively the control gains, that would result in large oscillations (especially in the presence of flexible cables), integrators are embedded in the MPC design, leading to an MPC with Embedded Integrator (MPC-EI), that exploits the formulation of the dynamic model through the “difference variables”. Numerical assessment of the control performances is made through a fully actuated three-DOF cable-suspended robot (i.e., with a lumped end-effector), controlled by three cables with a low axial stiffness. The simulated system is therefore underactuated, although controller design assumes a fully actuated configuration. The effect of the actuator and sensor dynamics are included as well, to obtain a reliable validation of the proposed control scheme.
The idea of designing controllers for flexible systems by means of just simplified model of elasticity is often a need in the case of complicated or industrial systems, or if computational effort of controller should be kept small, and some recent papers in the field of controlling multibody systems successfully address this issue (see e.g. [82,94]). This idea of control, that do not exploit a model of the cable flexibility, recalls a dichotomy in the field of motion planning of vibrating systems: oscillation reduction can be addressed either through model-based approaches, that exploits the dynamic model of the flexible system (see e.g. [95–97]), or by model-free approaches that wisely reduce jerk or snap (see e.g. [98,99]) without considering the mathematical model of the oscillations.

Another contribution of this work lies in the use of a novel formulation, inspired by the penalty method of multibody systems [100], to model the cable axial flexibility in a straightforward and effective way, leading to a model representation that is easy to handle and to integrate in numerical simulations.

4.2. Dynamic model

4.2.1. Studied system

The CSPR investigated in this work is represented in Figure 24. It consists of an end-effector made by a lumped mass, that is driven through three cables (and hence three actuators), thus resulting in a kinematically-determined, non-redundant configuration. This is a less common architecture [27], since redundant architectures are more often adopted to simplify motion planning and control (see e.g., [101,102]). The proposed control scheme can be extended to other kinematically-determined, non-redundant CDPRs, once the system dynamics is properly modelled.

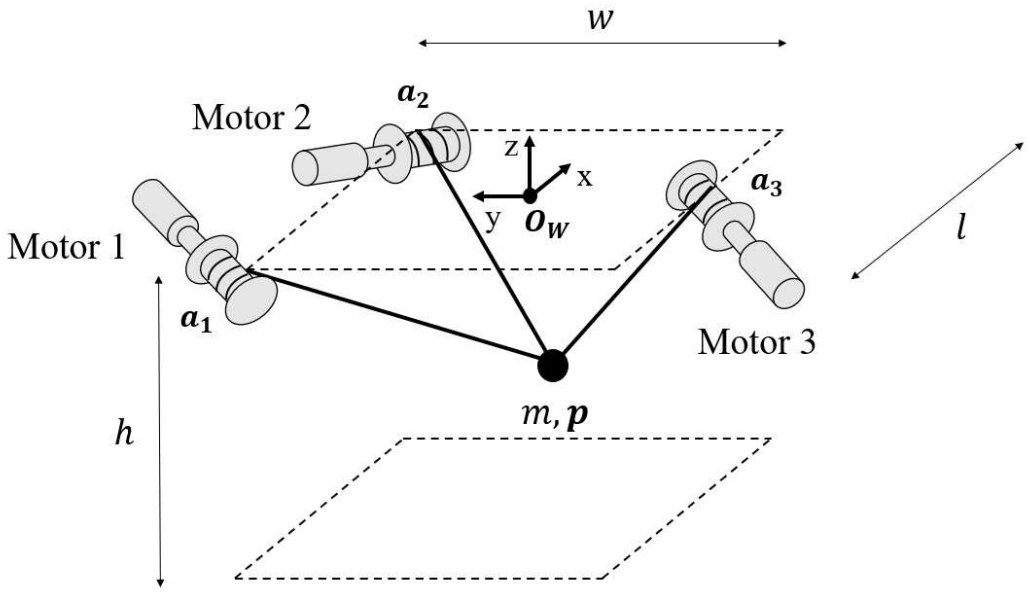


Figure 24. Scheme of the CSPR under investigation.

4.2.2. Dynamic model of the CSPR with stiff cables

The dynamic model can be inferred by adopting the usual formalism of multibody systems, by defining the vector of the 6 dependent coordinates $\mathbf{q} = \begin{bmatrix} \mathbf{p}^T & \boldsymbol{\theta}^T \end{bmatrix}^T \in \mathbb{R}^6$ where $\mathbf{p} = [x \quad y \quad z]^T \in \mathbb{R}^3$ is the vector collecting the end-effector absolute position, and $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \in \mathbb{R}^3$ collects the rotations of the three motors:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{J}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{f} \tag{63}$$

The mass matrix is defined as $\mathbf{M} = diag\left(m,\, m,\, m,\, J_1,\, J_2,\, J_3\right) \in \mathbb{R}^{6\times 6}$, where $m \in \mathbb{R}$ is the mass of the end-effector and $J_i \in \mathbb{R}$ $\left(i = 1,2,3\right)$ is the equivalent moment of inertia of each motor (that includes the contributions of the rotor, the shaft, and the drum). The external forces are collected into vector $\mathbf{f} \in \mathbb{R}^6$, that includes the gravity force acting in the z-direction, the torques exerted by the motors $\boldsymbol{\tau}_{\mathbf{m}} = \begin{bmatrix} \tau_{m1} & \tau_{m2} & \tau_{m3} \end{bmatrix}^T \in \mathbb{R}^3$ and friction forces on the motor shafts:

$$\mathbf{f} = \begin{bmatrix} m\mathbf{g} \\ \boldsymbol{\tau}_m - \mathbf{F}_v\dot{\boldsymbol{\theta}} \end{bmatrix} \tag{64}$$

where $\mathbf{g} = [0 \quad 0 \quad -g]^T \in \mathbb{R}^3$ is the vector of gravity acceleration, while $\mathbf{F}_v = diag\left(f_{v1}, f_{v2}, f_{v3}\right) \in \mathbb{R}^{3\times3}$ is the diagonal matrix containing the motor viscous friction coefficients. Finally, $\mathbf{J} \in \mathbb{R}^{3\times6}$ is the Jacobian matrix of the position kinematic constraint equations, while $\boldsymbol{\lambda} \in \mathbb{R}^3$ collects the three Lagrange multipliers.

The kinematic constraints are obtained by assuming that cables are always taut and not slack, and hence can be treated as holonomic constraints. The fulfilment of this assumption will be enforced in the controller design, by properly setting positiveness constraints on the cable tensions. Additionally, in the controller design, it is assumed that cables are stiff. This assumption will be removed in the development of the simulator adopted to assess the proposed control strategy, where flexible cables will be assumed. Let us consider the length of the i^{th} cable, henceforth denoted as $l_i \in \mathbb{R}^+$, that is wound on a pulley whose radius is r_i:

$$l_i = l_{i0} - r_i\theta_i \tag{65}$$

($l_{i0} \in \mathbb{R}$ is the length of the cable when $\theta_i = 0$). On the other hand, l_i can be expressed as the distance between the (fixed) i^{th} exit point (denoted as $\mathbf{a}_i$ in Figure 24 and whose absolute positions are collected into vector $\mathbf{a}_i \in \mathbb{R}^3$) and the end-effector position $\mathbf{p}$:

$$l_i = \left\|\mathbf{p} - \mathbf{a}_i\right\| \tag{66}$$

Eq. (65) and Eq. (66) lead to the i^{th} constraint equation representing the i^{th} cable, that resembles the one of a holonomic and scleronomic constraint:

$$\sqrt{\left(l_{i0} - r_i\theta_i\right)^2 - \left\|\mathbf{p} - \mathbf{a}_i\right\|^2} = 0 \tag{67}$$

To meet the requirement of control theory, that usually exploits non-redundant ODEs (ordinary differential equations), three independent ODEs can be obtained through the projection matrix $\mathbf{R} \in \mathbb{R}^{6\times3}$, defined as:

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{p}} \tag{68}$$

that leads to the following minimal set of nonlinear ODEs:

$$\left(\mathbf{R}^T\mathbf{M}\mathbf{R}\right)\ddot{\mathbf{p}} + \left(\mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\right)\dot{\mathbf{p}} = \mathbf{R}^T\mathbf{f} \tag{69}$$

4.2.3. Dynamic model of the CSPR with axially-flexible cables

To account for the elastic behaviour of the cables, together with the related damping and inertial effects (although the latter could be neglected in the case under investigation), a formulation inspired by the penalty method used in converting DAEs to ODEs [100] is proposed in this Chapter. Sagging and transversal vibrations are instead neglected, as most of the papers in the literature do, by considering that the payload is much heavier compared to the cables, and that proper tensioning is ensured by setting tensions lower bounds. For these reasons, several CDPRs designed under such assumptions show good performances in practice [103].

In this work, the model of a CSPR with axially-flexible cable is developed with reference to the system under investigation, with a lumped end-effector with 3 DOFs. However, the proposed approach is general and can be applied to CDPRs with different architectures and with end-effector with 6 DOFs as well, by properly formulating the governing DAEs (see e.g. [104]). The development

of this modeling approach is another novel contribution of this Chapter within the literature of CDPRs, as it will be explained in the following of this Section.
Penalty-based relaxation of the constraints is usually adopted to transform a system of DAEs into a set of redundant ODEs. With reference to a generic multibody system, the Lagrange multiplier associated with the i^{th} kinematic constraint ϕ_i assumes the following representation:

$$\lambda_i = \alpha_i \left(\ddot{\phi}_i + 2\omega_i \xi_i \dot{\phi}_i + \omega_i^2 \phi_i \right) \tag{70}$$

By means of Eq. (70), the magnitude of the generalized reaction force related to the i^{th} kinematic constraint depends on the violation of the constraint ϕ_i, and its first and second time-derivatives. The scalar parameters α_i, ξ_i, ω_i represent the parameters of a spring-mass-damper system that is the "penalized constraint". In this way, the "perfectly stiff" constraint (i.e., $\phi_i(t) = 0 \ \forall t$) is replaced by a flexible one, whose flexibility is assumed along the direction of the constrained motion. α_i, ξ_i, ω_i are usually treated as constant tuning parameters to be wisely chosen for trading-off between constraint violation, stability, and accuracy of the resulting ODEs. However, a proper choice of all the parameters in Eq. (70) can be exploited to provide an effective representation of the elastic properties of the system, and hence to obtain accurate estimates of the constraint reactions [105].
All these features make the penalty formulation suitable to model, in an efficient and simple way, the axial flexibility of the cables. Compared with the usual theory of the penalty formulation, that assumes constant values of α_i, ξ_i, ω_i, position-dependent parameters should be adopted since the axial stiffness of each cable k_i varies with the cable length l_i, as discussed by several papers in the literature of CDPRs (see e.g. [102]). However, compared with the models usually adopted in the field of CDPRs, this formulation allows for the straightforward model formulation in term of all the independent coordinates, in lieu of neglecting the motor coordinates as several papers do in the field of cable robotics, also including the inertial and speed-dependent effects of cables.
By adopting a spring-mass-damper formulation for each cable, the Lagrange multipliers in Eq. (63) are re-formulated in accordance with Eq. (70), leading to the following set of nonlinear ODEs:

$$\left(\mathbf{M} + \mathbf{J}^T \mathbf{M}_c \mathbf{J} \right) \ddot{\mathbf{q}} + \left(\mathbf{J}^T \mathbf{M}_c \dot{\mathbf{J}} + \mathbf{J}^T \mathbf{D}_c \mathbf{J} \right) \dot{\mathbf{q}} + \mathbf{J}^T \mathbf{K}_c \mathbf{\Phi} = \mathbf{f} \tag{71}$$

where $\mathbf{M}_c(\mathbf{q}) \in \mathbb{R}^{3\times3}$, $\mathbf{D}_c(\mathbf{q}) \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_c(\mathbf{q}) \in \mathbb{R}^{3\times3}$ are the (non-constant) mass, viscous friction and stiffness matrices. The following formulation is provided for the mass and stiffness matrices, by exploiting the physical properties of the cables, i.e., the Young's modulus E, the cross sectional area S, the linear mass density ρ:

$$\mathbf{M}_c(\mathbf{q}) = \begin{bmatrix} \frac{1}{3}\rho l_1 & 0 & 0 \\ 0 & \frac{1}{3}\rho l_2 & 0 \\ 0 & 0 & \frac{1}{3}\rho l_3 \end{bmatrix} \qquad \mathbf{K}_c(\mathbf{q}) = \begin{bmatrix} \frac{ES}{l_1} & 0 & 0 \\ 0 & \frac{ES}{l_2} & 0 \\ 0 & 0 & \frac{ES}{l_3} \end{bmatrix} \tag{72}$$

Such a formulation of $\mathbf{M}_c(\mathbf{q})$ has been found by imposing the equivalence of the kinetic energy of cables, that are springs with distributed mass. $\mathbf{\Phi} \in \mathbb{R}^3$ is the set of the three kinematic constraint equations ϕ_i $(i = 1,2,3)$:

$$\phi_i = \sqrt{\|\mathbf{p} - \mathbf{a}_i\|^2 - (l_{i0} - r_i \theta_i)^2} \tag{73}$$

As for $\mathbf{M}$, $\mathbf{J}$, $\mathbf{f}$, $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, they have the same meaning as the ones reported in Section 4.2.1. Since the length of the cables are pose-dependent, then $\mathbf{M}_c$ and $\mathbf{K}_c$ are position-dependent as well and in particular, by looking at Eq. (71), it can be noticed that the contribution of these terms is projected on the coordinates of the system through the Jacobian matrix $\mathbf{J}$. The viscous friction matrix $\mathbf{D}_c$ has been modeled in this work through the Rayleigh proportional damping model, as widely adopted in multibody systems (see e.g. [106]).
The resulting set of ODEs should be interpreted as a set of non-redundant ODEs, since the cables are actually flexible along the axial directions. Consequently, the CSPR modeled in Eq. (71) is an underactuated multibody system, since just three independent control forces (i.e., the three motor torques) are adopted to control the 6 DOFs.

4.3. Design of the control scheme

4.3.1. Underlying idea of the control scheme

In this Chapter, the control design is performed through the 3-DOF model by assuming rigid cables, as widely considered in the literature on motion planning and control of CSPR. Actually, cables sometimes exhibit axial flexibility that introduces underdamped vibrational dynamics that downgrade tracking performances if not properly accounted for in the controller design or in motion planning. In this work, rather than exploiting these dynamics in the controller design by adopting a 6-DOF model accurately representing cable flexibility, a simplified approach is proposed to handle flexibility. The control tensions are computed through a model that neglects the cable flexibility. Smoothness of such tensions, and then on the commanded motor torques (that are the actual control forces), is obtained by means of the wise definition of the cost function to be minimized, which penalizes large tension variations, of the presence of constraints on the maximum tension variations, and finally of the formulation with embedded integrator that allows precise path tracking without increasing excessively the gains.

4.3.2. Control architecture

The multibody model in Eq. (69) is nonlinear, since $\mathbf{R}$ depends on positions. To allow for a simpler control design and therefore a less cumbersome computational effort of the controller equations, that boosts the possibility to implement the controller in hard real-time environments, the control scheme proposed in this work does not directly exploits the nonlinear model of the whole system, that would lead to a nonlinear MPC. In contrast, a controller made by two sequential control action is proposed, by splitting the model into two subsystems.
Firstly, the model from the cable tensions to the motion of the suspended end-effector is considered, and feedback pose-dependent (and hence nonlinear) MPC with embedded integrator is formulated to compute the optimal positive tensions that ensure precise tracking of the reference spatial path. In this way, the dynamic matrix of the state-space representation of the dynamic model is constant, while just the input matrix depends on the system pose. Nonlinearities are handled by updating the model at each time step of the control loop, and then assuming the input matrix as constant along the prediction interval used for the MPC design, thus reducing the computational burden, and allowing for real-time calculation.
Secondly, the model from the motor torques to the cable tensions is exploited to obtain the reference optimal currents, that are proportional to the commanded torques, for the three motors.
The control assumes feedback of position and speed of the motors used for winding and unwinding the cables. Rigid cables are also assumed in the control design, thus adopting a simplified model of

the actual dynamics of the system. No tension feedback is instead adopted. To ensure effective tracking, an embedded integrator is here adopted in the MPC formulation (MPC-EI)
A representation of the control architecture is provided in Figure 25, and it will be explained with more details in Section 4.4.1.

4.3.3. Design of the MPC

4.3.3.1. State-Space model of the unconstrained end-effector

The design of the MPC-EI, in the first control action, relies on the following set of three ODEs, that represents the dynamic model of the unconstrained end-effector driven by the tensions of the three cables ($T_i \in \mathbb{R}^+$):

$$m\ddot{\mathbf{p}} = m\mathbf{g} + \sum_{i=1}^{3}\left(-T_i \frac{\mathbf{p}-\mathbf{a}_i}{\|\mathbf{p}-\mathbf{a}_i\|}\right) \tag{74}$$

It should be noted that Eq. (74) holds regardless of the model assumed to represent the cables.
By defining the state vector $\boldsymbol{\chi}_\mathbf{c} = \begin{bmatrix}\dot{\mathbf{p}}^T & \mathbf{p}^T\end{bmatrix}^T \in \mathbb{R}^6$ and the output vector $\mathbf{y}_\mathbf{c} \in \mathbb{R}^3$, a first-order representation of Eq. (74) is inferred, leading to the following nonlinear state-space system in the continuous-time domain:

$$\begin{cases}\dot{\boldsymbol{\chi}}_\mathbf{c}(t) = \mathbf{A}_\mathbf{c}\,\boldsymbol{\chi}_\mathbf{c}(t) + \mathbf{B}_\mathbf{c}(\mathbf{p})\mathbf{u}(t) + \mathbf{B}_\mathbf{gc}\,\mathbf{g} \\ \mathbf{y}_\mathbf{c}(t) = \mathbf{C}_\mathbf{c}\,\boldsymbol{\chi}_\mathbf{c}(t)\end{cases} \tag{75}$$

whose matrices assume the following meaning:

$$\mathbf{A}_\mathbf{c} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & \mathbf{0}_3\end{bmatrix} \quad \mathbf{B}_\mathbf{c}(\mathbf{p}) = \begin{bmatrix}-\frac{1}{m}\mathbf{V}(\mathbf{p}) \\ \mathbf{0}_3\end{bmatrix} \quad \mathbf{B}_\mathbf{gc} = \begin{bmatrix}\mathbf{I}_3 \\ \mathbf{0}_3\end{bmatrix} \quad \mathbf{C}_\mathbf{c} = \begin{bmatrix}\mathbf{0}_3 & \mathbf{I}_3\end{bmatrix} \tag{76}$$

$\mathbf{A}_\mathbf{c} \in \mathbb{R}^{6\times6}$ is the dynamic matrix, that is constant; $\mathbf{B}_\mathbf{gc} \in \mathbb{R}^{6\times3}$ is the constant input matrix of gravity acceleration, that is an external input; $\mathbf{B}_\mathbf{c}(\mathbf{p}) \in \mathbb{R}^{6\times3}$ is the pose-dependent input matrix of the control tensions $\mathbf{u} = \begin{bmatrix}T_1 & T_2 & T_3\end{bmatrix}^T \in \mathbb{R}^3$ (the letter $\mathbf{u}$ is used to recall a common notation used in control theory to denote the control input vector). Finally, $\mathbf{C}_\mathbf{c} \in \mathbb{R}^{3\times6}$ is the output matrix. In Eq. (76) the following matrices have been introduced: $\mathbf{V}(\mathbf{p}) \in \mathbb{R}^{3\times3}$ contains the unitary vectors in the directions of the cables; $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix; $\mathbf{0}_3 \in \mathbb{R}^{3\times3}$ is the null matrix.
By discretizing the nonlinear continuous-time state-space model in Eq. (75), a set of difference equations is formulated, as required by MPC theory:

$$\begin{cases}\boldsymbol{\chi}_\mathbf{d}(k+1) = \mathbf{A}_\mathbf{d}\,\boldsymbol{\chi}_\mathbf{d}(k) + \mathbf{B}_\mathbf{d}(\mathbf{p}(k))\mathbf{u}(k) + \mathbf{B}_\mathbf{gd}\,\mathbf{g} \\ \mathbf{y}_\mathbf{d}(k) = \mathbf{C}_\mathbf{d}\,\boldsymbol{\chi}_\mathbf{d}(k)\end{cases} \tag{77}$$

k denotes the generic time sample; $\boldsymbol{\chi}_\mathbf{d}$ and $\mathbf{y}_\mathbf{d}$ are the discrete-time state and output vectors, respectively; $\mathbf{A}_\mathbf{d} \in \mathbb{R}^{6\times6}$, $\mathbf{B}_\mathbf{d}(\mathbf{p}(k)) \in \mathbb{R}^{6\times3}$, $\mathbf{B}_\mathbf{gd} \in \mathbb{R}^{6\times3}$and $\mathbf{C}_\mathbf{d} \in \mathbb{R}^{3\times6}$ are the discrete-time counterparts of $\mathbf{A}_\mathbf{c}$, $\mathbf{B}_\mathbf{c}(\mathbf{p})$, $\mathbf{B}_\mathbf{gc}$ and $\mathbf{C}_\mathbf{c}$.
Any discretization scheme, among those used in multibody system dynamics or in control theory, can be adopted to obtain the model in Eq. (77). For example, a simple yet effective discretization approach

is the use of low-order exponential methods, that are infrequently used in the multibody dynamics literature [107] while are widely adopted in the fields of control theory,.
With the aim of reducing the computational effort required in the model discretization, a reasonable choice is to assume a zero-order hold (ZOH) approximation of the control input over the time step Δt (i.e., the sampling time of the control loop):

$$\mathbf{u}(t) = \mathbf{u}(k\,\Delta t), \quad k\,\Delta t \le t \le (k+1)\Delta t \tag{78}$$

that leads to the following form of Eq. (77):

$$\begin{cases} \boldsymbol{\chi}_{\mathbf{d}}(k+1) = e^{\mathbf{A}_{\mathbf{c}}\Delta t}\,\boldsymbol{\chi}_{\mathbf{d}}(k) + \left(\int_0^{\Delta t} e^{\mathbf{A}_{\mathbf{c}}\tau}\,\mathbf{B}_{\mathbf{c}}\left(\mathbf{p}(k)\right)\partial\tau\right)\mathbf{u}(k) + \left(\int_0^{\Delta t} e^{\mathbf{A}_{\mathbf{c}}\tau}\,\partial\tau\right)\mathbf{B}_{\mathbf{gc}}\mathbf{g} \\ \mathbf{y}_{\mathbf{d}}(k) = \mathbf{C}_{\mathbf{d}}\,\boldsymbol{\chi}_{\mathbf{d}}(k) \end{cases} \tag{79}$$

Comparing Eq. (79) and Eq. (77) clearly reveals the expressions of the matrices of the discrete-time state-space:

$$\mathbf{A}_{\mathbf{d}} = e^{\mathbf{A}_{\mathbf{c}}\Delta t} \quad \mathbf{B}_{\mathbf{d}}\left(\mathbf{p}(k)\right) = \int_0^{\Delta t} e^{\mathbf{A}_{\mathbf{c}}\tau}\mathbf{B}_{\mathbf{c}}\left(\mathbf{p}(k)\right)\partial\tau \quad \mathbf{B}_{\mathbf{gd}} = \left(\int_0^{\Delta t} e^{\mathbf{A}_{\mathbf{c}}\tau}\,\partial\tau\right)\mathbf{B}_{\mathbf{gc}} \quad \mathbf{C}_{\mathbf{d}} = \mathbf{C}_{\mathbf{c}} \tag{80}$$

By assuming $\mathbf{B}_{\mathbf{c}}\left(\mathbf{p}(k)\right)$ as constant over Δt, that is consistent with using a ZOH, it holds that:

$$\mathbf{B}_{\mathbf{d}}\left(\mathbf{p}(k)\right) = \mathbf{A}_{\mathbf{c}}^{-1}\left(e^{\mathbf{A}_{\mathbf{c}}\Delta t} - \mathbf{I}\right)\mathbf{B}_{\mathbf{c}}\left(\mathbf{p}(k)\right) \tag{81}$$

Since the exact evaluation of matrix exponentials can be very demanding from the computational point of view, $\mathbf{A}_{\mathbf{d}}$ and $\mathbf{B}_{\mathbf{d}}$ can be approximated through low-order Padè approximants or Taylor's series expansions [108]. For example, in this work a first-order approximation is adopted, that can be seen as both the first-order Taylor's series expansions and the (1,0) Padé approximants:

$$e^{\mathbf{A}_{\mathbf{c}}\Delta t} \approx \mathbf{I} + \mathbf{A}_{\mathbf{c}}\Delta t \tag{82}$$

that, finally, lead to the following matrices of the discrete-time state-space in Eq. (77):

$$\mathbf{A}_{\mathbf{d}} = \mathbf{I} + \Delta t\,\mathbf{A}_{\mathbf{c}} \quad \mathbf{B}_{\mathbf{d}}\left(\mathbf{p}(k)\right) = \Delta t\,\mathbf{B}_{\mathbf{c}}\left(\mathbf{p}(k)\right) \quad \mathbf{B}_{\mathbf{gd}} = \Delta t\,\mathbf{B}_{\mathbf{gc}} \quad \mathbf{C}_{\mathbf{d}} = \mathbf{C}_{\mathbf{c}} \tag{83}$$

This choice remarkably reduces the computational effort without introducing noticeable discretization errors in the controller design. Higher-order discretization methods, as well as higher-order approximations of the exponential, could be used at the cost of an increase of the computational effort.

4.3.3.2. Embedding of the integrator

To embed the integrator within the MPC design, with the goal of ensuring precise path tracking, the so-called "difference variables" are introduced:

$$\Delta\boldsymbol{\chi}_{\mathbf{d}}(k) = \boldsymbol{\chi}_{\mathbf{d}}(k) - \boldsymbol{\chi}_{\mathbf{d}}(k-1) \tag{84}$$

$$\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1) \tag{85}$$

and, therefore, an augmented state vector $\boldsymbol{\chi} \in \mathbb{R}^9$ is adopted:

$$\boldsymbol{\chi}(k) = \begin{bmatrix} \Delta\boldsymbol{\chi}_{\mathbf{d}}(k) \\ \mathbf{y}_{\mathbf{d}}(k) \end{bmatrix} \tag{86}$$

Finally, the following augmented first-order model is formulated:

$$\begin{cases} \boldsymbol{\chi}(k+1) = \mathbf{A}\,\boldsymbol{\chi}(k) + \mathbf{B}\left(\mathbf{p}\left(k\right)\right)\boldsymbol{\Delta}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\,\boldsymbol{\chi}(k) \end{cases} \tag{87}$$

where $\mathbf{A} \in \mathbb{R}^{9\times9}$, $\mathbf{B}\left(\mathbf{p}\left(k\right)\right) \in \mathbb{R}^{9\times3}$, $\mathbf{C} \in \mathbb{R}^{3\times9}$ are defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_\mathbf{d} & \mathbf{0}_{6\times3} \\ \mathbf{C}_\mathbf{d}\mathbf{A}_\mathbf{d} & \mathbf{I}_3 \end{bmatrix} \quad \mathbf{B}\left(\mathbf{p}\left(k\right)\right) = \begin{bmatrix} \mathbf{B}_\mathbf{d}\left(\mathbf{p}\left(k\right)\right) \\ \mathbf{C}_\mathbf{d}\mathbf{B}_\mathbf{d}\left(\mathbf{p}\left(k\right)\right) \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \mathbf{0}_{3\times6} & \mathbf{I}_3 \end{bmatrix} \tag{88}$$

Again, as in the continuous time model, $\mathbf{A}$ and $\mathbf{C}$ are constant matrices, while $\mathbf{B}\left(\mathbf{p}\left(k\right)\right)$ changes with the pose.

4.3.3.3. Evaluation of the prediction matrices for MPC-EI

The computation of the optimal tensions by the MPC-EI is based on a prediction of the state trajectory over the prediction horizon $N_p \in \mathbb{R}$. Due to the presence of a variable input matrix $\mathbf{B}\left(\mathbf{p}\left(k\right)\right)$, the dynamic model adopted for prediction is updated at each time step k based on the pose. In contrast, such a model is kept constant over the whole prediction horizon to predict the state (and the output) with a smaller computational effort. Hence, the output vector predicted at each time step, denoted as $\mathbf{Y} \in \mathbb{R}^{3N_p}$ (by assuming a vectorial representation of the prediction over the whole prediction horizon) is described through the following linear model:

$$\mathbf{Y} = \mathbf{F}\mathbf{x}(k_i) + \boldsymbol{\Theta}\left(\mathbf{p}\left(k_i\right)\right)\boldsymbol{\Delta}\mathbf{u}_{\mathbf{N_c}} \tag{89}$$

where $\mathbf{F} \in \mathbb{R}^{3N_p\times9}$ and $\boldsymbol{\Theta}\left(\mathbf{p}\left(k_i\right)\right) \in \mathbb{R}^{3N_p\times3N_c}$ are defined as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N_p} \end{bmatrix} \quad \boldsymbol{\Theta}\left(\mathbf{p}\left(k_i\right)\right) = \begin{bmatrix} \mathbf{CB}\left(\mathbf{p}\left(k_i\right)\right) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CAB}\left(\mathbf{p}\left(k_i\right)\right) & \mathbf{CB}\left(\mathbf{p}\left(k_i\right)\right) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{N_p-1}\mathbf{B}\left(\mathbf{p}\left(k_i\right)\right) & \mathbf{CA}^{N_p-2}\mathbf{B}\left(\mathbf{p}\left(k_i\right)\right) & \cdots & \mathbf{CA}^{N_p-N_c}\mathbf{B}\left(\mathbf{p}\left(k_i\right)\right) \end{bmatrix} \tag{90}$$

4.3.3.4. Constrained optimization problem: formulation and solution

The path tracking problem is translated into the MPC design by defining a suitable objective function to minimize, $\Psi \in \mathbb{R}$, trading between the requirements of reducing the error and the control effort. In this work, to ensure adequate smoothness that allows handling the presence of cable flexibility, the control effort is included in Ψ through the tension time-derivatives, that is in turn represented through the tension variation $\boldsymbol{\Delta}\mathbf{u}$ defined in Eq. (85). The following cost function is therefore adopted:

$$\Psi = (\mathbf{Y}^{\mathbf{des}} - \mathbf{Y})^T \mathbf{R}_\mathbf{Y} (\mathbf{Y}^{\mathbf{des}} - \mathbf{Y}) + \boldsymbol{\Delta}\mathbf{u}_{\mathbf{N_c}}{}^T \mathbf{R}_{\boldsymbol{\Delta}\mathbf{u}} \boldsymbol{\Delta}\mathbf{u}_{\mathbf{N_c}} \tag{91}$$

$\mathbf{R}_\mathbf{Y} \in \mathbb{R}^{3N_p\times3N_p}$ and $\mathbf{R}_{\boldsymbol{\Delta}\mathbf{u}} \in \mathbb{R}^{3N_c\times3N_c}$ are weighing matrices adopted to tune the controller; $\mathbf{Y}^{\mathbf{des}} \in \mathbb{R}^{3N_p}$ is the vector of the reference trajectories over the prediction horizon, represented through the vector of reference trajectories (defined at the time step k_i):

$$\mathbf{Y}^{\mathbf{des}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 & \cdots & \mathbf{I}_3 \end{bmatrix}^T \mathbf{r}(k_i) = \mathbf{F}^{\mathbf{des}}\mathbf{r}(k_i) \tag{92}$$

($\mathbf{F}^{\mathbf{des}} \in \mathbb{R}^{3N_p\times3}$ is introduced for brevity of representation). $\boldsymbol{\Delta}\mathbf{u}_{\mathbf{N_c}} \in \mathbb{R}^{3N_c}$ collects the optimal values of tension variations for the three cables over the control horizon lasting $N_c \in \mathbb{R}$ samples.

Bounds on the feasible tensions of the three cables $\mathbf{u}(k)$ are accounted for in the control design by means of upper ($\mathbf{u_{max}} = T_{\max}\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \in \mathbb{R}^3$) and lower ($\mathbf{u_{min}} = T_{\min}\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \in \mathbb{R}^3$) bounds. $T_{\min}$ sets the positiveness tension requirements, together with a certain safety margin; $T_{\max}$ embeds the constraint on the maximum admissible strain, to avoid cable failure. Since MPC-EI formulates the cost function in terms of $\mathbf{\Delta u}$, constraints on the tension variations can be easily included as well, to ensure bounded derivatives of tension and hence improve smoothness of the control action.
The solution of the control design problem relies on quadratic programming algorithms in the presence of linear inequalities. For example, in this work it is adopted the Hildreth's method ([40]) that ensures fast computation and good numerical conditioning, and hence is suitable for real-time computation.
The solution of the optimization problem defined through the cost function in Eq. (91) leads to the vector of the optimal predicted control inputs over the control horizon $N_c \in \mathbb{R}$, that is expressed in the following vectorial form:

$$\mathbf{\Delta u_{N_c}} = \begin{bmatrix}\mathbf{\Delta u}^T(k_i) & \mathbf{\Delta u}^T(k_i+1) & \cdots & \mathbf{\Delta u}^T(k_i+N_c-1)\end{bmatrix}^T \tag{93}$$

The control horizon is the number of samples along which the optimal control action will be spread. However, this latter is not going to be applied directly one by one, in a consecutive manner, otherwise an open-loop controller would be achieved. Indeed, to avoid this aspect, in accordance with the Receding Horizon Principle, only the control action related to the first sample, in this particular case the three entries of $\mathbf{\Delta u}(k_i)$, is considered and then transformed into the commanded tensions to be exerted, while the other ones are discarded. This process is repeated at each time step.

4.3.4. Computation of the motor torques

The optimal values of the cable tensions computed by the MPC-EI, here denoted T_i^{MPC} (that are collected into vector $\mathbf{u}^{MPC}$), are fed into the second term of the controller, to transform them into the commanded torques (and hence currents) for the three motors. This transformation relies on the dynamic model of each motor, that is represented through the following ODE:

$$J_i\ddot{\theta}_i(t) + f_{v,i}\dot{\theta}_i(t) = \tau_{m,i}(t) - T_i(t)\,r_i \tag{94}$$

$\theta_i \in \mathbb{R}$ is the absolute rotation of the shaft of the i^{th} motor; J_i is the moment of inertia of the rotor, plus the drum and idle pulleys; $f_{v,i} \in \mathbb{R}$ is the viscous friction coefficient; $\tau_{m,i} \in \mathbb{R}$ is the motor torque; $r_i \in \mathbb{R}$ is the radius drum.
Since no feedback of the actual tensions is supposed to be available, a model-based approach is exploited to compute the motor torques $\tau_{m,i}$ that lead to the required cable tensions. Let us compute the reference acceleration $\ddot{\theta}_i^{ref}(t)$ and speed $\dot{\theta}_i^{ref}(t)$ by means of the inverse kinematics. Hence, the algebraic inverse dynamic model is exploited as follows for each motor:

$$\tau_{m,i} = J_i\ddot{\theta}_i^{ref}(t) + f_{v,i}\dot{\theta}_i^{ref}(t) + r_i T_i^{MPC}(t) \tag{95}$$

The required motor torques are, in turn, transformed into the commanded currents for the three independent motor current controllers, i_i^{ref}, by taking advantage of the motor torque constant $k_{t,i}$:

$$i_i^{ref} = \frac{\tau_{m,i}}{k_{t,i}} \tag{96}$$

It should be noted that no model of the cable flexibility is assumed in Eq. (95).

4.4. Numerical results

4.4.1. System description

To propose a severe test for the proposed controller, very flexible cables are assumed with $ES = 24.9\ kN$. This value has been taken from [90] as a meaningful example of a flexible CSPR, since it uses values of ES that are smaller of several order of magnitudes, if compared to other papers (see e.g. [102]).
The simulator that emulates the multibody system to be controlled includes a simplified dynamic model of the servo-controlled electric motors, whose frequency response is modeled as a first order linear system whose bandwidth is $2\ kHz$, as typically occurs in off-the-shelf brushless motors (see e.g. [109]). The three motors are supposed to be equipped by encoders with 14400 pulses per revolution. Speed estimation is made by means of numerical derivatives plus low pass filtering. The presence of quantization noise and phase lag, due to low pass filtering, create a realistic test case that prevent excessive increases of the controller gains. No other sensors are available; the controller therefore relies on just the motor encoder measurements. Since the proposed MPC-EI exploits the load positions and speeds, that are not measured, estimation is performed through the forward kinematics, i.e., by assuming the rigid-cable model. The position forward kinematics equations for the system sketched in Figure 24 (with the following cartesian coordinates of the cable exit points: $\mathbf{a}_1 = \begin{bmatrix} -\frac{l}{2} & \frac{w}{2} & 0 \end{bmatrix}^T$, $\mathbf{a}_2 = \begin{bmatrix} \frac{l}{2} & \frac{w}{2} & 0 \end{bmatrix}^T$, $\mathbf{a}_3 = \begin{bmatrix} 0 & -\frac{w}{2} & 0 \end{bmatrix}^T$) are:

$$\begin{aligned} x &= \frac{\rho_1^{\,2} - \rho_2^{\,2}}{2l} \\ y &= \frac{-2\rho_1^{\,2} - 2\rho_2^{\,2} + 4\rho_3^{\,2} + l^2}{8w} \\ z &= -\sqrt{\rho_3^{\,2} - x^2 - \left(y + \frac{w}{2}\right)^2} \end{aligned} \tag{97}$$

As for the speed forward kinematics problem, the relationship between $\dot{\mathbf{p}}$ and $\dot{\boldsymbol{\theta}}$ can be easily inferred from proper partitioning of matrix $\mathbf{R}$ in Eq. (68).
This choice of sensed variables, that exploits the position sensors already available in electric motors, leads to a collocated control scheme. A paramount advantage of collocated control schemes is that they have higher stability margins than non-collocated architectures where direct measurement of the load speed and position is exploited, as widely recognized in the literature of underactuated flexible multibody systems (see e.g. [110]). In contrast, a purely collocated control is not able to sense, and hence to control, static deflection due to external forces applied to the load.
The resulting control architecture is shown in Figure 25, where the motor torques and the reference currents are collected in vectors $\boldsymbol{\tau}_m \in \mathbb{R}^3$ and $\boldsymbol{i}^{ref} \in \mathbb{R}^3$, respectively. No feedforward is exploited, to provide a more severe assessment of the effectiveness of the proposed control approach.
The sample time Δt of the discrete-time controller is chosen to be 2×10^{-3} s, as a compromise between the need of granting adequate controller bandwidth and reducing the number of calculations required to solve the optimal control problem. The tuning parameters of the controller and the physical parameters of the simulated CSPR are listed in Table 7.

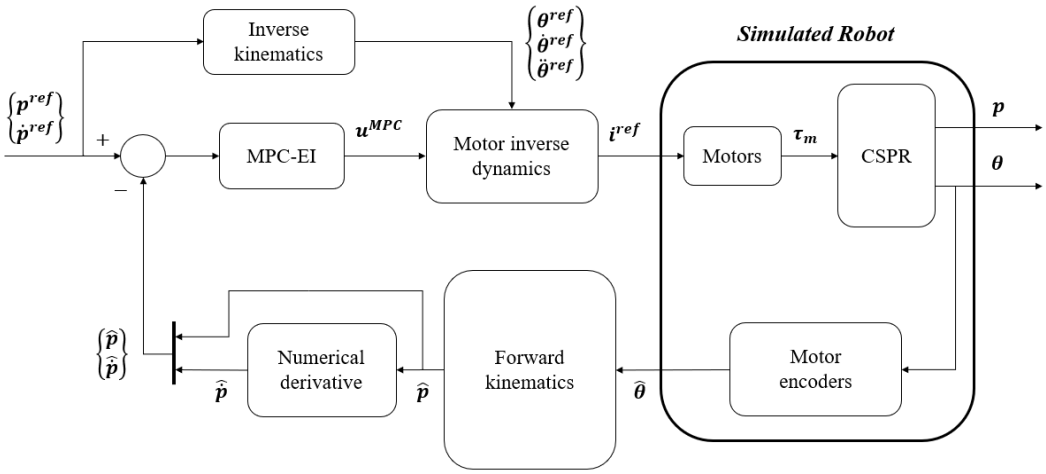


Figure 25. Architecture of the control scheme and block scheme of the simulated system.

The goal of the control is to make the end-effector tracking the spatial path. Three different motion references are therefore tested and proposed in the following Sections to highlight different features of the proposed control schemes and its effectiveness in tracking the spatial path with reduced contour error and negligible excitations of the cable flexibility. The same tuning of the controller parameters is adopted for all the test case, both in the presence of rigid and elastic cables, to provide a fair validation. Comparison with a standard MPC is also provided in one of the tests, to show the superiority of the proposed scheme to ensure precise path tracking with negligible excitation of the (unmodeled) flexible dynamics.

Table 7. System parameters

Parameter	Value
w	1.69 m
l	1.78 m
h	1.95 m
$J_{m,1}, J_{m,2}, J_{m,3}$	2.6×10^{-5} kgm^2
$f_{v,1}, f_{v,2}, f_{v,3}$	5×10^{-3} Nms/rad
r_1, r_2, r_3	0.036 m
m	10 kg
ES	24,9 kN
$T_{\min}$; $T_{\max}$	10 ; 200 N
Δt	2×10^{-3} s
N_c ; N_p	3 ; 120
$\mathbf{R}_{\mathbf{Y}}$; $\mathbf{R}_{\Delta\mathbf{u}}$	$\mathbf{I}_{240}$; $1\times10^{-3}\,\mathbf{I}_3$

4.4.2. Test cases

4.4.2.1. Test 1: point-to-point motion with an unfeasible trajectory

The first trajectory consists of a descending step reference along the z-axis, while keeping the references on x-axis and y-axis unchanged. This trajectory is, clearly, unfeasible since tracking the discontinuous commanded position imposes negative infinite speeds and accelerations along the z-

axis, thus causing negative tensions, and hence leading to a possible slack condition for the cables. For this reason, it is never used for motion planning in CSPRs (and, in general, in underactuated multibody systems as well).

The temporal tracking response of the proposed control scheme is shown in Figure 26 and reveals that the proposed control scheme allows for a quick and accurate settling to the final target position, with just a small overshoot even in the presence of flexible cables. Such a smooth response is achieved thanks to the formulation with embedded integrator in the presence of penalization and hard constraints on the tension derivatives.

In the case of rigid cables, the controller ensures no steady state error since the estimation of the position obtained through forward kinematics is exact (except for the error due to the encoder quantization). In contrast, in the case of flexible cables, both the initial and final positions do not match the target one because of the static cable elongation. It should be noted that such an error cannot be compensated by a collocated control scheme that does not sense it. Compensation of such an error could be done if direct measurement of the end-effector position is adopted or state observers based on the dynamic model of the flexible system are implemented, leading to a non-collocated position loop, or to a state-feedback controller. This control approach, however, is out of the scope of this work.

The actual cable tensions are shown in Figure 27: it clearly confirms the capability of the proposed control scheme to ensure bounded and smooth tensions. In the case of flexible cables, the actual tensions go slightly beyond the lower bound of $10\ N$, by reaching the minimum value of $6.5\ N$, which is however not critical if just a small safety margin is chosen in setting the lower bounds.

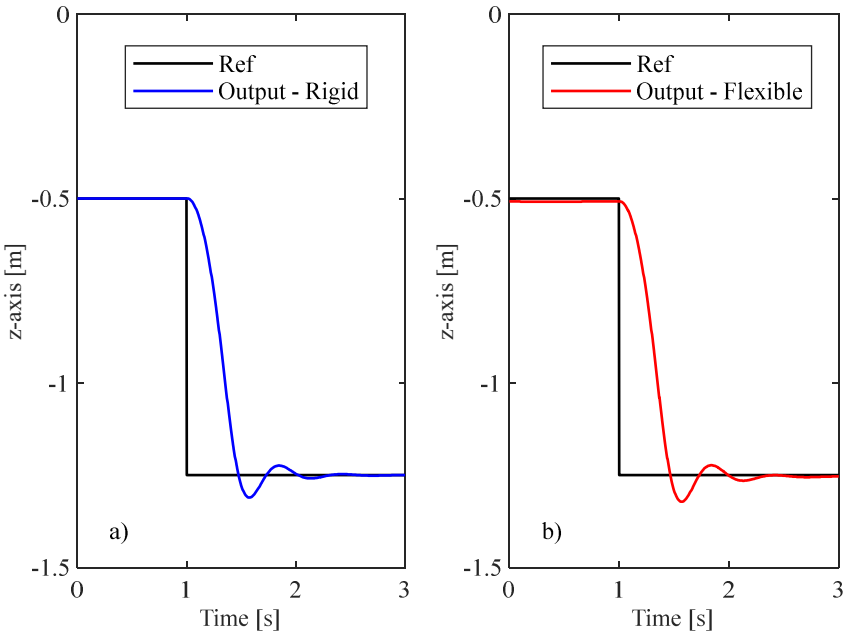


Figure 26. Temporal tracking responses with step reference, in the presence of rigid model (a) and flexible one (b).

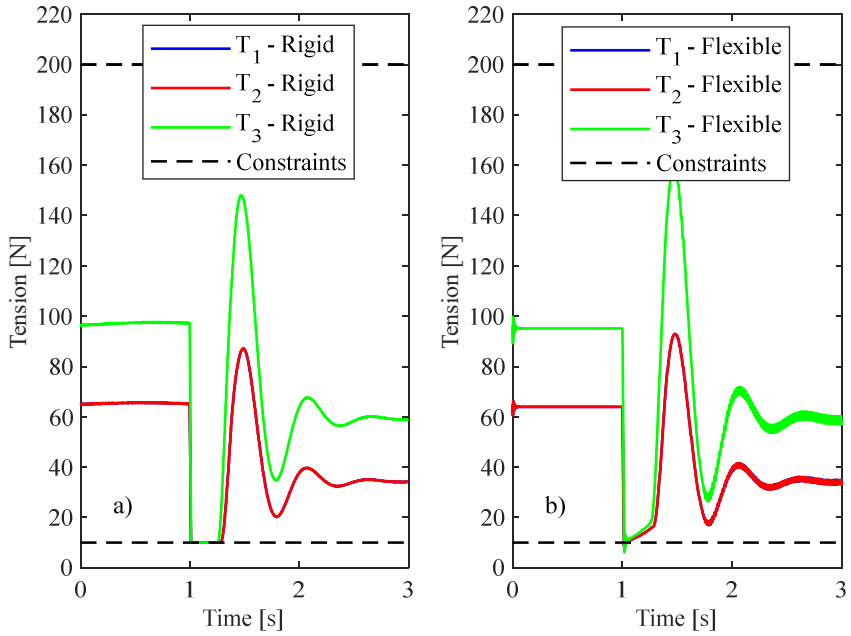


Figure 27. Cable tensions with step reference, in the presence of rigid model (a) and flexible one (b).

4.4.2.2. Discussion on the computational effort

This Section discusses the issue of the computational effort by analysing the required CPU time at each time step, to evaluate whether the proposed controller architecture is suitable to be performed in real-time controllers. The numerical results have been obtained using a laptop PC with a 16.0 *GB* RAM and an octa-core processor (11th Gen Intel(R) Core(TM) i7-11800H) characterized by a clock frequency of 2.30 *GHz* . It should be noted that no parallel computing has been adopted, and therefore just one core performs calculation.

The entire control algorithm can be described as a sequence of the following actions: update of continuous input matrix $\mathbf{B}_c(\mathbf{p})$ through Eq.(76) (including the estimation of the pose through the forward kinematics in Eq.(97)); discretization of the continuous state-space model through Eq.(83); condensing process for the evaluation of the prediction matrices $\mathbf{F}$ and $\boldsymbol{\Theta}(\mathbf{p}(k_i))$ through Eq.(90); constrained minimization of the cost function Ψ expressed in Eq.(91); evaluation of the commanded motor torques $\tau_{m,i}$ for each motor by exploiting Eq.(95) (and including the inverse kinematic as well).

Therefore, to have a clear and detailed image of the computational burden, Figure 28 shows the CPU time required at each time step, with reference to test 1. Such a test has been chosen to be shown in this work since it is the most critical for the minimization of the cost function of the MPC-EI algorithm, due to the achievement of the tension lower bounds in part of the motion. Slightly smaller computational efforts are required in the other tests proposed in this Chapter, and therefore are omitted for brevity.

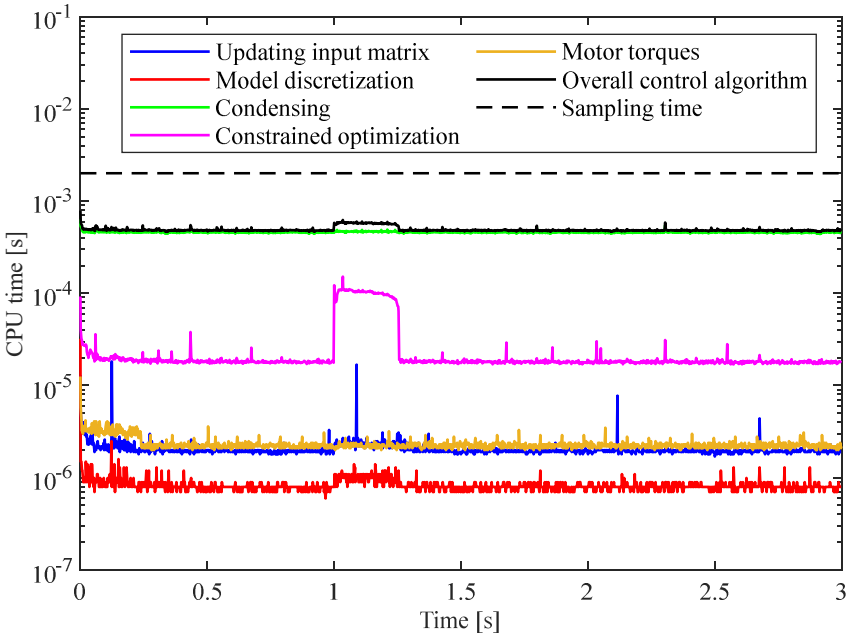


Figure 28. CPU times required for each process that constitutes the proposed control algorithm during step response.

Comparing the overall CPU time at each time step with the control sampling time $\Delta t = 2\times10^{-3}$ s clearly highlights that the proposed controller is not so demanding, requiring a maximum value equal to 6.3×10^{-4} s and an average equal to 4.9×10^{-4} s. It should be noted that the computational effort could be further decreased by optimizing the code implementing the control scheme. This issue will be objective of further investigations.

4.4.2.3. Test 2: triangular path

4.4.2.3.1 The proposed control scheme

The second test case consists of a planar triangular path (in a horizontal plane placed at $z = -0.5$ *m*), where each side is performed in a motion time of 2 *s* through a 5th-degree polynomial, rest-to-rest motion law. A rest interval of 1 *s* is included at each vertex of the planar triangle, leading to an overall execution time of 10 *s*. The path tracking response is shown in Figure 29: the path executed by the CSPR with flexible cables is almost overlapped with the reference path (denoted in the figures as "Ref") both in the presence of rigid and elastic cables. A clearer evaluation of the result can be inferred from Figure 30 which shows the contour error in the plane of the triangle, that is defined as the difference between the reference and the actual path. In the case of rigid cables the maximum value is 3.7 *mm* and the RMS (Root Mean Square) value is just 1.0 *mm* ; in the presence of flexible cables, its maximum and RMS values are, respectively, 4.1 *mm* and 1.4 *mm*. The static cable deformation introduces an initial and final contour error component that is equal to 0.6 *mm*. As already mentioned, such an error cannot be neither detected by the encoders nor controlled by a collocated control scheme. Apart from such a contribution, the increase of the contour error is almost negligible, since no relevant oscillations appear in the plot, due to the smooth control action computed by the designed MPC-EI.

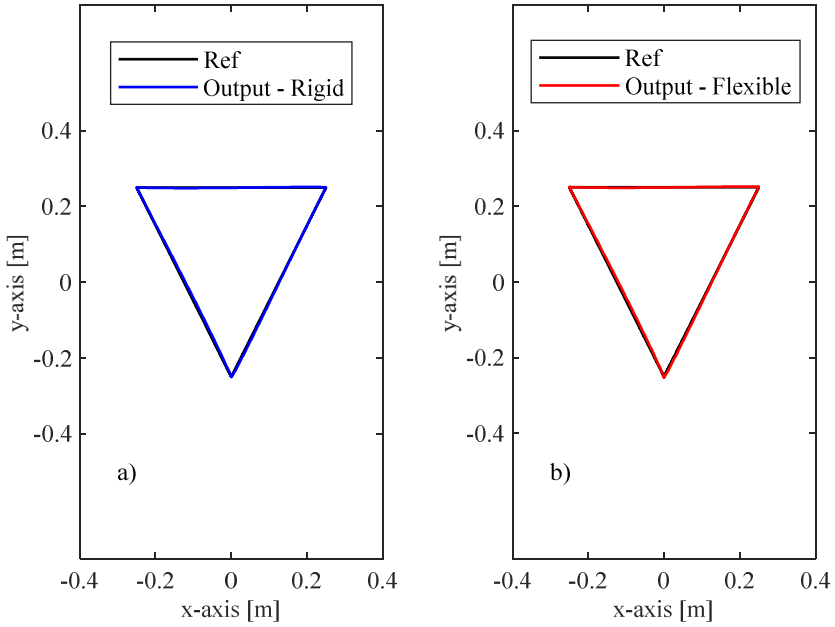


Figure 29. Path tracking responses with triangular reference, in the presence of rigid model (a) and flexible one (b).

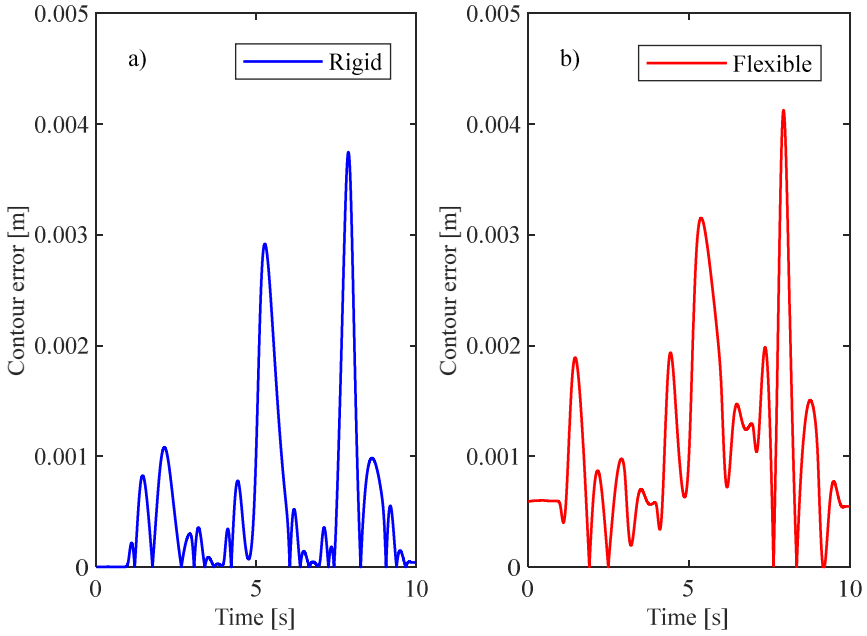


Figure 30. Contour errors with triangular reference, in the presence of rigid model (a) and flexible one (b).

Finally, the cable tensions are shown in Figure 31, that confirms that bounds are satisfied, and smooth tension changes are obtained as required in the controller design: despite the cable flexibility, no meaningful oscillations appear at the frequency components of motion. Just some small amplitude, high frequency oscillations are excited by the quantization errors of the encoders, that lead to noisy sensed position and estimated speed. Control gains could be reduced if the feedback controller is supported by a feedforward compensation of the load gravity and inertia forces, as often done in controlling CSPRs; however, in this work it is not accounted for to provide a more severe test for the proposed feedback controller.

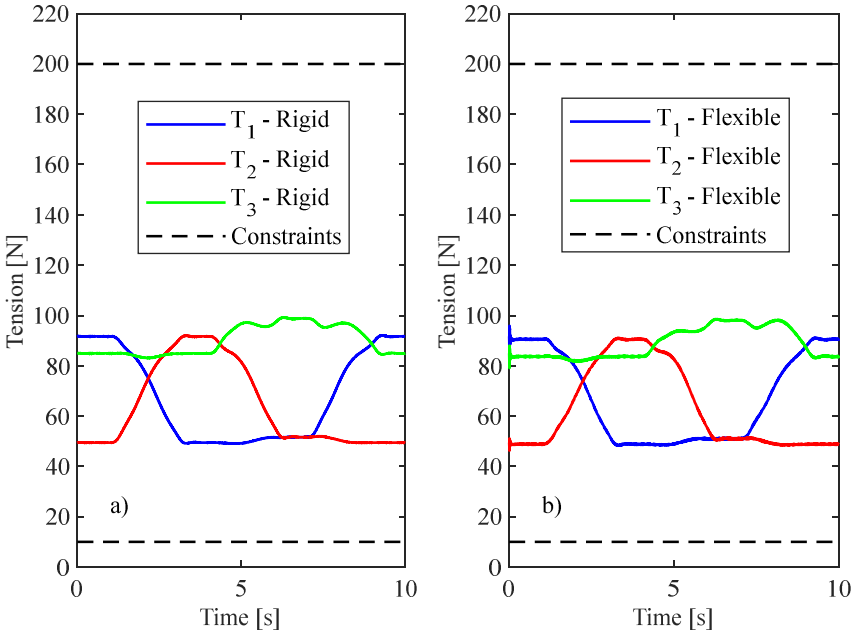


Figure 31. Cable tensions with triangular reference, in the presence of rigid model (a) and flexible one (b).

4.4.2.3.2 Use of a benchmark controller

To further stress the benefits of the proposed control scheme, a "classical" MPC formulation without embedded integrator is also implemented. With the goal to make this comparison as fair as possible, the same prediction and control horizon are assumed. In contrast, the weight $\mathbf{R}_u$ has been tuned, by showing the results of two different choices of such a matrix. A first controller has been designed by setting $\mathbf{R}_u = 5\times10^{-6}\,\mathbf{I}_3$, which will be denoted as the "expensive controller", since it is aimed at obtaining more precise path tracking at the cost of large control effort, i.e. large tensions; a smoother tuning has been obtained by setting $\mathbf{R}_u = 5\times10^{-5}\,\mathbf{I}_3$, which will be indicated as the "cheap controller", to increase the penalty of large control efforts. Figure 32 shows the path tracking results for both the benchmark controllers. It is evident that increasing the gains is necessary to ensure acceptable path tracking; in contrast, the MPC controller with $\mathbf{R}_u = 5\times10^{-5}\,\mathbf{I}_3$ leads to large errors. The evaluation of the contour errors, shown in Figure 33, reveals that large errors are obtained by both controllers, compared with those provided by the MPC-EI. The "expensive controller" leads to a maximum contour error equal to $13.8\ mm$ and to a RMS value of $4.9\ mm$ when the rigid model response is considered; these values remarkably increase if the flexible model is considered, reporting a maximum contour error of $63.4\ mm$ and a RMS value equal to $24.6\ mm$. In contrast, the "cheap controller" has a maximum contour error equal to $18.6\ mm$, with a RMS value of $6.3\ mm$, if the rigid model response is taken into account; these values remarkably degrade if the flexible model response is considered, leading to a maximum contour error of $85.1\ mm$ and a RMS value equal to $32.1\ mm$. In both cases, low frequency oscillations appear due to the controller and the cable flexibility.

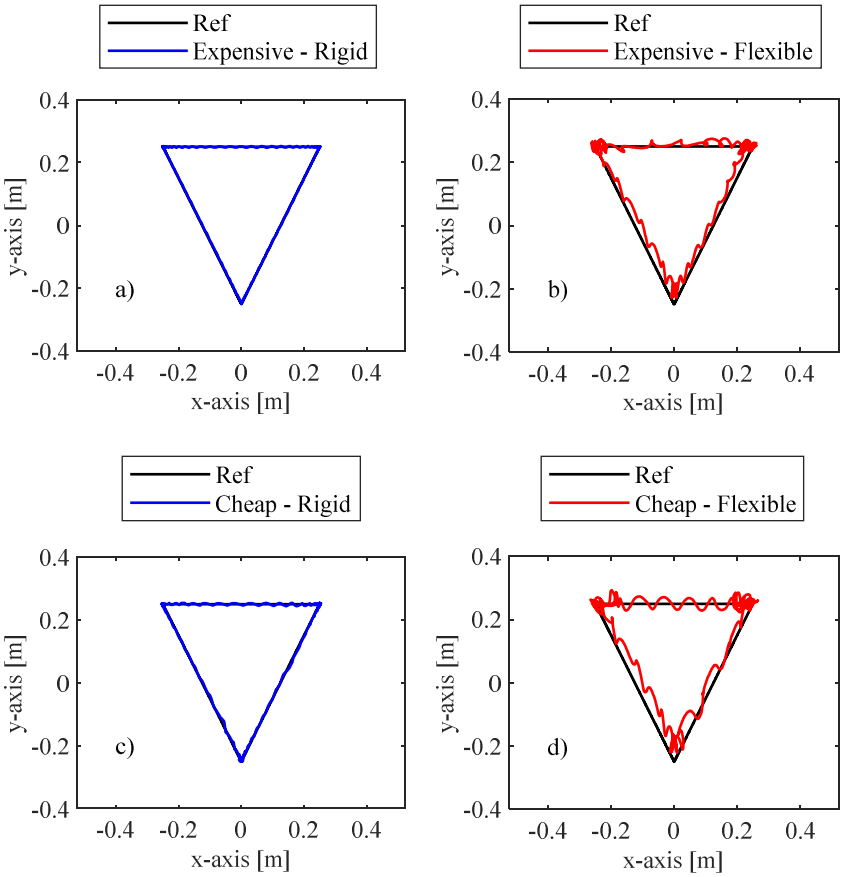


Figure 32. Path tracking responses with triangular reference: benchmark controllers with different tuning parameters in the presence of rigid model (a, c) and flexible one (b, d).

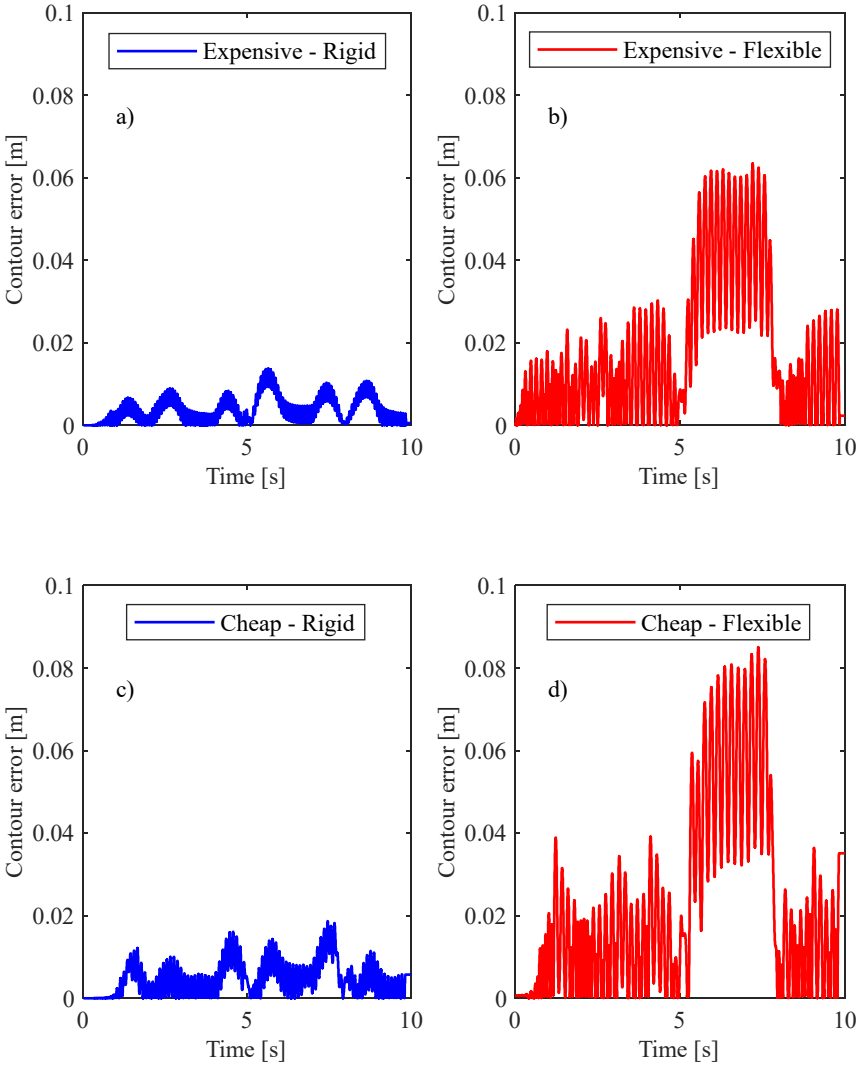


Figure 33. Contour errors with triangular reference: benchmark controllers with different tuning parameters in the presence of rigid model (a, c) and flexible one (b, d).

4.4.2.4. Test 3: spiral path

The last test case considers a more complex scenario: an ascending spatial Archimedean' spiral path. To define such a path, polar coordinates (r , θ) are assumed in the *xy*-plane (see Figure 34):

$$\begin{aligned} x(t) &= r(\theta)\cos(\theta) \\ y(t) &= r(\theta)\sin(\theta) \end{aligned} \tag{98}$$

where:

$$r(\theta) = \psi + \gamma\theta(t) \tag{99}$$

(with ψ and γ related to the values assumed in the starting and final positions). In turn, $\theta(t)$ is defined as a 5th-degree polynomial, rest-to-rest motion law from 1.57 *rad* to 26.70 *rad* (i.e., four revolutions are considered), with a motion time of 12 *s* (the test also includes a rest interval of 1 *s*

both at the beginning and at the ending). The displacement along the z-axis is defined as a 5^{th}-degree polynomial, rest-to-rest motion law from $-1.5\ m$ to $-0.5\ m$.

Such a path goes sometimes outside the static workspace, which is a condition that is rarely considered in the literature. Additionally, it almost reaches the bounds of the dynamic workspace, defined as the set of all the end-effector poses and accelerations for which all cables are in tension [111], and therefore its exact tracking would sometimes require tensions that are smaller than the lower bound on the feasible tensions. These features lead to a very challenging test case for validating the effectiveness of the proposed control scheme, since ensuring bounded tensions, despite the unfeasible reference adopted, is a duty of the proposed control scheme.

Path tracking responses are reported both in Figure 34, by projecting the path in a *xy*-plane , and Figure 35 through a three-dimensional representation. Both figures clearly show that a very precise contouring is ensured by the proposed control architecture, since the actual and the reference paths are almost overlapped.

A more detailed evaluation of the path tracking capability is provided by Figure 36 that shows the three-dimensional contour errors: taking into account the rigid model response, the maximum value is equal to 8.2 *mm* and the RMS value is just 2.8 *mm* while, considering the flexible model outcome, its maximum and RMS values are equal to 12.6 *mm* and 7.2 *mm*, respectively. As in the previous test, the initial and final contour errors are greater than zero because of the uncontrollable static deflection due to the cable flexibility. Apart from such a component, very small error in contouring the path is achieved, despite the severe lower bound on the feasible cable tensions.

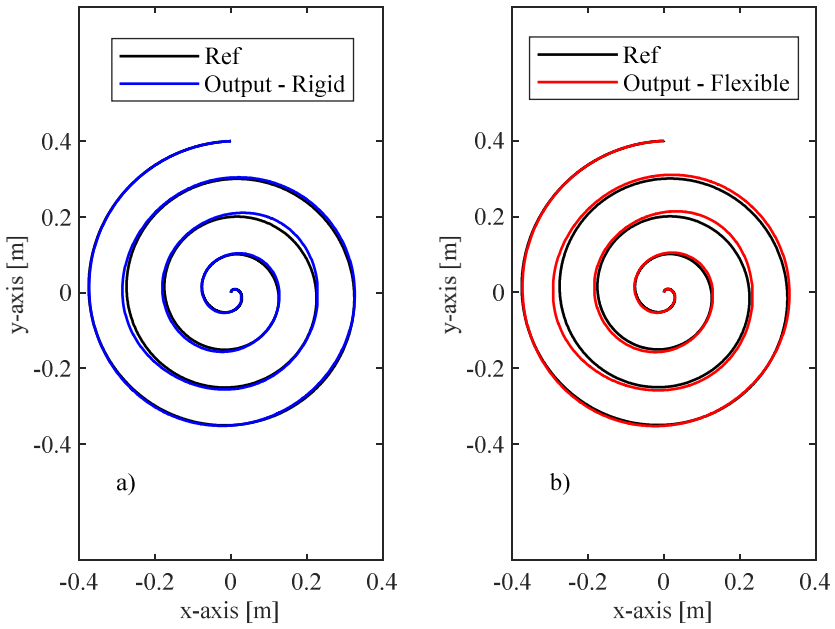


Figure 34. Path tracking responses with spiral reference, in the presence of rigid model (a) and flexible one (b).

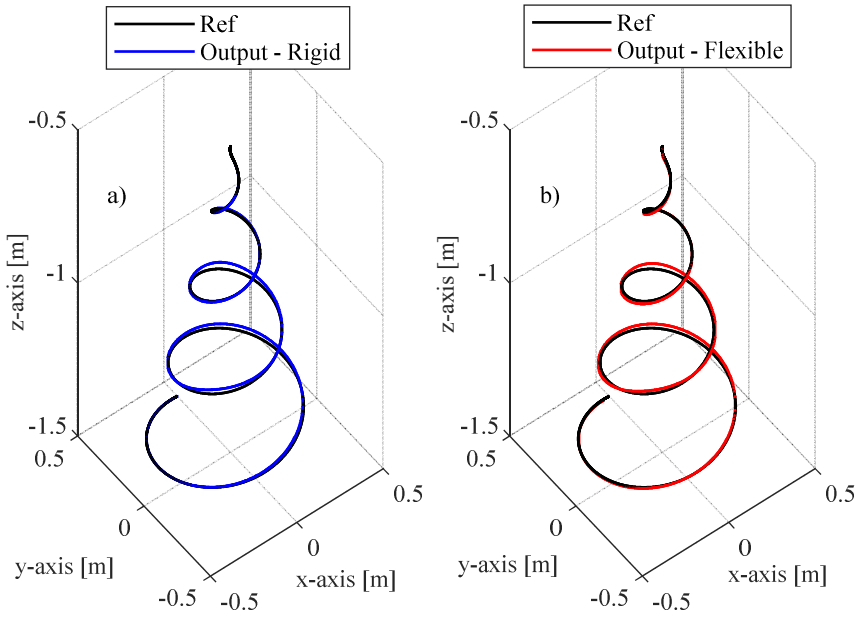


Figure 35. 3D-path tracking responses with spiral reference, in the presence of rigid model (a) and flexible one (b).

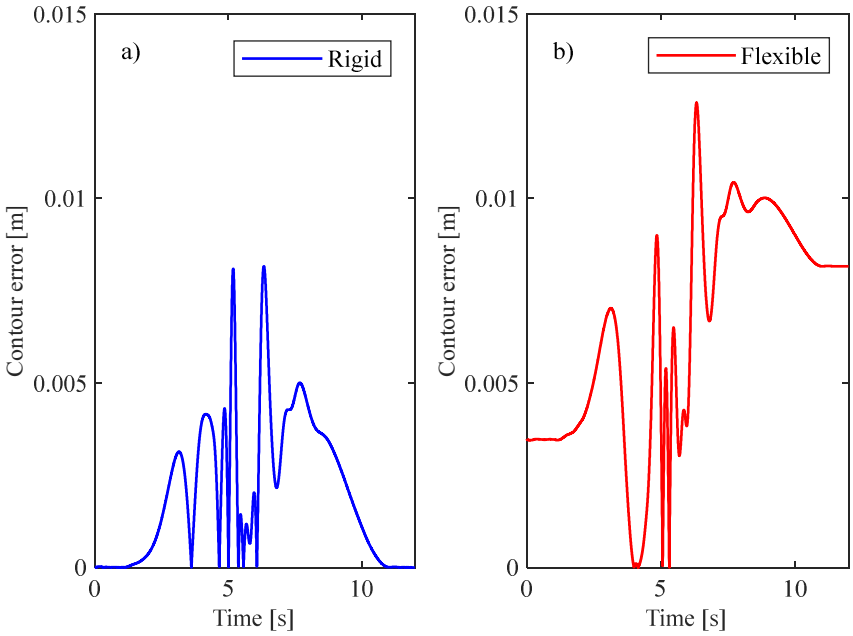


Figure 36. Contour errors with spiral reference, in the presence of rigid model (a) and flexible one (b).

Finally, the commanded cable tensions are displayed in Figure 37. Once again, the bounds are satisfied, and smooth changes tensions are obtained as required in the controller design.

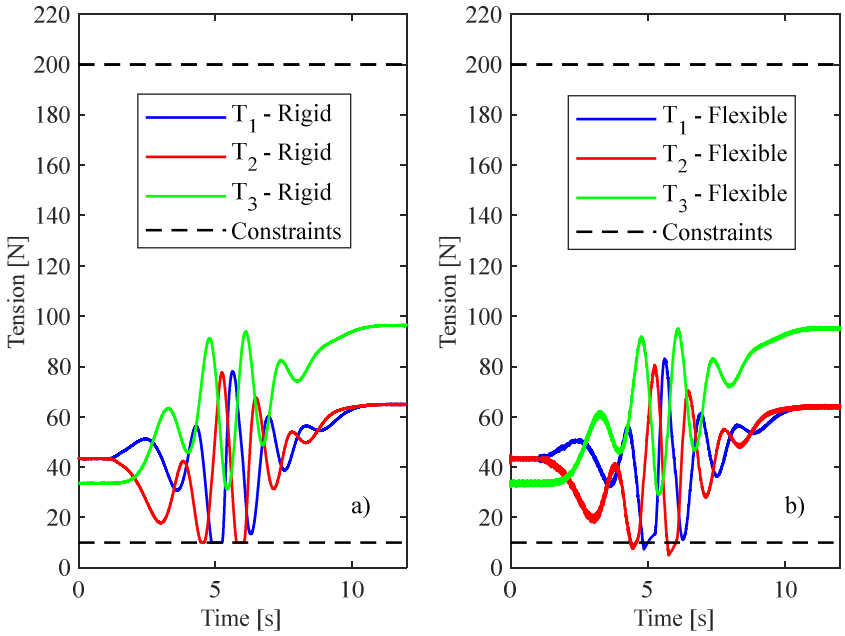


Figure 37. Cable tensions with spiral reference, in the presence of rigid model (a) and flexible one (b).

It should be noted that if this test is performed through the two benchmark controllers discussed in Section 4.4.2.3.2, severe increases of the contour error are achieved, together with an oscillating behaviour of the cable tensions. These results are not shown since they recall those already proposed in Section 4.4.2.3.2.

4.4.2.5. Robustness margin analysis

A final remark is carried on considering the robustness properties of the proposed control algorithm. Since the controller is position-dependent, due to the nonlinear dynamics of the CSPR under investigation, local analysis of the robustness margins is performed by linearizing the model of closed-loop system along the performed trajectory. In particular, the spiral path of Section 4.4.2.3 is considered since it is the one that cover the larger workspace among the three test cases presented. The same evaluation carried on with the other paths, by following the same approach here proposed, leads to similar results, and is therefore here omitted for brevity.

Robustness analysis is essential in any control design because “all models are wrong” [112] and therefore mismatches between the system model and the dynamics of the real setup are inevitable. Classical margins take into account these mismatches by considering gain and phase perturbations in the closed-loop system in an independent manner. However, “real systems differ from their mathematical models in both magnitude and phase” [42] therefore simultaneous perturbations are here considered thanks to the exploitations of the disk margin approach [42]. Since gain and phase perturbations are considered acting simultaneously on the closed-loop system, this approach could lead to smaller stability margins, respect to the classical approach, but in reality more reliable. Indeed, a dynamic model could be characterized by large gain and phase margins considering the classical approach, leading to think that the real system could be highly robust but, actually, a small combination of gain and phase perturbations could cause instability, underlying the limitations of the classical approach.

Additionally, given the MIMO nature of CSPRs, and CDPRs in general, two possible paths can be followed to evaluate the robustness of the closed-loop system when the disk margin method is considered: “loop-at-a-time” margins and “multiloop” margins. The former represents a typical extension of classical margins for MIMO systems and it consists on perturbing each feedback channel

singularly, while the latter corresponds to the disk margin technique applied to MIMO systems, where all feedback channels are perturbed simultaneously. However, the "loop-at-a-time" analysis is not able to capture the effect of simultaneous perturbations occurring in multiple channels and, for this reason, it can provide overly optimistic robustness parameters. To overcome this limitation, the "multiloop" technique is exploited in this work, which leads to smaller stability margins, compared to the previous mentioned analysis, but with a higher level of reliability.
More precisely, Figure 38 shows the relationship between stability margins and frequency, perturbing all system outputs (end-effector cartesian position and speed). It can be noticed that the minimum gain margin is $6.2\ dB$, while the minimum phase margin is $37.9\ °$. Remembering that both gain and phase perturbations are applied simultaneously in a "multiloop" manner, it can be concluded that the closed-loop system presents a high level of robustness with stability margins that are reliable for a real test case. This aspect has been indirectly seen thanks to the results reported in the different test cases, where model mismatches were included through cable flexibility while stability was qualitatively confirmed by the behaviour of the optimal cable tensions coming from the MPC-EI algorithm.

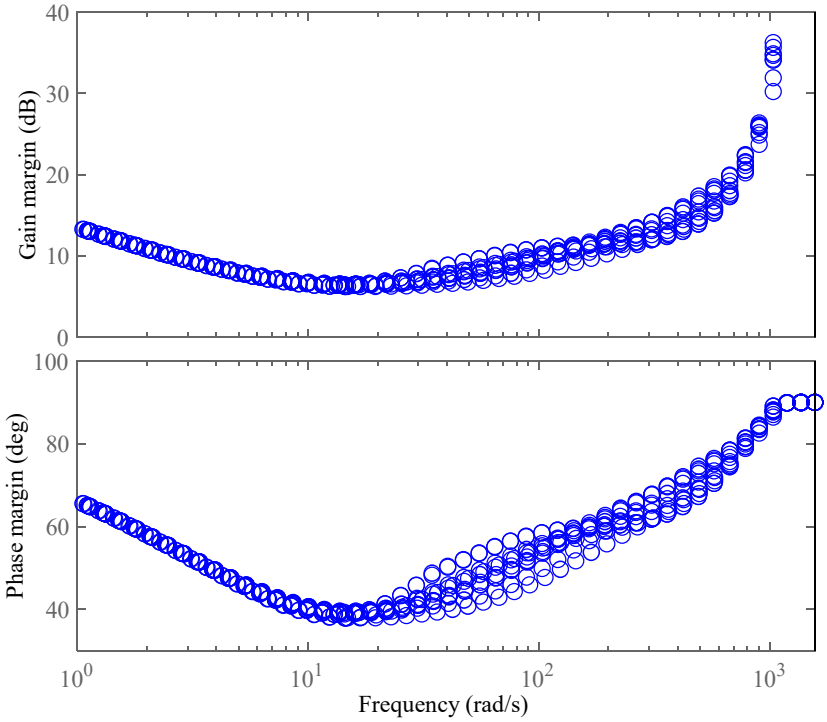


Figure 38. Frequential behaviour of gain and phase margins with spiral reference.

4.5. Conclusions

This Chapter proposes a novel controller architecture for path tracking control in kinematically-determined, non-redundant Cable Driven Parallel Robots, that exploits the idea of Model Predictive Control (MPC). To handle the highly nonlinear dynamic model of a CDPR with a small computational effort, for boosting the real time control, two sequential actions have been included. The first action computes positive and bounded cable tensions through a constrained nonlinear (pose-dependent) MPC with embedded integrator (MPC-EI) that minimizes a performance index that includes the error in tracking the reference and the tension variation. The use of an embedded

integrator has been included as well, to ensure accurate path tracking with small gains. The second control action translates the optimal tensions into the required motor torques and hence into commanded currents for the motor current-loops. The presence of penalizations on large tensions and on large variations in time of the command tensions, allows effectively exploiting the proposed control scheme to control CDPRs and CSPRs with axially-flexible cables, without the need of a model explicitly representing such an elastic behaviour.
Three numerical test-cases have been proposed through a three-cable spatial CSPR with a lumped end-effector, showing good performances in path tracking tasks and ensuring the positiveness of the cable tensions. To better understand the advantages coming from the MPC-EI, the proposed approach has been also compared with a classical MPC formulation, without embedded integrator, which represents an easier and more common algorithm. By looking at the contour error, it has been noticed that MPC-EI has been capable to reduce it by one order of magnitude, ensuring at the same time the verification of the boundaries on all cable tensions and, therefore, suggesting superior performance.
The application of the controller to the model of a CSPR with very elastic cables, whose axial stiffness has been taken by the literature, shows that an almost-negligible deterioration of the path tracking performances is achieved, thanks to the wise controller design. In the case of controlling a flexible CSPR, that is an underactuated system, a collocated control architecture is adopted, meaning that just the positions and speeds at the motor sides are measured, and then translated into positions and speeds of the end-effector through a forward kinematics based on a rigid-cable model. The use of a collocated control, as well-known in the field of vibration control, increases the stability margins and hence allows handling the non-ideal dynamics of sensors and actuators.
The analysis of the computational effort shows that, although optimizing the code was not the goal of this work, the CPU time at each time step is remarkably smaller than the time step of the control loop.

5. Precise Path and Trajectory Tracking in Cable Driven Parallel Robots through Model Predictive Control with Embedded Reference Dynamics

The outcomes of the previous Chapter highlights that Model Predictive Controllers represent a very good candidate for achieving high performances in Cable Driven Parallel Robots, also in the presence of …

This chapter is not shown because the related content is not published yet.

5.1. Introduction

5.1.1. Motivations and state-of-the-art

5.1.2. Contributions of this Chapter

5.2. System model

5.3. The control scheme

5.3.1. Control overview

5.3.2. The state-space model with the difference variables

5.3.3. Model for the prediction

5.3.4. Solution of a standard MPC problem

5.3.5. Embedding of the reference dynamics

5.3.6. Autonomous state-space model of the reference trajectory

5.3.7. Introduction of constraints

5.3.7.1. Overview of the constraints

5.3.7.2. Constraint on the feasible positive tensions

5.3.7.3. Constraint on tension derivatives

5.3.7.4. Constraint on motor torques

5.3.8. Solution of the constrained MPC problem

5.3.9. Computation of the motor torques and currents

5.4. Numerical validation

5.4.1. General description

5.4.2. Controller stability margins

5.4.3. Test case 1: trajectory tracking with a step reference

5.4.4 Test case 2: trajectory tracking with planar circumference

5.4.4.1 Comparison with a classic MPC without embedded integrator

5.4.4.2 Comparison with a MPC with just the embedded integrator

5.4.4.3 Tracking in the presence of severe motor constraints

5.4.5. Test case 3: trajectory tracking of a spatial spiral

5.5. Conclusions

6. Stable Inverse Dynamics for Feedforward Control of Nonminimum-phase Underactuated Systems

Unless advanced controllers, such as the proposed MPC-ERD, are used, the results of Chapter 3 and Chapter 4 have highlighted the fundamental need to use feedforward control in order to improve the performances of the closed-loop systems. The solution of inverse dynamics problem is usually trivial for fully-actuated systems but, unfortunately, it is not straightforward for underactuated ones. Therefore, to overcome the limitations that affect this research field, an enhanced inverse dynamics approach is presented in this Chapter for feedforward control of underactuated multibody systems, such as mechanisms or robots where the number of independent actuators is smaller than the number of degrees of freedom. The method exploits the concept of partitioning the independent coordinates into actuated and unactuated ones (through a *QR-decomposition*) and of linearly combined output, to obtain the internal dynamics of the nonminimum-phase system and then to stabilize it through proper output redefinition. Then, the exact algebraic model of the actuated sub-system is inverted, by leading to the desired control forces with just minor approximations and no need of pre-actuation.
The effectiveness of the proposed approach is assessed by three numerical test cases, by comparing it with some meaningful benchmarks taken from the literature. Finally, experimental verification through an underactuated robotic arm with two degrees of freedom is performed.

6.1. Introduction

6.1.1. Motivations and state-of-the-art

Precise trajectory tracking is a relevant research topic in the field of mechanisms, multibody systems and robotics. To achieve it, controllers exploiting both feedforward (open-loop) and feedback (closed-loop) actions are usually adopted: the first contribution is aimed at ensuring faster responses with lower tracking errors and low feedback gains (and hence higher robustness margins); the second one is aimed at compensating for model-plant mismatches, at changing the dynamic properties (e.g., the eigenstructure [68,137,138]) and at ensuring disturbance rejection. As for feedforward, its implementation requires the computation of the inverse dynamic model of the system under investigation. In the case of fully actuated mechanisms and robots, this computation is usually straightforward and just relies on simple algebraic calculations. On the contrary, when the number of independent actuators is smaller than the number of degrees of freedom (DOF), i.e., underactuated multibody systems are considered, the computation of inverse dynamics is challenging and often relies on the solution of both algebraic and differential equations. The explicit, algebraic solution is possible if the system is differentially flat [139]. In differentially flat systems, inverse dynamics is solved through an algebraic model, and the control forces are algebraically specified by the required output trajectory. Therefore, no pre and post actuations are required. Flatness is a system property that is not straightforward to assess, i.e., it requires a lot of algebraic manipulations and finding a flat output definition is not trivial (see e.g. [104]), since no systematic approach exists. Additionally, exploiting flatness in underactuated multibody systems imposes very smooth reference trajectories, usually with four continuous derivatives [140]. For example, up to 9^{th}-degree polynomials are usually adopted in the literature. In contrast, high-degree polynomials are not widely adopted in motion planning of mechatronic systems because of large requirements of speed, acceleration (both peak and average) and hence of torque, power and energy, that make these motion profiles often unfeasible when considering limits of the components (in particular, motor and gearbox [134,141,142]).
The difficulties in inverting the dynamic model of underactuated multibody systems are exacerbated for some definitions of desired output, such as in the case of noncolocation of actuators and controlled outputs [2], that lead to unstable internal dynamics. In the case of linear systems, for example, this is related to the presence of right half-plane zeros: by inverting the model to compute the suitable control

forces to track the desired output trajectory, these zeros would become right half-plane poles, causing the divergence of the control actions. Generally speaking, with reference to both linear and nonlinear systems, the solution of the inverse dynamic problem of these so called nonminimum-phase systems leads to unbounded solutions, unless noncausal inversion is adopted [41], or approximated causal solutions are sought [2].
To overcome these difficulties, and given the relevance of the topic in the field of motion control, several papers have been proposed in the recent years. An inversion-based approach to exact nonlinear output tracking control is proposed in [41] where a feedback term is introduced to stabilize the system along the desired trajectory; the problem is solved by applying the Byrnes-Isidori regulator to a specific trajectory, introducing boundedness and integrability requirements. However, the solution is noncausal, i.e., the control forces begin before the starting of the trajectory, and therefore this technique results to be not implementable through an online procedure. In [143] an extension of stable inversion to nonlinear time-varying systems is carried on, considering both minimum-phase and nonminimum-phase systems. This technique consists of linearizing the system dynamics and subsequently partitioning it into time-varying stable and unstable sub-systems, which are used to design time-varying filters that ensure a bounded inverse input-state trajectory. However, this approach is local to the time-varying path and, furthermore, it requires that internal dynamics vary slowly. An interesting analysis on the concept of approximated model inversion for nonminimum-phase systems is proposed in [144], with reference to linear systems. Three stable approximate model inversion techniques are examined: the nonminimum-phase zeros ignore (NPZ-Ignore), the zero-phase-error tracking controller (ZPETC) and the zero-magnitude-error tracking controller (ZMETC). Since the proper choice of one of these techniques highly depends on the system under investigation, this work also discusses how the position of the zeros can indicate the more effective technique to be used to maximize the performances. When dealing with nonlinear systems, an effective and simple approach is the output redefinition, i.e., the approximation of the desired output with a formulation that ensures simpler problem solution and stable internal dynamics. This idea has been firstly introduced in the milestone paper [2], and then also exploited with a different formulation in [3] and [145]. In both the papers, an algebraic-differential scheme is adopted, where the differential part can be easily solved through a standard numerical scheme for integration of ODEs (ordinary differential equations) representing the internal dynamics. A remarkably different approach is proposed in [4], by handling dynamic models formulated either with ODEs or with DAEs (differential algebraic equations), and without requiring the explicit derivation of the internal dynamics model. In such a paper, inverse dynamics is formulated as an optimal control problem, and a non-causal solution is obtained.

6.1.2. Contributions of this Chapter

In this Chapter a feedforward computation algorithm for nonminimum-phase underactuated multibody systems, modeled through ODEs, is proposed by exploiting and extending the idea of linearly combined output adopted in [2,3,145]. Compared to the usual formulation of such an approximation, a more general formulation of the linearly combined output is handled in this work. The goal of such an output representation is to easily obtain the internal dynamics for stabilizing it and then to obtain a stable model inversion, without pre-actuation. The method includes a systematic approach to stabilize the unstable internal dynamics, by means of performed through proper output redefinition. The resulting differential scheme is solved through the forward dynamics of the stabilized internal dynamics, by exploiting any scheme for ODE numerical integration. Then, the approximated formulation of the controlled output is replaced by its actual nonlinear formulation (at both the position, speed and acceleration levels) to compute the commanded value of the actuated variables and to perform exact algebraic inversion of the dynamic model governing the sub-system made by the actuated coordinated. Since no output approximation is done in solving such an algebraic model, the proposed method provides an improvement compared to the existing methods: the

approximation assumed through the linearly combined and redefined output is just used in the internal dynamics and hence causes small errors in a large variety of underactuated multibody and robotic systems.
Finally, to tackle different topologies of actuations, an approach for partitioning the model coordinates into actuated and unactuated ones is proposed, by exploiting the *QR-decomposition* of the input matrix, that allows handling a large variety of underactuated systems.
The method effectiveness is proved through three test cases, aimed at highlighting different features of the proposed method also by providing fair comparison with several benchmarks.
Additionally, the experimental verification through a 2-DOF underactuated robotic arm is proposed, thus providing a further improvement of the literature which usually just provides numerical applications.
As a result, a comprehensive, and computationally simple, method for precise and accurate inverse dynamics is formulated, that can be applied to several underactuated systems, by improving the existing state-of-the-art.

6.2. Method description

6.2.1. Definitions

Let us consider the dynamic model of a multibody system, in the presence of holonomic constraints and with n DOFs, formulated through a set of n ODEs:

$$\mathbf{M}_z(\mathbf{z})\ddot{\mathbf{z}} + \mathbf{c}_z(\dot{\mathbf{z}},\mathbf{z}) + \mathbf{K}_z(\mathbf{z})\mathbf{z} = \mathbf{f}_z(\dot{\mathbf{z}},\mathbf{z},t) + \mathbf{B}_z(\mathbf{z})\mathbf{u} \tag{100}$$

$\mathbf{z} \in \mathbb{R}^n$ is the vector of the independent generalized coordinates, $\mathbf{M}_z(\mathbf{z}) \in \mathbb{R}^{n\times n}$ is the mass matrix, $\mathbf{c}_z(\dot{\mathbf{z}},\mathbf{z}) \in \mathbb{R}^n$ contains the Coriolis terms, $\mathbf{K}_z(\mathbf{z}) \in \mathbb{R}^{n\times n}$ is the stiffness matrix and $\mathbf{f}_z(\dot{\mathbf{z}},\mathbf{z},t) \in \mathbb{R}^n$ are the external forces projected on the directions of $\mathbf{z}$. t denotes the time; along the Chapter, the dependence on t is usually not written, unless the cases where it makes the comprehension clearer.
The system is controlled through $m < n$ independent control forces collected in vector $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{B}_z \in \mathbb{R}^{n\times m}$ is the force distribution matrix, that is assumed to be full column-rank; such matrix can be dense, since this work will propose a technique to handle arbitrary topologies of $\mathbf{B}_z$.
Since $\text{rank}(\mathbf{B}_z) = m < n$, the system is said to be underactuated. In underactuated systems, the dimension of the configuration space is greater than the one of the control input space.
The output vector $\mathbf{y} \in \mathbb{R}^m$ is defined in accordance with the goal of control, by assuming a number of independent outputs equal to the number of independent control inputs that makes the map $\mathbf{h}_z : \mathbb{R}^n \mapsto \mathbb{R}^m$ invertible:

$$\mathbf{y} = \mathbf{h}_z(\mathbf{z}) \tag{101}$$

The goal of inverse dynamics is to compute the time history of the control input $\mathbf{u}(t)$ to make $\mathbf{y}(t)$ track the desired output trajectory $\mathbf{y}^{\text{des}}(t) \in \mathbb{R}^m$ as closely as possible. It is assumed that $\mathbf{y}^{\text{des}}(t)$ is at least twice differentiable, as usually done in motion planning of flexible systems ([98,99]).
It is also assumed that the system initial conditions $\mathbf{z}(0)$ and $\dot{\mathbf{z}}(0)$ are known, as well as the external forces $\mathbf{f}_z$.
In fully actuated systems, i.e., when $n = m$, model inversion is straightforward and just relies on simple algebraic calculations by taking advantages of the inversion of matrix $\mathbf{B}_z$. In contrast, in the case of underactuated systems, $\mathbf{B}_z$ is rectangular and cannot be inverted; therefore, model inversion should rely on more complicated solutions, as the one discussed in the following of this Chapter.

6.2.2. Coordinate partitioning

In the general case where $\mathbf{B}_z$ is a dense matrix with more than m not-null entries, the model in Eq. (100) does not allow to immediately partition the coordinates into actuated and unactuated ones (e.g. coordinates permutation is not an effective solution). A coordinate transformation should be therefore performed by exploiting the *QR-decomposition* of $\mathbf{B}_z$. It yields an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ ($\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix of proper dimension) and an upper triangular matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$. $\mathbf{B}$ is upper-triangular, hence it can be written through its sub-matrix $\mathbf{B}_A \in \mathbb{R}^{m \times m}$ which collects all its non-null entries and, moreover, is nonsingular, i.e., $\text{rank}(\mathbf{B}_A) = \text{rank}(\mathbf{B}_z) = m$:

$$\mathbf{B}_z = \mathbf{Q}\mathbf{B} = \mathbf{Q}\begin{bmatrix} \mathbf{B}_A \\ \mathbf{0} \end{bmatrix} \tag{102}$$

where $\mathbf{0}$ denotes a null matrix of proper dimension. A new vector of the generalized coordinates $\mathbf{q} \in \mathbb{R}^n$ is introduced, such that:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_U \end{bmatrix} = \mathbf{Q}^T\mathbf{z} \tag{103}$$

Although the entries of $\mathbf{q}$ have sometimes no straightforward physical interpretation, this transformation partitions the vector of the generalized coordinates into m actuated coordinates, $\mathbf{q}_A \in \mathbb{R}^m$, and $n-m$ unactuated coordinates, $\mathbf{q}_U \in \mathbb{R}^{n-m}$.

In the following it is assumed that $\mathbf{Q}$ is constant, otherwise the definition of $\mathbf{q}$ should change during the motion or in time. In contrast, $\mathbf{B}_A$ can depend on $\mathbf{q}$: this is the occurrence solved in this work, that is a common situation in several multibody and robotic systems (as shown through the three examples). Nonetheless, this technique provides a more general formulation than the papers in the literature, such as [2,3,145], that just handle models that are already partitioned.

Pre-multiplying Eq.(100) by $\mathbf{Q}^T$ and introducing the new coordinate vector $\mathbf{q}$, the system dynamic model is transformed into the following one:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\dot{\mathbf{q}},\mathbf{q}) + \mathbf{K}(\mathbf{q})\mathbf{q} = \mathbf{f}(\dot{\mathbf{q}},\mathbf{q},t) + \mathbf{B}(\mathbf{q})\mathbf{u} \tag{104}$$

Partitioning the matrices in Eq.(100) accordingly to the entries of $\mathbf{q}$, it yields:

$$\begin{bmatrix} \mathbf{M}_{AA}(\mathbf{q}) & \mathbf{M}_{AU}(\mathbf{q}) \\ \mathbf{M}_{AU}^T(\mathbf{q}) & \mathbf{M}_{UU}(\mathbf{q}) \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{q}}_A \\ \ddot{\mathbf{q}}_U \end{bmatrix} + \begin{bmatrix} \mathbf{c}_A(\dot{\mathbf{q}},\mathbf{q}) \\ \mathbf{c}_U(\dot{\mathbf{q}},\mathbf{q}) \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{AA}(\mathbf{q}) & \mathbf{K}_{AU}(\mathbf{q}) \\ \mathbf{K}_{AU}^T(\mathbf{q}) & \mathbf{K}_{UU}(\mathbf{q}) \end{bmatrix}\begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_U \end{bmatrix} = \\ = \begin{bmatrix} \mathbf{f}_A(\dot{\mathbf{q}},\mathbf{q},t) \\ \mathbf{f}_U(\dot{\mathbf{q}},\mathbf{q},t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_A(\mathbf{q}) \\ \mathbf{0} \end{bmatrix}\mathbf{u} \tag{105}$$

where the entries of $\mathbf{M}$, $\mathbf{c}$, $\mathbf{K}$ and $\mathbf{f}$ have the following dimensions: $\mathbf{M}_{AA}, \mathbf{K}_{AA} \in \mathbb{R}^{m \times m}$, $\mathbf{M}_{AU}, \mathbf{K}_{AU} \in \mathbb{R}^{m \times (n-m)}$, $\mathbf{M}_{AU}^T, \mathbf{K}_{AU}^T \in \mathbb{R}^{(n-m) \times m}$, $\mathbf{M}_{UU}, \mathbf{K}_{UU} \in \mathbb{R}^{(n-m) \times (n-m)}$, $\mathbf{c}_A, \mathbf{f}_A \in \mathbb{R}^m$ and $\mathbf{c}_U, \mathbf{f}_U \in \mathbb{R}^{(n-m)}$. The output equation in Eq.(101) is modified as well, by expressing $\mathbf{y}$ as a function of the new coordinate vector $\mathbf{q}$ through function $\mathbf{h}: \mathbb{R}^n \mapsto \mathbb{R}^m$

$$\mathbf{y} = \mathbf{h}(\mathbf{q}) \tag{106}$$

The first m equations of the model in Eq. (105) represent the dynamics of the actuated sub-system, that is directly affected by the control forces $\mathbf{u}$ through the input matrix $\mathbf{B}_A$ and by the external forces $\mathbf{f}_A$. Conversely, the remaining $n-m$ equations represent the dynamics of the unactuated

coordinates that are forced by the external disturbance forces $\mathbf{f}_U$ and are indirectly controlled by the evolution of the actuated coordinates:

$$\mathbf{M}_{UU}\ddot{\mathbf{q}}_U + \mathbf{c}_U + \mathbf{K}_{UU}\mathbf{q}_U = \mathbf{f}_U - \mathbf{M}_{AU}^T\ddot{\mathbf{q}}_A - \mathbf{K}_{AU}^T\mathbf{q}_A \tag{107}$$

Equation (107) is a set of second-order nonholonomic constraints [146], and hence cannot be used to reduce the system model dimension by removing some generalized coordinates. In order to be feasible, any required trajectory should satisfy such constraints at any time. For this reason, the number of independent outputs has been assumed to be equal to the number of independent control inputs.

6.2.3. Approximation of the internal dynamics through linearly combined output

A simple but effective approach to simplify the mechanical and control design of underactuated nonminimum-phase multibody systems, that has been proposed by several authors in the literature [2,3,145], is to approximate the desired output as a linear combination of $\mathbf{q}$. Compared to the mentioned literature, a more general form is proposed in this Chapter, leading to the following formulation:

$$\mathbf{y} = \mathbf{\Gamma}_A\mathbf{q}_A + \mathbf{\Gamma}_U\mathbf{q}_U \tag{108}$$

where $\mathbf{\Gamma}_A \in \mathbb{R}^{m\times m}$ and $\mathbf{\Gamma}_U \in \mathbb{R}^{m\times(n-m)}$. Both matrices are assumed to be full rank.

This use of linear approximations of the output map accomplishes several tasks: on the one hand it easily allows formulating the inverse dynamics; on the other one, it can be exploited to stabilize the internal dynamics whenever it is unstable. The tracking errors that could because of such an approximation are treated, in practice, as those due to the unavoidable mismatches between the model and the dynamics of the actual plant, and hence compensated by the necessary feedback control schemes. Indeed, as mentioned in the introduction, feedforward control should be supported by feedback control if precise motion is required.

On the other hand, Eq. (108) might degrade performances in some particular systems where such an approximation is rough; nonetheless, the use of local approximations, such as piecewise-linear models, overcomes this issue.

The time-history of the actuated coordinates (in terms of position $\mathbf{q}_A(t)$, speed $\dot{\mathbf{q}}_A(t)$ and acceleration $\ddot{\mathbf{q}}_A(t)$), just for the purpose of solving the internal dynamics, can be written as functions of the desired output $\mathbf{y}^{\mathbf{des}}(t)$ and its derivatives (that are expressed through the desired motion law in time), and of the unactuated coordinates by exploiting Eq. (108):

$$\begin{aligned}
\mathbf{q}_A &= \mathbf{\Gamma}_A^{-1}\left(\mathbf{y}^{\mathbf{des}} - \mathbf{\Gamma}_U\mathbf{q}_U\right)\\
\dot{\mathbf{q}}_A &= \mathbf{\Gamma}_A^{-1}\left(\dot{\mathbf{y}}^{\mathbf{des}} - \mathbf{\Gamma}_U\dot{\mathbf{q}}_U\right)\\
\ddot{\mathbf{q}}_A &= \mathbf{\Gamma}_A^{-1}\left(\ddot{\mathbf{y}}^{\mathbf{des}} - \mathbf{\Gamma}_U\ddot{\mathbf{q}}_U\right)
\end{aligned} \tag{109}$$

and the dynamics of the unactuated DOFs in Eq. (107) can be written accordingly by removing $\mathbf{q}_A$ and its derivatives, leading to the system internal dynamics:

$$\begin{aligned}
&\mathbf{M}_{UU}\ddot{\mathbf{q}}_U + \mathbf{c}_U + \mathbf{K}_{UU}\mathbf{q}_U =\\
&= \mathbf{f}_U - \mathbf{M}_{AU}^T\mathbf{\Gamma}_A^{-1}\left(\ddot{\mathbf{y}}^{\mathbf{des}} - \mathbf{\Gamma}_U\ddot{\mathbf{q}}_U\right) - \mathbf{K}_{AU}^T\mathbf{\Gamma}_A^{-1}\left(\mathbf{y}^{\mathbf{des}} - \mathbf{\Gamma}_U\mathbf{q}_U\right)
\end{aligned} \tag{110}$$

The term in Eq. (110) are not constant since $\mathbf{M}_{UU} = \mathbf{M}_{UU}\left(\mathbf{y}^{\mathbf{des}}, \mathbf{q}_U\right)$, $\mathbf{c}_U = \mathbf{c}_U\left(\dot{\mathbf{y}}^{\mathbf{des}}, \dot{\mathbf{q}}_U, \mathbf{y}^{\mathbf{des}}, \mathbf{q}_U\right)$, $\mathbf{f}_U = \mathbf{f}_U\left(\dot{\mathbf{y}}^{\mathbf{des}}, \dot{\mathbf{q}}_U, \mathbf{y}^{\mathbf{des}}, \mathbf{q}_U, t\right)$, $\mathbf{M}_{AU}^T = \mathbf{M}_{AU}^T\left(\mathbf{y}^{\mathbf{des}}, \mathbf{q}_U\right)$, $\mathbf{K}_{AU}^T = \mathbf{K}_{AU}^T\left(\mathbf{y}^{\mathbf{des}}, \mathbf{q}_U\right)$ (these dependencies have been omitted in Eq. (110) for brevity of representation).

The stability of the internal dynamics in Eq. (110) strongly depends on the chosen output $\mathbf{y}$. Usually, the presence of noncolocated pairs of control forces and desired output leads to zeros of the input-output normal form having positive real part.
If a causal solution of the inverse dynamics problem of a nonminimum-phase system is sought, i.e., no pre actuation is wanted, stabilization of the internal dynamics should be done to ensure bounded internal dynamics, and hence bounded control forces. A stable approximation of the unstable system should be therefore properly designed. An effective approach is the output redefinition, that is further approximating $\mathbf{y}$ within the internal dynamics so that Eq.(110) is stable, leading to the following output to be tracked:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{\Gamma}}_{\mathbf{A}}\mathbf{q}_{\mathbf{A}} + \tilde{\mathbf{\Gamma}}_{\mathbf{U}}\mathbf{q}_{\mathbf{U}} \tag{111}$$

$\tilde{\mathbf{\Gamma}}_{\mathbf{A}}$ and $\tilde{\mathbf{\Gamma}}_{\mathbf{U}}$ should be properly chosen to ensure stability with negligible perturbation on the other dynamics. The choice of $\tilde{\mathbf{\Gamma}}_{\mathbf{A}}$ and $\tilde{\mathbf{\Gamma}}_{\mathbf{U}}$ will be discussed during the numerical assessment of the method. In this Chapter, output redefinition is just used with the goal of stabilizing the internal dynamics, i.e., in the differential part of the computation.
Once the stabilizing $\tilde{\mathbf{\Gamma}}_{\mathbf{A}}$ and $\tilde{\mathbf{\Gamma}}_{\mathbf{U}}$ are chosen, Eq. (110) is in turn recast in the form of a nonlinear, stable, dynamic system forced by the external forces (that are supposed to be known) and by the output reference:

$$\left(\mathbf{M}_{\mathbf{UU}} - \mathbf{M}_{\mathbf{AU}}^{T}\tilde{\mathbf{\Gamma}}_{\mathbf{A}}^{-1}\tilde{\mathbf{\Gamma}}_{\mathbf{U}}\right)\ddot{\mathbf{q}}_{\mathbf{U}} + \mathbf{c}_{\mathbf{U}} + \left(\mathbf{K}_{\mathbf{UU}} - \mathbf{K}_{\mathbf{AU}}^{T}\tilde{\mathbf{\Gamma}}_{\mathbf{A}}^{-1}\tilde{\mathbf{\Gamma}}_{\mathbf{U}}\right)\mathbf{q}_{\mathbf{U}} = \mathbf{f}_{\mathbf{U}} - \mathbf{M}_{\mathbf{AU}}^{T}\tilde{\mathbf{\Gamma}}_{\mathbf{A}}^{-1}\ddot{\mathbf{y}}^{\mathbf{des}} - \mathbf{K}_{\mathbf{AU}}^{T}\tilde{\mathbf{\Gamma}}_{\mathbf{A}}^{-1}\mathbf{y}^{\mathbf{des}} \tag{112}$$

The ODEs in Eq.(112) can be solved by forward integration to compute $\ddot{\mathbf{q}}_{\mathbf{U}}(t)$, $\dot{\mathbf{q}}_{\mathbf{U}}(t)$ and $\mathbf{q}_{\mathbf{U}}(t)$, i.e. the time history of unactuated coordinates in the execution of the desired motion profile $\ddot{\mathbf{y}}^{\mathbf{des}}(t)$, in the presence of the external forces in $\mathbf{f}_{\mathbf{U}}(t)$. The selection of the discretization time step and of the numerical scheme adopted for integration relies on the usual rules adopted in the numerical simulations of multibody systems (see e.g. [100]).

6.2.4. Solution of the algebraic equation for the actuated coordinates

Once $\mathbf{q}_{\mathbf{U}}(t)$, $\dot{\mathbf{q}}_{\mathbf{U}}(t)$ and $\ddot{\mathbf{q}}_{\mathbf{U}}(t)$ are computed over the whole cycle time (and will be henceforth denoted as $\mathbf{q}_{\mathbf{U}}^{\mathbf{des}}(t)$, $\dot{\mathbf{q}}_{\mathbf{U}}^{\mathbf{des}}(t)$, $\mathbf{q}_{\mathbf{U}}^{\mathbf{des}}(t)$), the control force vector $\mathbf{u}(t)$ is computed by solving the m algebraic equations representing the inverse dynamics problem of the actuated sub-system. Rather than using again the approximations of the desired output provided by Eqs. (108), (109) and (111), this second step can be performed by inverting the actual map $\mathbf{y}^{\mathbf{des}} = \mathbf{h}\left(\mathbf{q}_{\mathbf{A}}, \mathbf{q}_{\mathbf{U}}\right)$, together with its first and second time derivatives $\dot{\mathbf{y}}^{\mathbf{des}} = \mathbf{H}\left(\mathbf{q}\right)\dot{\mathbf{q}}$ and $\ddot{\mathbf{y}}^{\mathbf{des}} = \mathbf{H}\left(\mathbf{q}_{\mathbf{A}}, \mathbf{q}_{\mathbf{U}}\right)\ddot{\mathbf{q}} + \dot{\mathbf{H}}\left(\mathbf{q}_{\mathbf{A}}, \mathbf{q}_{\mathbf{U}}, \dot{\mathbf{q}}_{\mathbf{A}}, \dot{\mathbf{q}}_{\mathbf{U}}\right)\dot{\mathbf{q}}$, to compute the reference trajectory for the actuated coordinates $\mathbf{q}_{\mathbf{A}}^{\mathbf{des}}(t)$, $\dot{\mathbf{q}}_{\mathbf{A}}^{\mathbf{des}}(t)$, $\ddot{\mathbf{q}}_{\mathbf{A}}^{\mathbf{des}}(t)$, where $\mathbf{H}\left(\mathbf{q}_{\mathbf{A}}, \mathbf{q}_{\mathbf{U}}\right) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix relating the desired output with the actuated and unactuated coordinates, while $\dot{\mathbf{H}}\left(\mathbf{q}_{\mathbf{A}}, \mathbf{q}_{\mathbf{U}}, \dot{\mathbf{q}}_{\mathbf{A}}, \dot{\mathbf{q}}_{\mathbf{U}}\right) \in \mathbb{R}^{m \times n}$ indicates its time derivative. This original approach, proposed in this Chapter, partially compensates for the approximation introduced by the linear approximation, as well as by its stabilizing form in Eq. (111). By defining as $\mathbf{g} : \mathbb{R}^{n} \mapsto \mathbb{R}^{m}$ such an inverse map, the motion of the actuated coordinates at the position level is inferred as follows:

$$\mathbf{q}_{\mathbf{A}}^{\mathbf{des}} = \mathbf{g}\left(\mathbf{y}^{\mathbf{des}}, \mathbf{q}_{\mathbf{U}}\right) \tag{113}$$

Eq. (113) is an algebraic, and usually nonlinear, vectorial equations that can be often analytically solved by exploiting trigonometric relations, or numerically through well-established and reliable

methods. The inversion at the velocity and acceleration levels are, in contrast, linear, once Eq. (113) is solved

$$\dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{des}} = \mathbf{G}\left(\mathbf{y}^{\mathrm{des}}, \mathbf{q}_{\mathrm{U}}\right) \begin{bmatrix} \dot{\mathbf{y}}^{\mathrm{des}} \\ \dot{\mathbf{q}}_{\mathrm{U}} \end{bmatrix} \tag{114}$$

$$\ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{des}} = \mathbf{G}\left(\mathbf{y}^{\mathrm{des}}, \mathbf{q}_{\mathrm{U}}\right) \begin{bmatrix} \ddot{\mathbf{y}}^{\mathrm{des}} \\ \ddot{\mathbf{q}}_{\mathrm{U}} \end{bmatrix} + \dot{\mathbf{G}}\left(\mathbf{y}^{\mathrm{des}}, \mathbf{q}_{\mathrm{U}}, \dot{\mathbf{y}}^{\mathrm{des}}, \dot{\mathbf{q}}_{\mathrm{U}}\right) \begin{bmatrix} \dot{\mathbf{y}}^{\mathrm{des}} \\ \dot{\mathbf{q}}_{\mathrm{U}} \end{bmatrix} \tag{115}$$

$\mathbf{G}$ is the Jacobian matrix of $\mathbf{g}$, i.e., $\dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{des}} = \dfrac{\partial \mathbf{g}\left(\mathbf{y}^{\mathrm{des}}, \mathbf{q}_{\mathrm{U}}\right)}{\partial \mathbf{y}^{\mathrm{des}}} \dot{\mathbf{y}}^{\mathrm{des}} + \dfrac{\partial \mathbf{g}\left(\mathbf{y}^{\mathrm{des}}, \mathbf{q}_{\mathrm{U}}\right)}{\partial \mathbf{q}_{\mathrm{U}}} \dot{\mathbf{q}}_{\mathrm{U}}$.

By imposing such references into the first m equations in Eq. (105), that represent the forced dynamics of the actuated sub-system, the required control forces can be inferred from the following algebraic computation, that provides exact inversion of the actuated part of the dynamic model:

$$\mathbf{u}_{\mathrm{FFW}}(t) = \mathbf{B}_{\mathrm{A}}^{-1}\left(\mathbf{M}_{\mathrm{AA}}\ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{des}} + \mathbf{M}_{\mathrm{AU}}\ddot{\mathbf{q}}_{\mathrm{U}}^{\mathrm{des}} + \mathbf{c}_{\mathrm{A}} + \mathbf{K}_{\mathrm{AA}}\mathbf{q}_{\mathrm{A}}^{\mathrm{des}} + \mathbf{K}_{\mathrm{AU}}\mathbf{q}_{\mathrm{A}}^{\mathrm{des}} - \mathbf{f}_{\mathrm{A}}\right) \tag{116}$$

where $\mathbf{M}_{\mathrm{AA}} = \mathbf{M}_{\mathrm{AA}}\left(\mathbf{q}^{\mathrm{des}}\right)$, $\mathbf{M}_{\mathrm{AU}} = \mathbf{M}_{\mathrm{AU}}\left(\mathbf{q}^{\mathrm{des}}\right)$, $\mathbf{c}_{\mathrm{A}} = \mathbf{c}_{\mathrm{A}}\left(\dot{\mathbf{q}}^{\mathrm{des}}, \mathbf{q}^{\mathrm{des}}\right)$, $\mathbf{K}_{\mathrm{AA}} = \mathbf{K}_{\mathrm{AA}}\left(\mathbf{q}^{\mathrm{des}}\right)$, $\mathbf{K}_{\mathrm{AU}} = \mathbf{K}_{\mathrm{AU}}\left(\mathbf{q}^{\mathrm{des}}\right)$, $\mathbf{f}_{\mathrm{A}} = \mathbf{f}_{\mathrm{A}}\left(\dot{\mathbf{q}}^{\mathrm{des}}, \mathbf{q}^{\mathrm{des}}, t\right)$.

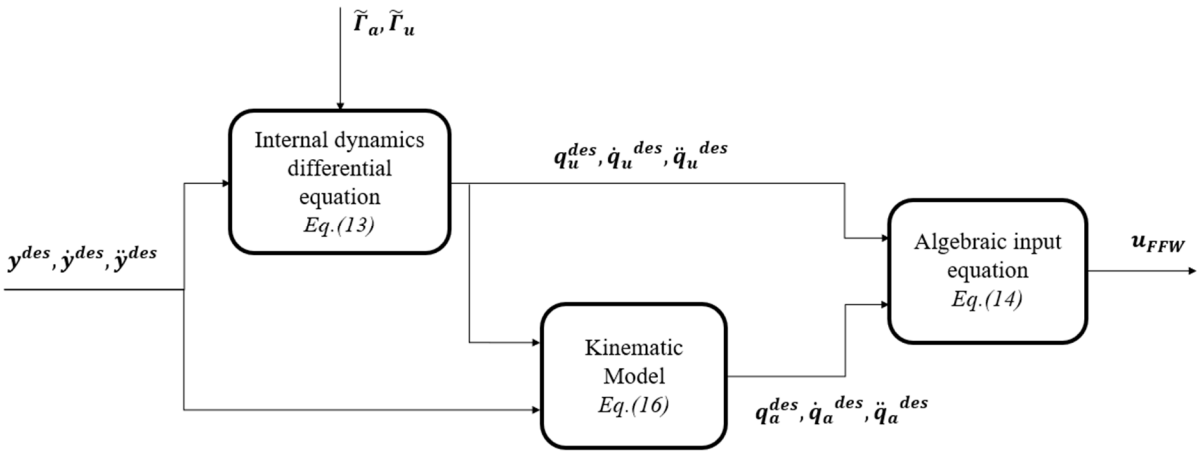


Figure 39. Block diagram of the method for computing the feedforward force.

Besides computing the feedforward control force, the proposed method can be exploited to generate consistent references to the actuated variables ($\mathbf{q}_{\mathrm{A}}^{\mathrm{des}}(t), \dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{des}}(t)$) that can be fed to feedback control loop (or loops in case of decoupled feedback controllers) to compute the closed loop corrections required to compensate for the unavoidable model mismatches.

6.3. Test case 1: flexible beam with lumped masses

6.3.1. Description and system model

The first test case is a linear system composed by a flexible beam, modeled through the finite-element approach, plus some lumped spring-damper-mass systems, that recall the model of linear vibratory feeders often adopted in manufacturing plants to convey small components [147]. Despite the

apparent simplicity of the system, due to its linearity, the proposed test case has several challenging issues to be properly tackled.

The system is represented in Figure 40. It consists of a flexible beam modeled through four Euler-Bernoulli finite elements (each one with length 0.75 m, flexural rigidity $EJ = 1.93\times10^{5}$ N$\cdot$m^{2} and linear mass density $\rho A = 22.87$ kg$\cdot$m^{-1}). The beam is connected to the frame through two springs ($k_l = k_r = 1.5\times10^{5}$ N$\cdot$m^{-1}). Three actuators exert three pairs of independent forces, as shown in Figure 40, and are modeled through three lumped masses ($m_{ai} = 23$ kg, with $i = 1,\ldots,3$) connected to the beam by means of three linear springs (with stiffness $k_{ai} = 4.6e5$ N$\cdot$m^{-1}, with $i = 1,\ldots,3$). The resulting model has 13 DOFs, which include 5 absolute transversal translations (η_j, with $j = 1,\ldots,5$), 5 absolute rotations (φ_j, with $j = 1,\ldots,5$) of the beam and the 3 absolute translations of the actuators ($\eta_{a_1}, \eta_{a_2}, \eta_{a_3}$):

$$\mathbf{z} = \begin{bmatrix} \eta_1 & \varphi_1 & \eta_2 & \varphi_2 & \eta_3 & \varphi_3 & \eta_4 & \varphi_4 & \eta_5 & \varphi_5 & \eta_{a_1} & \eta_{a_2} & \eta_{a_3} \end{bmatrix}^T \tag{117}$$

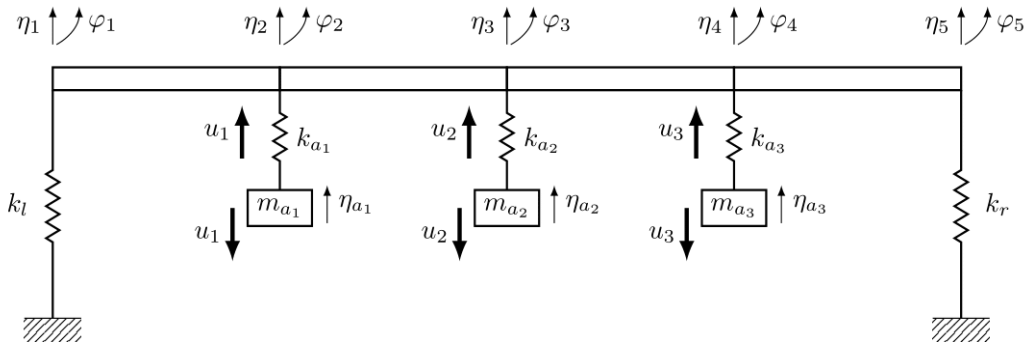


Figure 40. Sketch of the linear vibratory feeder.

A linear model is obtained, that recalls the usual form of a second order vibrating system [148]:

$$\mathbf{M}_{\mathbf{z}}\,\ddot{\mathbf{z}} + \mathbf{C}_{\mathbf{z}}\,\dot{\mathbf{z}} + \mathbf{K}_{\mathbf{z}}\,\mathbf{z} = \mathbf{B}_{\mathbf{z}}\,\mathbf{u} \tag{118}$$

where $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \in \mathbb{R}^3$ is the vector collecting the three independent control forces. $\mathbf{M}_{\mathbf{z}} \in \mathbb{R}^{13\times13}$ and $\mathbf{K}_{\mathbf{z}} \in \mathbb{R}^{13\times13}$ are obtained by assembling the mass and stiffness sub-matrices of the finite elements. The damping matrix $\mathbf{C}_{\mathbf{z}} \in \mathbb{R}^{13\times13}$ is obtained through the well-known Rayleigh model, imposing low viscous friction coefficients at each damped spring with the goal to fairly reproduce an underdamped real case scenario.

The definition of the vector of the physical displacements $\mathbf{z} \in \mathbb{R}^{13}$ leads to the following dense matrix $\mathbf{B}_{\mathbf{z}} \in \mathbb{R}^{13\times3}$, that does not directly partition the dynamics into actuated and unactuated coordinates:

$$\mathbf{B}_{\mathbf{z}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}^T \tag{119}$$

The equation of motion of the system in the transformed coordinates $\mathbf{q} \in \mathbb{R}^{13}$ is achieved through *QR decomposition* of $\mathbf{B}_{\mathbf{z}} = \mathbf{Q}\,\mathbf{B}$, leading to:

$$\mathbf{M}\ddot{\mathbf{q}}+\mathbf{C}\dot{\mathbf{q}}+\mathbf{K}\mathbf{q}=\mathbf{B}\mathbf{u} \tag{120}$$

where $\mathbf{M}=\mathbf{Q}^T\mathbf{M}_z\mathbf{Q}\in\mathbb{R}^{13\times13}$, $\mathbf{C}=\mathbf{Q}^T\mathbf{C}_z\mathbf{Q}\in\mathbb{R}^{13\times13}$, $\mathbf{K}=\mathbf{Q}^T\mathbf{K}_z\mathbf{Q}\in\mathbb{R}^{13\times13}$ and $\mathbf{B}\in\mathbb{R}^{13\times3}$, which is the input distribution matrix of the new system of coordinates.
The following state transformation matrix is obtained:

$$\mathbf{Q}^T=\begin{bmatrix}
0 & 0 & -0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0\\
0 & 0 & 0 & 0 & -0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -0.7 & 0 & 0 & 0 & 0 & 0 & 0.7\\
0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.7 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0\\
0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.5 & 0 & -0.4 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.4 & 0 & 0.5\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0\\
0.7 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0\\
0 & 0.7 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0\\
-0.5 & 0 & 0.4 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.4 & 0 & 0.5
\end{bmatrix} \tag{121}$$

Matrix $\mathbf{Q}^T\in\mathbb{R}^{13\times13}$ in Eq. (121), that maps $\mathbf{z}$ into $\mathbf{q}=\begin{bmatrix}\mathbf{q}_A & \mathbf{q}_U\end{bmatrix}^T$, with $\mathbf{q}_A\in\mathbb{R}^3$ and $\mathbf{q}_U\in\mathbb{R}^{10}$, reveals that the actuated coordinates $\mathbf{q}_A$ represent the relative translations of the nodes where each actuator is placed. In contrast, many coordinates in $\mathbf{q}_U$ have not a clear physical meaning.
The desired output vector $\mathbf{y}^{des}\in\mathbb{R}^3$, here chosen as a meaningful example, includes the transversal displacement of the beam in the 3 attachment nodes:

$$\mathbf{y}^{des}=\begin{bmatrix}\eta_2 & \eta_3 & \eta_4\end{bmatrix}^T \tag{122}$$

Since $\mathbf{y}^{des}$ is a subset of $\mathbf{z}$, by considering the linear state transformation due to the *QR decomposition*, $\mathbf{z}=\mathbf{Q}\mathbf{q}$, it can be easily inferred that $\mathbf{\Gamma}_A\in\mathbb{R}^{3\times3}$ and $\mathbf{\Gamma}_U\in\mathbb{R}^{3\times10}$ are proper sub-matrices of $\mathbf{Q}$. Hence, the use of linearly combined output is an exact representation for the test case under consideration.
Inverse dynamics is exploited to compute the actuator forces that make $\mathbf{y}$ track a periodic, non-sinusoidal, asymmetric reference waveform, as sometimes adopted in feeders [149]. Each period of the wave consists of a rising phase lasting 25 ms, followed by a drop phase of 5 ms, and the absolute value of the peak displacement is 5 mm. The period of oscillation is 35 Hz, as often adopted in feeding of small parts ([148], [150]). The position references of the three controlled output are set with a $40°$ phase shift.

6.3.2. Stabilization of the internal dynamics

The internal dynamics of this system is modeled through the following linear equations:

$$\mathbf{M}_{AU}^T\ddot{\mathbf{q}}_A+\mathbf{M}_{UU}\ddot{\mathbf{q}}_U+\mathbf{C}_{AU}^T\dot{\mathbf{q}}_A+\mathbf{C}_{UU}\dot{\mathbf{q}}_U+\mathbf{K}_{AU}^T\mathbf{q}_A+\mathbf{K}_{UU}\mathbf{q}_U=\mathbf{0} \tag{123}$$

where $\mathbf{M}_{AU}^{T}, \mathbf{C}_{AU}^{T}, \mathbf{K}_{AU}^{T} \in \mathbb{R}^{10\times 3}$ and $\mathbf{M}_{UU}, \mathbf{C}_{UU}, \mathbf{K}_{UU} \in \mathbb{R}^{10\times 10}$. The introduction of the linearly combined output, see Eq.(108), enables to express the actuated coordinates as a function of the desired displacements and the unactuated coordinates, leading to:

$$\mathbf{M}_{ID}\ddot{\mathbf{q}}_{U} + \mathbf{C}_{ID}\dot{\mathbf{q}}_{U} + \mathbf{K}_{ID}\mathbf{q}_{U} = \mathbf{B}_{ID}\mathbf{u}_{ID} \tag{124}$$

where $\mathbf{M}_{ID} = \mathbf{M}_{UU} - \mathbf{M}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1}\,\boldsymbol{\Gamma}_{U} \in \mathbb{R}^{10\times 10}$, $\mathbf{C}_{ID} = \mathbf{C}_{UU} - \mathbf{C}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1}\,\boldsymbol{\Gamma}_{U} \in \mathbb{R}^{10\times 10}$, $\mathbf{K}_{ID} = \mathbf{K}_{UU} - \mathbf{K}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1}\,\boldsymbol{\Gamma}_{U} \in \mathbb{R}^{10\times 10}$ and $\mathbf{B}_{ID} = -\begin{bmatrix}\mathbf{M}_{ID}^{-1}\,\mathbf{K}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1} & \mathbf{M}_{ID}^{-1}\,\mathbf{C}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1} & \mathbf{M}_{ID}^{-1}\,\mathbf{M}_{AU}^{T}\,\boldsymbol{\Gamma}_{A}^{-1}\end{bmatrix} \in \mathbb{R}^{10\times 9}$. The input $\mathbf{u}_{ID} \in \mathbb{R}^{9}$ of the ODEs governing the internal dynamics is the output reference, in terms of position, speed and acceleration:

$$\mathbf{u}_{ID} = \begin{bmatrix}\mathbf{y}^{des}\\ \dot{\mathbf{y}}^{des}\\ \ddot{\mathbf{y}}^{des}\end{bmatrix} \tag{125}$$

The eigenvalue analysis of the internal dynamics reveals that it is unstable, as it can be inferred through Figure 41, because of the presence of poles with positive real parts.

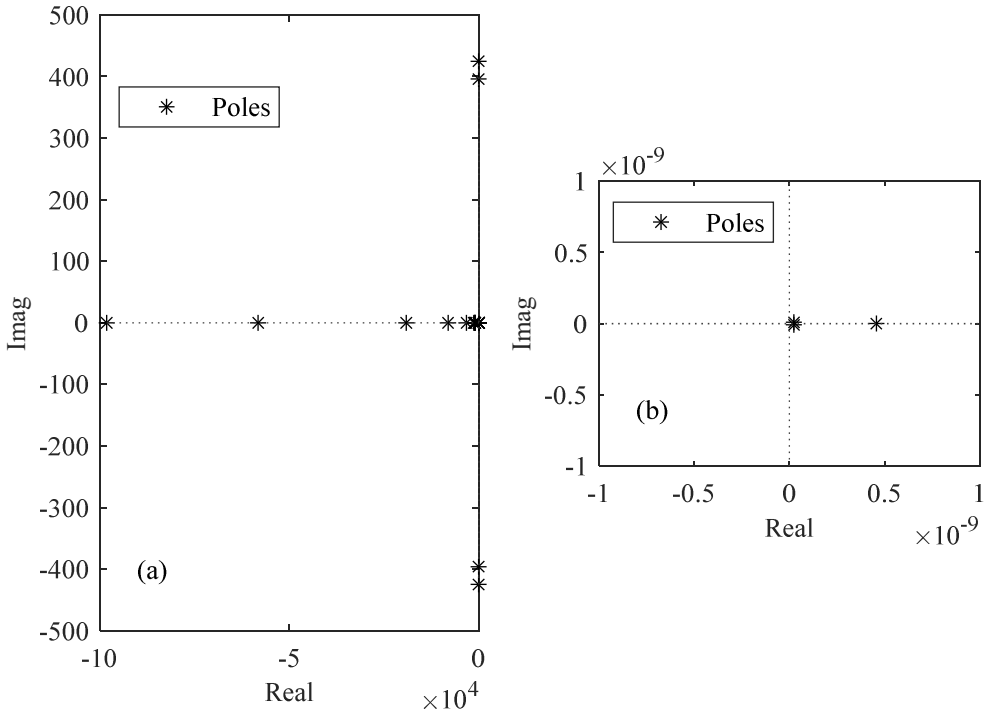


Figure 41. Internal dynamics poles of the vibratory feeder system (a); detail about the imaginary axis (b).

The literature sometimes suggests a simple approach to stabilize nonminimum-phase system by removing from the linear model the nonminimum-phase zeros, and hence the unstable poles of the internal dynamics, leading to the so called "nonminimum-phase zeros ignore" (NPZ-ignore) approaches [144]. Although this option could seem effective at a first glance because the unstable poles are quite far from the reference harmonic components, its application leads to large errors. The results of the application of the NPZ-ignore approach are therefore omitted to keep the work concise. Therefore, the stabilization approach by exploiting approximated values of $\boldsymbol{\Gamma}_{A}$ and $\boldsymbol{\Gamma}_{U}$ in Eq. (111), hereafter denoted $\tilde{\boldsymbol{\Gamma}}_{A}$ and $\tilde{\boldsymbol{\Gamma}}_{U}$, is adopted. Such matrices should be wisely chosen to ensure

asymptotic stability with negligible spillover on the other poles, while allowing for a precise approximation of the tracked output. To simplify the analysis, $\mathbf{\Gamma}_A$ has been kept constant while the approximation of $\mathbf{\Gamma}_U$ is parametrized through just a scalar α, as $\tilde{\mathbf{\Gamma}}_U = \alpha \mathbf{\Gamma}_U$. Figure 42 shows that $\alpha = 1$ is the boundary condition between stable and unstable internal dynamics.

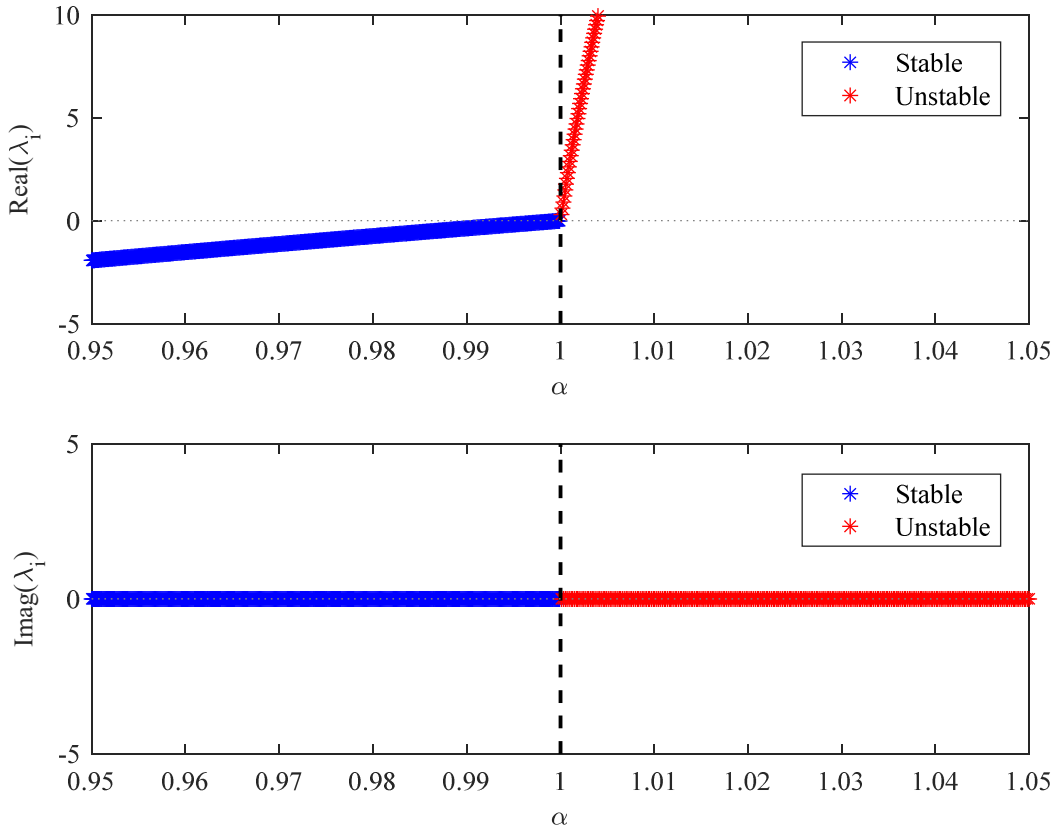


Figure 42. Stability analysis by considering different values of α.

6.3.3. Numerical results

To prevent from instability arising from numerical integration error, it has been assumed $\alpha = 0.99$. The ODEs in Eq.(124) have been integrated through the 4th-order Runge-Kutta scheme with a sampling time of 10 μs, due to presence of some high frequency poles. The time history of the control forces $\mathbf{u}$ is reported in Figure 43, while the tracking responses are reported in Figure 44 together with the corresponding tracking errors. The maximum tracking error is 5.2×10^{-2} mm, that is the 1.0 % of the amplitude of the periodic motion; the average RMS (root mean square) error is instead 2.1×10^{-2} mm. Such errors are due to the output redefinition to stabilize the internal dynamics; additionally, the presence of underdamped dominant dynamics in the system model exacerbates such errors, by creating some residual, although small, oscillations after motion completion.
It should be noted that no pre-actuation is required. In contrast, post-actuation is required after the end of the required motion, with a small force to be exerted by the actuators (and a very small position error that quickly settles) since the system is not differentially flat.

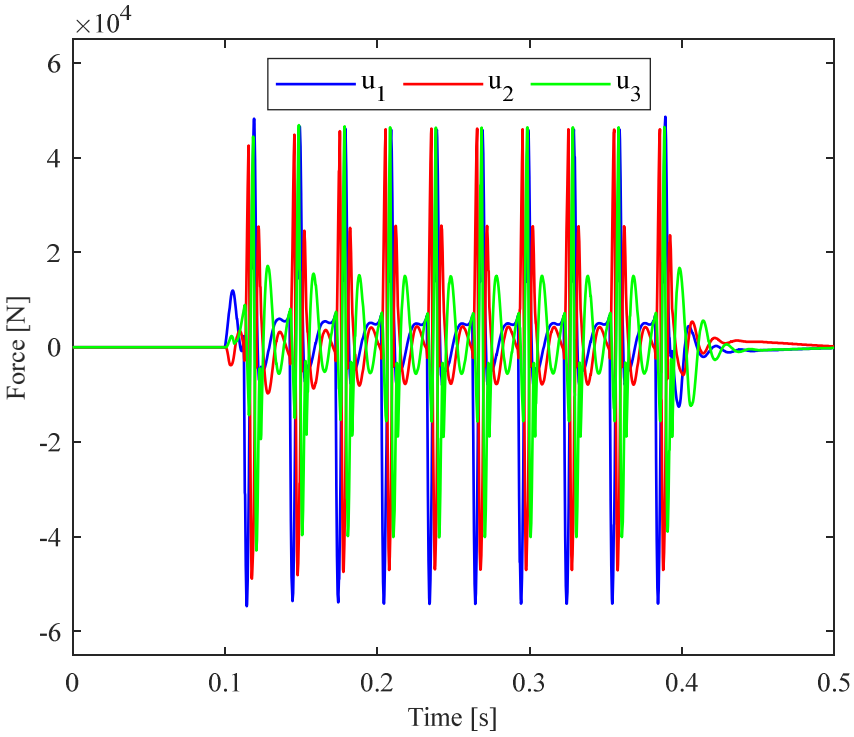


Figure 43. Computed forces.

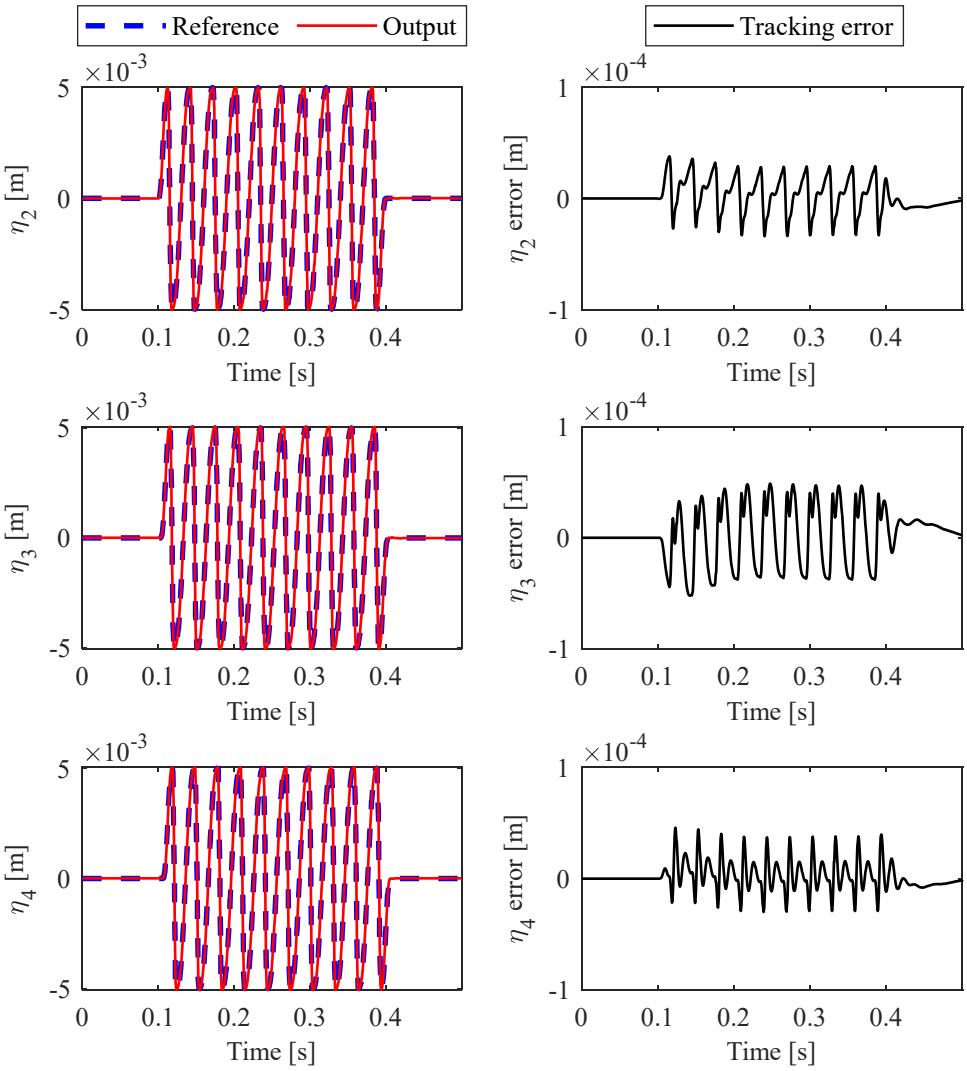


Figure 44. Tracking responses (on the left) and related tracking errors (on the right).

6.4. Test case 2: overhead cartesian crane

6.4.1. Description and system model

The second test-case is an underactuated nonlinear system consisting of a 4-DOF overhead cartesian crane, as sketched in Figure 45. The following independent coordinates are assumed to model the system: two translations describing the planar motion of the platform, that are the two actuated coordinates $\mathbf{q}_\mathbf{A} = \begin{bmatrix} x_P & y_P \end{bmatrix}^T$, and two unactuated components of the swing angle $\mathbf{q}_\mathbf{U} = \begin{bmatrix} \theta_X & \theta_Y \end{bmatrix}^T$. θ_X is the oscillation angle projected on the XZ plane, while θ_Y is the angle measured from the XZ plane. By assuming that the cable is taut with tensile stress and that hoisting is not allowed (i.e., the cable length h is constant), the following four nonlinear equations describe the system dynamic behaviour [151]:

$$\begin{bmatrix} (M_X+m) & 0 & mh\cos(\theta_X)\cos(\theta_Y) & -mh\sin(\theta_X)\sin(\theta_Y) \\ 0 & (M_Y+m) & 0 & mh\cos(\theta_Y) \\ mh\cos(\theta_X)\cos(\theta_Y) & 0 & mh^2\cos^2(\theta_Y) & 0 \\ -mh\sin(\theta_X)\sin(\theta_Y) & mh\cos(\theta_Y) & 0 & mh^2 \end{bmatrix} \begin{bmatrix} \ddot{x}_P \\ \ddot{y}_P \\ \ddot{\theta}_X \\ \ddot{\theta}_Y \end{bmatrix} +$$

$$+ \begin{bmatrix} -mh\dot{\theta}_X^2\sin(\theta_X)\cos(\theta_Y) - mh\dot{\theta}_Y^2\sin(\theta_X)\cos(\theta_Y) - 2mh\dot{\theta}_X\dot{\theta}_Y\cos(\theta_X)\sin(\theta_Y) + c_X\dot{x}_P \\ -mh\dot{\theta}_Y^2\sin(\theta_Y) + c_Y\dot{y}_P \\ -2mh^2\dot{\theta}_X\dot{\theta}_Y\sin(\theta_Y)\cos(\theta_Y) + c_\theta\dot{\theta}_X \\ mh^2\dot{\theta}_X^2\sin(\theta_Y)\cos(\theta_Y) + c_\theta\dot{\theta}_Y \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ 0 \\ -mhg\sin(\theta_X)\cos(\theta_Y) \\ -mhg\cos(\theta_X)\sin(\theta_Y) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

(126)

The following definitions have been introduced in Eq.(126): u_X, u_Y are the forces that are applied to the platform along the X and Y directions; m is the lumped mass of the suspended load; c_X, c_Y, c_θ are the viscous friction coefficients; M_X, M_Y are the translational masses of the platform along the X and Y directions respectively; g is the gravity acceleration. The values of all the parameters of the system model are reported in Table 8.

Table 8. Values of the parameters of the overhead cartesian crane.

Parameter	*Value*
M_X, M_Y	30 kg
m	0.7 kg
h	1 m
c_X, c_Y	0.5 $\mathrm{N\cdot s\cdot m^{-1}}$
c_θ	0.25 $\mathrm{N\cdot m\cdot s\cdot rad^{-1}}$

The goal of feedforward control is to make the load track a prescribed path in the cartesian plane, defined through the load cartesian absolute coordinates $\begin{bmatrix} x_L & y_L \end{bmatrix}^T$. The following output vector is therefore defined:

$$\mathbf{y} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} x_P + h\sin\theta_X\cos\theta_Y \\ y_P + h\sin\theta_Y \end{bmatrix} \tag{127}$$

In this example, both the dynamic model and the output map are nonlinear. The approximation of $\mathbf{y}$ as a linear combination of $\mathbf{q} = \begin{bmatrix} \mathbf{q_A}^T & \mathbf{q_U}^T \end{bmatrix}^T$ is obtained by linearizing Eq. (127) about the vertical equilibrium configuration (i.e. $\theta_X = \theta_Y = 0°$), leading to $\mathbf{\Gamma_A} = \mathbf{I} \in \mathbb{R}^{2\times 2}$ and $\mathbf{\Gamma_U} = h\mathbf{I} \in \mathbb{R}^{2\times 2}$.

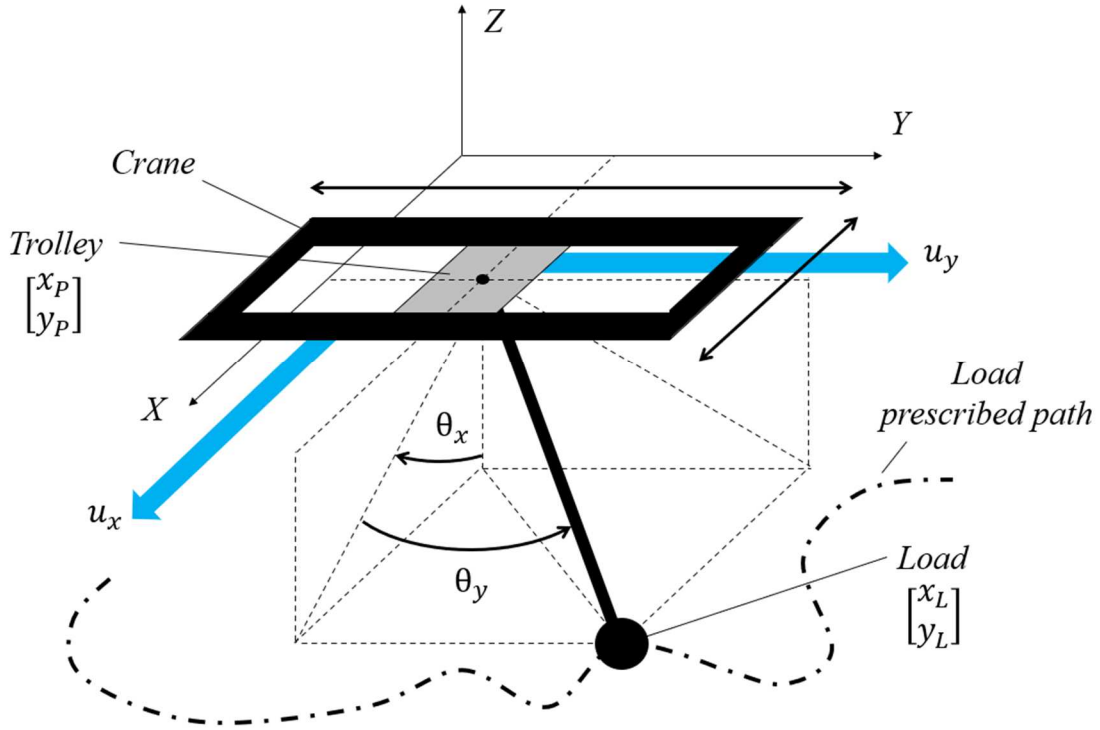


Figure 45. Sketch of the overhead cartesian crane.

6.4.2. Stabilization of the internal dynamics

The dynamic model can be directly partitioned as follows without the state transformation of Eq. (103), where the swing angle components are the unactuated coordinates:

$$\begin{bmatrix} \mathbf{M}_{\mathbf{AA}} & \mathbf{M}_{\mathbf{AU}} \\ \mathbf{M}_{\mathbf{AU}}^T & \mathbf{M}_{\mathbf{UU}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\mathbf{A}} \\ \ddot{\mathbf{q}}_{\mathbf{U}} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_{\mathbf{A}} \\ \mathbf{c}_{\mathbf{U}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{A}} \\ \mathbf{f}_{\mathbf{U}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{\mathbf{A}} \\ \mathbf{B}_{\mathbf{U}} \end{bmatrix} \mathbf{u} \tag{128}$$

where $\mathbf{u} = \begin{bmatrix} u_X & u_Y \end{bmatrix}^T \in \mathbb{R}^2$; $\mathbf{B}_{\mathbf{A}} = \mathbf{I} \in \mathbb{R}^{2\times 2}$ and $\mathbf{B}_{\mathbf{U}} = \mathbf{0} \in \mathbb{R}^{2\times 2}$ ($\mathbf{0}$ denotes a null matrix).

To study the stability of the internal dynamics, the equations of motion of the unactuated sub-system ($\mathbf{M}_{\mathbf{AU}}^T \ddot{\mathbf{q}}_{\mathbf{A}} + \mathbf{M}_{\mathbf{UU}} \ddot{\mathbf{q}}_{\mathbf{U}} + \mathbf{c}_{\mathbf{U}} = \mathbf{f}_{\mathbf{U}}$) are linearized around the stable downward equilibrium point defined by $\theta_X = \theta_Y = 0°$ and, secondly, $\ddot{\mathbf{q}}_{\mathbf{A}}$ is replaced thanks to the linearly combined output technique described in Section 6.2.3, with $\ddot{\mathbf{y}}^{\mathbf{des}} = \ddot{\mathbf{q}}_{\mathbf{A}} + \boldsymbol{\Gamma}_{\mathbf{U}} \ddot{\mathbf{q}}_{\mathbf{U}}$.

If the exact $\boldsymbol{\Gamma}_{\mathbf{U}}$ is used, the internal dynamics is unstable. As in the previous test, stabilization is achieved by simply defining $\tilde{\boldsymbol{\Gamma}}_{\mathbf{U}} = \alpha \boldsymbol{\Gamma}_{\mathbf{U}}$, leading to the following model:

$$\mathbf{M}_{\mathbf{ID}} \ddot{\mathbf{q}}_{\mathbf{U}} + \mathbf{C}_{\mathbf{ID}} \dot{\mathbf{q}}_{\mathbf{U}} + \mathbf{K}_{\mathbf{ID}} \mathbf{q}_{\mathbf{U}} = \mathbf{B}_{\mathbf{ID}} \ddot{\mathbf{y}}^{\mathbf{des}} \tag{129}$$

where $\mathbf{M}_{\mathbf{ID}}, \mathbf{C}_{\mathbf{ID}}, \mathbf{K}_{\mathbf{ID}}, \mathbf{B}_{\mathbf{ID}} \in \mathbb{R}^{2\times 2}$ are defined as follows:

$$\mathbf{M}_{\mathbf{ID}} = \begin{bmatrix} mh^2(1-\alpha) & 0 \\ 0 & mh^2(1-\alpha) \end{bmatrix} \qquad \mathbf{C}_{\mathbf{ID}} = \begin{bmatrix} c_\theta & 0 \\ 0 & c_\theta \end{bmatrix}$$
$$\mathbf{K}_{\mathbf{ID}} = \begin{bmatrix} mhg & 0 \\ 0 & mhg \end{bmatrix} \qquad \mathbf{B}_{\mathbf{ID}} = \begin{bmatrix} -mh & 0 \\ 0 & -mh \end{bmatrix} \tag{130}$$

If the exact $\mathbf{\Gamma}_{\mathbf{U}}$ is used, and therefore $\alpha = 1$, $\mathbf{M}_{\mathbf{ID}}$ becomes a null matrix and therefore the internal dynamics features poles with infinite positive values. To assess the effect of a on the stability, Eq.(129) can be treated as two independent scalar equations by considering that $\mathbf{M}_{\mathbf{ID}}$, $\mathbf{C}_{\mathbf{ID}}$ and $\mathbf{K}_{\mathbf{ID}}$ are diagonal matrices, and the two characteristic polynomials are separately discussed. Since all the entries of $\mathbf{C}_{\mathbf{ID}}$ and $\mathbf{K}_{\mathbf{ID}}$ are positive, the Descartes' Rule of Signs reveals that the internal dynamics is stable if and only if $\alpha < 1$. As in the previous test case, the parameter α should be, therefore, chosen as close as possible to the unitary value, without exceeding it, ensuring a good reconstruction of the desired output and making negligible the pole spillover, while accounting for numerical issues of the ODE solver that can makes the system unstable even with $\alpha < 1$. In the following, it has been chosen $\alpha = 0.99$ that leads to the following poles of the internal dynamics: $\lambda_1 = \lambda_2 = -17.86 + 25.73i$ and $\lambda_3 = \lambda_4 = -17.86 - 25.73i$.

6.4.3. Numerical results

The test consists of a planar circle with a diameter of 0.5 m in the XY plane. The entire circular path is set to be completed in 10 s through a 5^{th}-degree polynomial timing law; to assess the behaviour of the load at steady-state conditions, two idle intervals of 4 s are introduced both at the beginning and at the end of the trajectory. The ODEs in Eq.(124) have been integrated through the 4^{th}-order Runge-Kutta scheme with a sampling time of 10 μs.

The displacements of the load and the trolley are shown in Figure 46, that also provides details on the tracking errors: the reference trajectory is tracked with high precision, with a maximum transient error of 0.11 mm, and with a negligible residual oscillation after motion completion.

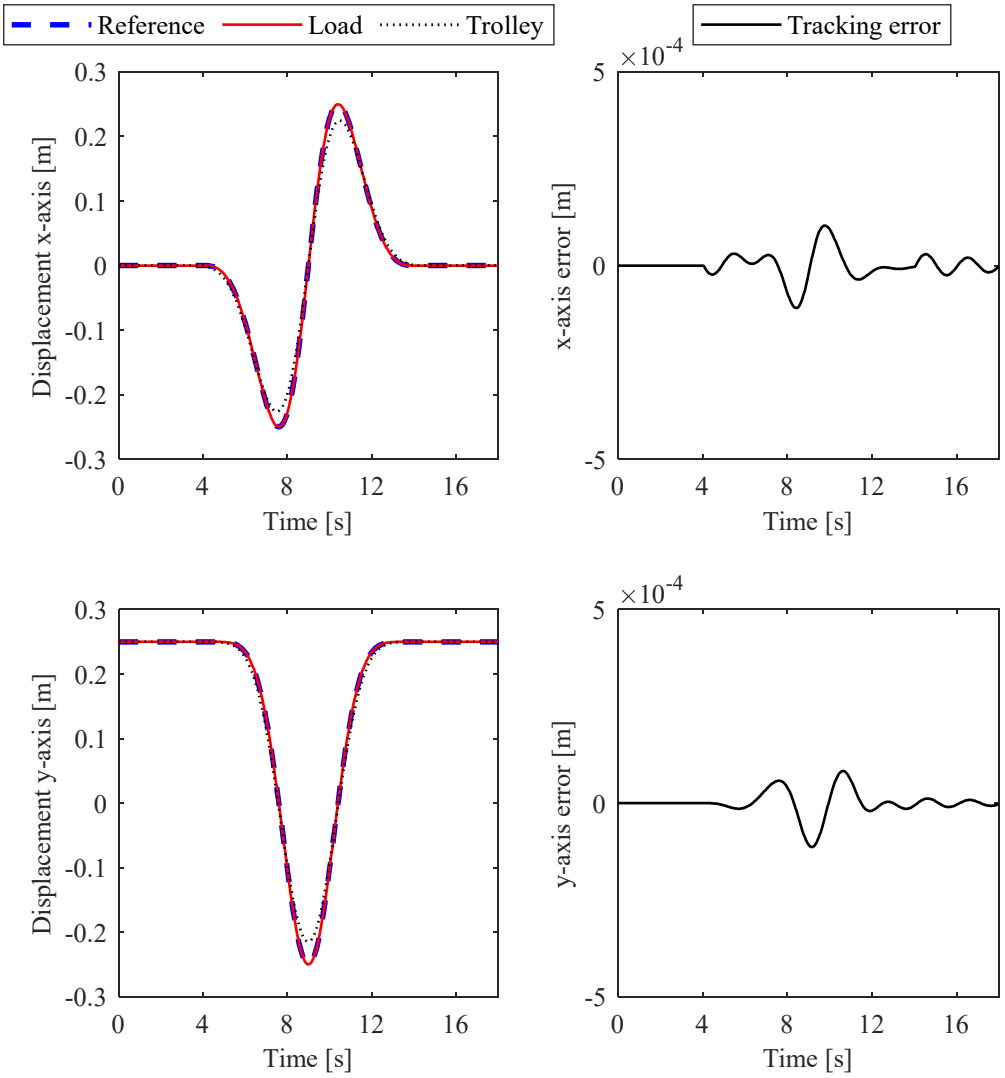


Figure 46. Reference tracking responses (on the left), and related tracking errors (on the right).

The path tracking response is shown in Figure 47. Figure 47.a compares the motion of the trolley with the one of the load, by clearly showing that the proposed technique properly shapes the trolley path (by means of the control forces applied on it), by considering that centrifugal effects will make the load path differ from the one of the load. Figure 47.b details the contour error of the load, with respect to the reference path: the maximum value is 0.12 mm, the RMS error during the motion is 0.04 mm, while the residual error after motion completion is negligible.
Finally, Figure 48 displays the computed forces applied to perform the desired task.

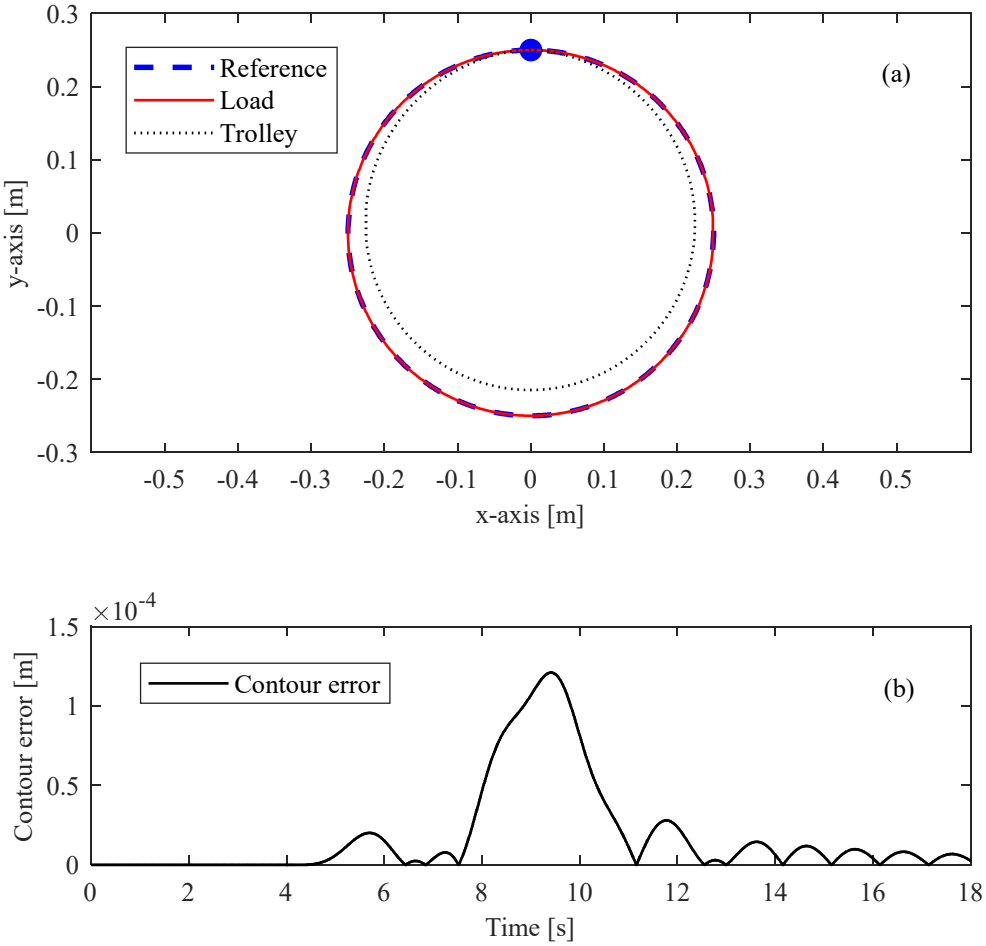


Figure 47. Path tracking response (a), and related contour error (b).

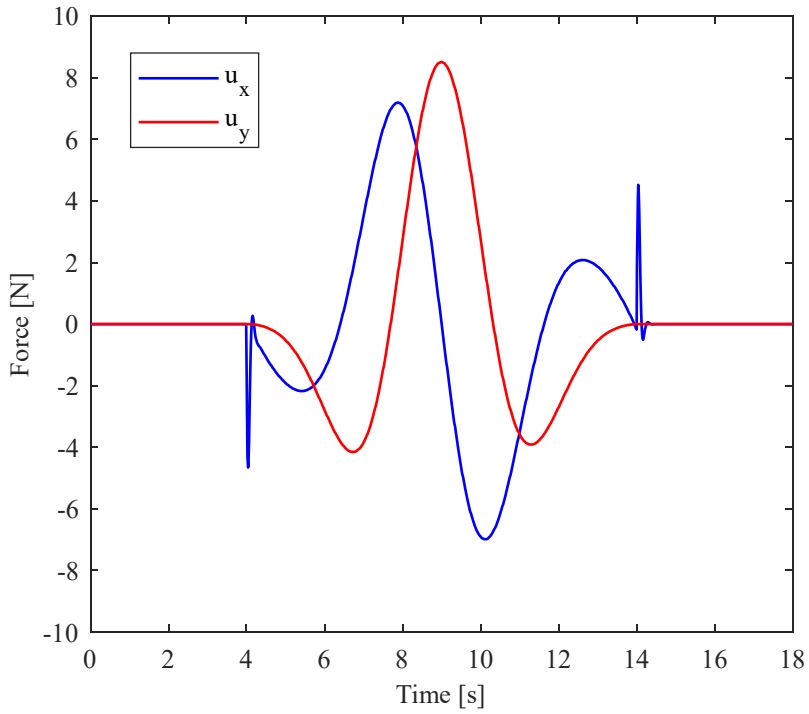


Figure 48. Computed forces.

A final assessment of the capability of the proposed method is obtained by comparing it with the method presented in [3], by adopting the same stabilized linearly combined output. The results are summarized in Figure 49, which shows the tracking and the contour errors sported by the two different ways of calculating $\mathbf{u}_{\mathbf{FFW}}$, by adopting the same stabilization scheme for the internal dynamics. It is evident the benefit of the approach proposed in this Chapter, that remarkably reduces the errors: using [3] leads to a maximum tracking error equal to 0.35 mm, and a maximum contour error equal to 0.35 mm as well. Indeed, due to the centrifugal forces, the mismatch between the non-linear model and the linearized one becomes relevant.

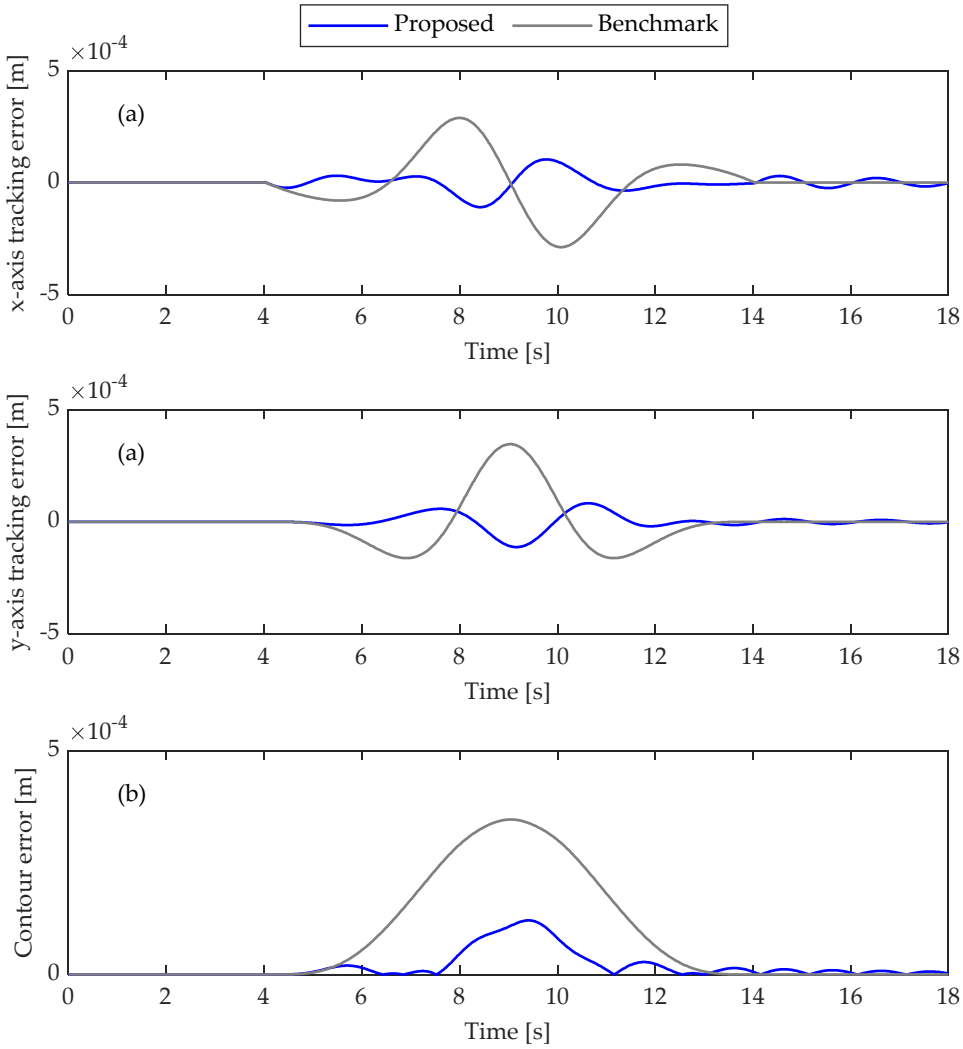


Figure 49. Comparison of the tracking (a) and contour (b) errors sported by proposed method and benchmark [3].

6.5. Test case 3: underactuated robotic arm with flexible passive joint

6.5.1. Description and system model

The third test case is a planar flexible robotic arm lying in the vertical plane, as shown in Figure 50, made by two rigid links connected through a passive revolute joint (B) with a linear torsional spring. Another joint (A) is directly actuated by a DC (direct current) motor. The actuated link is of length l_1, mass m_1 and moment of inertia J_1 (defined with respect of point A). The moment of inertia of the motor, J_m, and of the rigid coupling, J_d, should be accounted for as well. The other link is of length l_2, mass m_2 and a moment of inertia J_2 (defined with respect of point B). A nodal mass m_B is placed at joint B to include the mass contribution of the rotational encoder. Two linear springs couple the two links, resulting into an equivalent rotational spring with rotational stiffness k_s. All the values of the mentioned parameters are reported in Table 9.

The computed torque is commanded by translating it as the reference current through the nominal torque constant of the motor. Such a reference current is fed to the drive current loop through Simulink Real Time scheme that communicates the current to be delivered at each instant to the motor through an external input-output board. It should be noted that the current loop is the sole closed-loop

controller adopted in these experiments: neither the position nor the speed feedback loops are used, since trajectory tracking just relies on an open loop scheme to assess the effectiveness of the proposed approach.

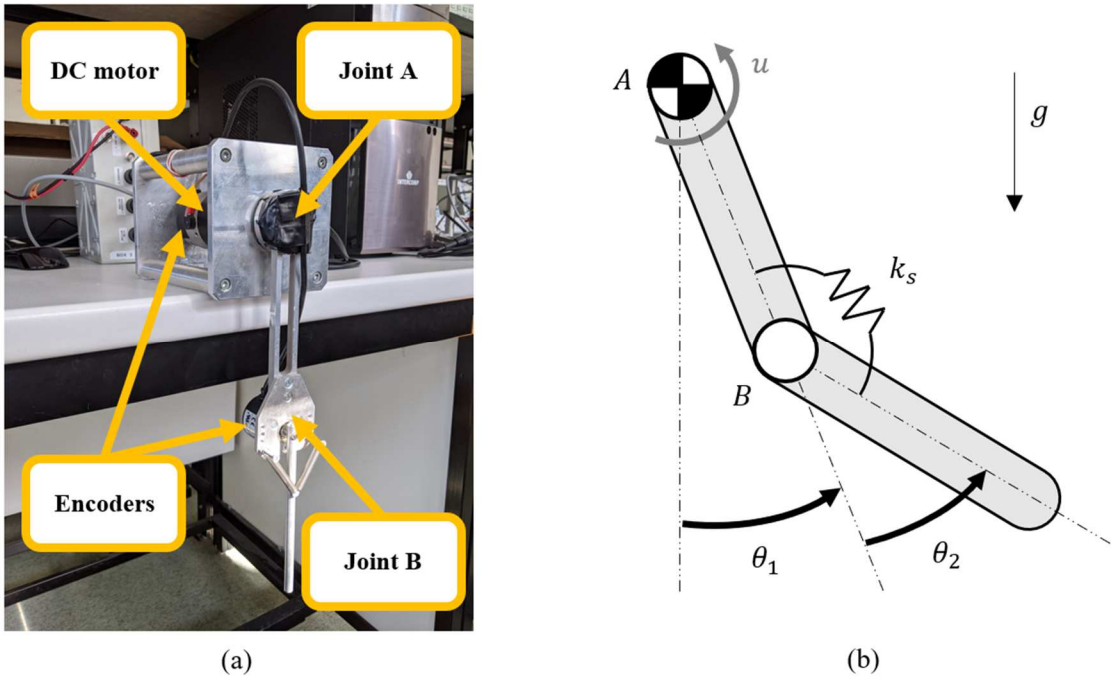


Figure 50. The flexible robotic arm: picture of the experimental setup (a) and model definitions (b)

Table 9. Nominal values of the parameters of the flexible robotic arm.

Parameter	Value	Parameter	Value
l_1	0.17 m	l_2	0.155 m
m_1	0.05 kg	m_2	0.021 kg
J_1	4.82×10^{-4} kg·m^2	J_2	1.68×10^{-4} kg·m^2
J_m	2.7×10^{-5} kg·m^2	J_d	2.3×10^{-5} kg·m^2
m_B	0.178 kg	k_s	0.177 N·m/rad

The two independent coordinates $\mathbf{q}=[\theta_1 \quad \theta_2]^T$ are the absolute rotation of the actuated link (controlled through the motor torque u) with respect to the vertical configuration, and the relative position of the passive link with respect to the actuated one, denoted θ_1 and θ_2 respectively, as shown in Figure 50. The following nonlinear model of the system is formulated:

$$\begin{bmatrix} J_m + J_d + J_1 + (m_2 + m_B) l_1^2 + m_2 l_1 l_2 \cos(\theta_2) + J_2 & \frac{1}{2} m_2 l_1 l_2 \cos(\theta_2) + J_2 \\ \frac{1}{2} m_2 l_1 l_2 \cos(\theta_2) + J_2 & J_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} +$$
$$+ \begin{bmatrix} -\frac{1}{2} m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) + c_m \dot{\theta}_1 \\ \frac{1}{2} m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1^2 + c_s \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ k_s \theta_2 \end{bmatrix} = \tag{131}$$
$$= \begin{bmatrix} -\left(\frac{1}{2} m_1 + m_2 + m_B\right) g l_1 \sin(\theta_1) - \frac{1}{2} m_2 g l_2 \sin(\theta_1 + \theta_2) \\ -\frac{1}{2} m_2 g l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

6.5.2. Stabilization of the internal dynamics

Describing the system through the relative coordinate of the passive link allows to directly partition Eq.(131) as follows:

$$\begin{bmatrix} M_{AA} & M_{AU} \\ M_{AU} & M_{UU} \end{bmatrix} \begin{bmatrix} \ddot{q}_A \\ \ddot{q}_U \end{bmatrix} + \begin{bmatrix} c_A \\ c_U \end{bmatrix} + \begin{bmatrix} k_A \\ k_U \end{bmatrix} = \begin{bmatrix} f_A \\ f_U \end{bmatrix} + \begin{bmatrix} B_A \\ B_U \end{bmatrix} u \tag{132}$$

where $\ddot{q}_A = \ddot{\theta}_1$ and $\ddot{q}_U = \ddot{\theta}_2$. Considering only the unactuated sub-system, its dynamics is described by:

$$M_{AU}\, \ddot{q}_A + M_{UU}\, \ddot{q}_U + c_U + k_U = f_U \tag{133}$$

The tip horizontal displacement is set as the desired output and, to formulate the linearly combined output technique, it is defined through the equivalent rotation of a one-DOF rigid link whose length is $l_1 + l_2$, linearized around the stable downward equilibrium point defined by $\theta_1 = \theta_2 = 0$:

$$y^{des} = \frac{l_1}{l_1 + l_2} \sin(\theta_1) + \frac{l_2}{l_1 + l_2} \sin(\theta_1 + \theta_2) \simeq \theta_1 + \frac{l_2}{l_1 + l_2} \theta_2 = q_A + \Gamma_U\, q_U \tag{134}$$

To study the stability of the internal dynamics, Eq.(133) is also linearized around the same configuration, by obtaining:

$$\left(\bar{M}_{UU} - \bar{M}_{AU}\, \Gamma_U\right) \ddot{q}_U = \bar{f}_U - \bar{c}_U - \bar{k}_U - \bar{M}_{AU}\, \ddot{y}^{des} \tag{135}$$

where $\bar{M}_{UU}$, $\bar{M}_{AU}$, $\bar{f}_U$, $\bar{c}_U$, $\bar{k}_U \in \mathbb{R}$ are the linearized forms of M_{UU}, M_{AU}, f_U, c_U, k_U respectively:

$$\bar{M}_{UU} = J_2 \qquad \bar{M}_{AU} = \frac{1}{2} m_2 l_1 l_2 + J_2$$
$$\bar{f}_U = -\frac{1}{2} m_2 g l_2 (\theta_1 + \theta_2) \qquad \bar{c}_U = c_s \dot{\theta}_2 \tag{136}$$
$$\bar{k}_U = k_s \theta_2$$

Therefore, the linearized internal dynamics is reformulated as follows for stability analysis:

$$M_{ID}\,\ddot{q}_U + C_{ID}\,\dot{q}_U + K_{ID}\,q_U = B_{ID}\,\ddot{y}^{des} \tag{137}$$

where $M_{ID}, C_{ID}, K_{ID}, B_{ID} \in \mathbb{R}$ assume the following meaning:

$$\begin{aligned} M_{ID} &= J_2\left(\left(\frac{3}{2}\frac{l_1}{l_2}+1\right)\Gamma_U - 1\right) \qquad C_{ID} = c_s \\ K_{ID} &= \frac{1}{2}m_2\,g\,l_2 + k_s \qquad B_{ID} = -\frac{1}{2}m_2\,l_1\,l_2 + J_2 \end{aligned} \tag{138}$$

The following eigenvalues $\lambda_{1,2} \in \mathbb{C}$ of the homogeneous characteristic equation associated to Eq.(137) are calculated:

$$\lambda_{1,2} = \frac{-M_{ID}^{-1}\,C_{ID} \pm \sqrt{\left(M_{ID}^{-1}\,C_{ID}\right)^2 - 4M_{ID}^{-1}\,K_{ID}}}{2} \tag{139}$$

$\mathrm{Re}\{\lambda_{1,2}\} < 0$ if and only if $M_{ID}^{-1}\,K_{ID} > 0$ and hence if and only if:

$$\tilde{\Gamma}_U < \frac{2l_2}{3l_1 + 2l_2} \tag{140}$$

To provide a similar treatment as the previous test cases, defining $\tilde{\Gamma}_U = \alpha\,\Gamma_{\lim}$ (with $\alpha \in \mathbb{R}$) and $\Gamma_{\lim} = \dfrac{2l_2}{3l_1 + 2l_2}$ recasts the inequality in (140) as $\alpha < 1$. The same result can be obtained by considering the stability features of the associated zero-dynamics [2], that is unstable in nonminimum-phase systems.

As previously seen, the parameter α should be chosen as close as possible to the unitary value, without exceeding it, ensuring a good reconstruction of the desired output and, at the same time, making the spillover effect of the poles negligible.

As a final consideration, it is useful to highlight that the exact value of Γ_U, defined in Eq.(134), leads to an unstable internal dynamics, as shown in Figure 51, since $\Gamma_U > \Gamma_{\lim}$.

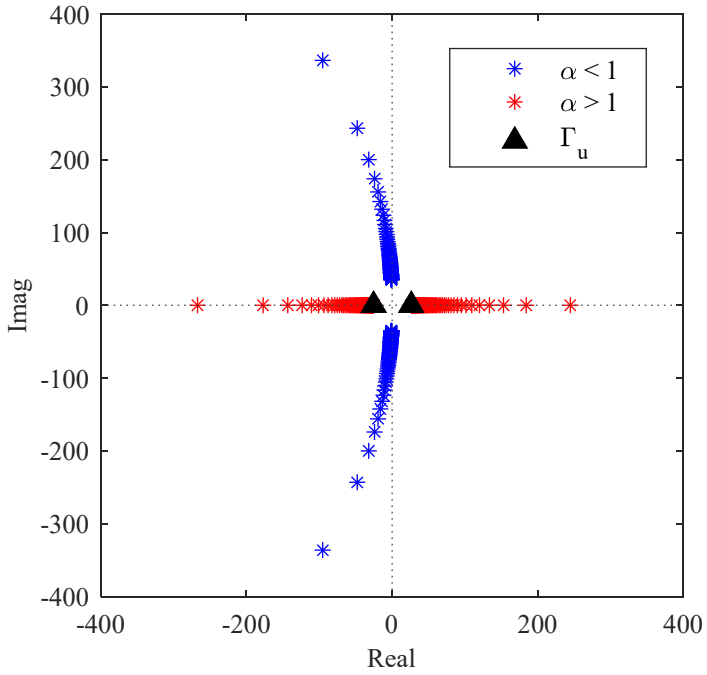


Figure 51. Internal dynamics poles of the flexible robotic arm, with different choices of $\tilde{\Gamma}_U$.

6.5.3. Numerical application

The test consists of a pick-and-place repetitive cycle, from the pick-position to the place one, and then back again to the pick-location. Indeed, pick-and-place tasks have a great importance for robots employed in manufacturing plants [76,77] and hence the end-effector motion should be properly controlled. Two different amplitudes of the motion have been considered, corresponding to two different opening angles of the equivalent rigid single-link mechanism equal to $30°$ and $50°$. The motion time of each displacement is 1.2 s and 5^{th}-degree polynomial timing laws have been assumed. Additionally, rest intervals, lasting 1 s for the $30°$ case and 2 s for the $50°$ one, are required between each motion.

The method application has led to the torques shown in Figure 52, which are continuous and do not require pre-actuation. The ODEs in Eq.(124) have been integrated through the 4^{th}-order Runge-Kutta scheme with a sampling time of 1 ms.

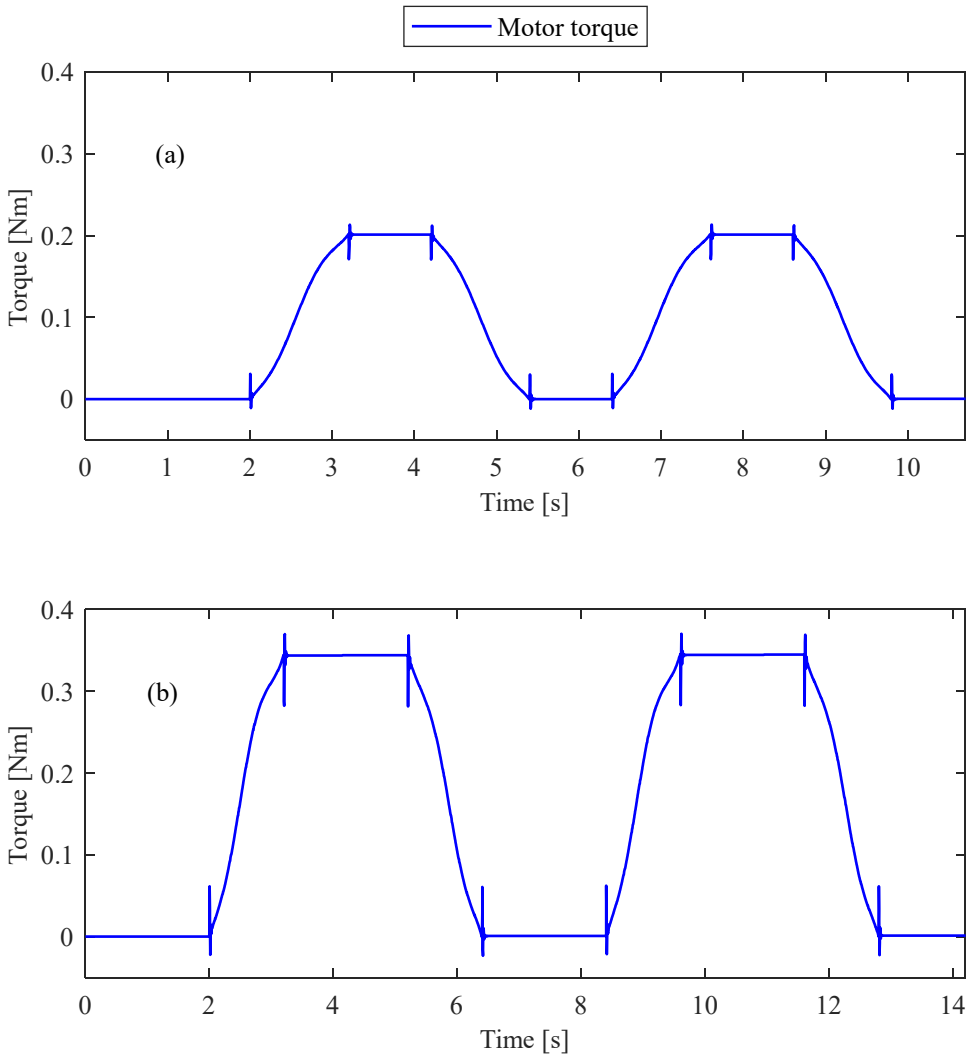


Figure 52. Feedforward motor torques with different opening angles: 30° (a) and 50° (b).

Figure 53 shows the result obtained, by applying the computed torque to the simulator of the nonlinear mechanism: precise tracking is obtained for the three references, and just some small errors are present since references and actual displacement are almost overlapped.

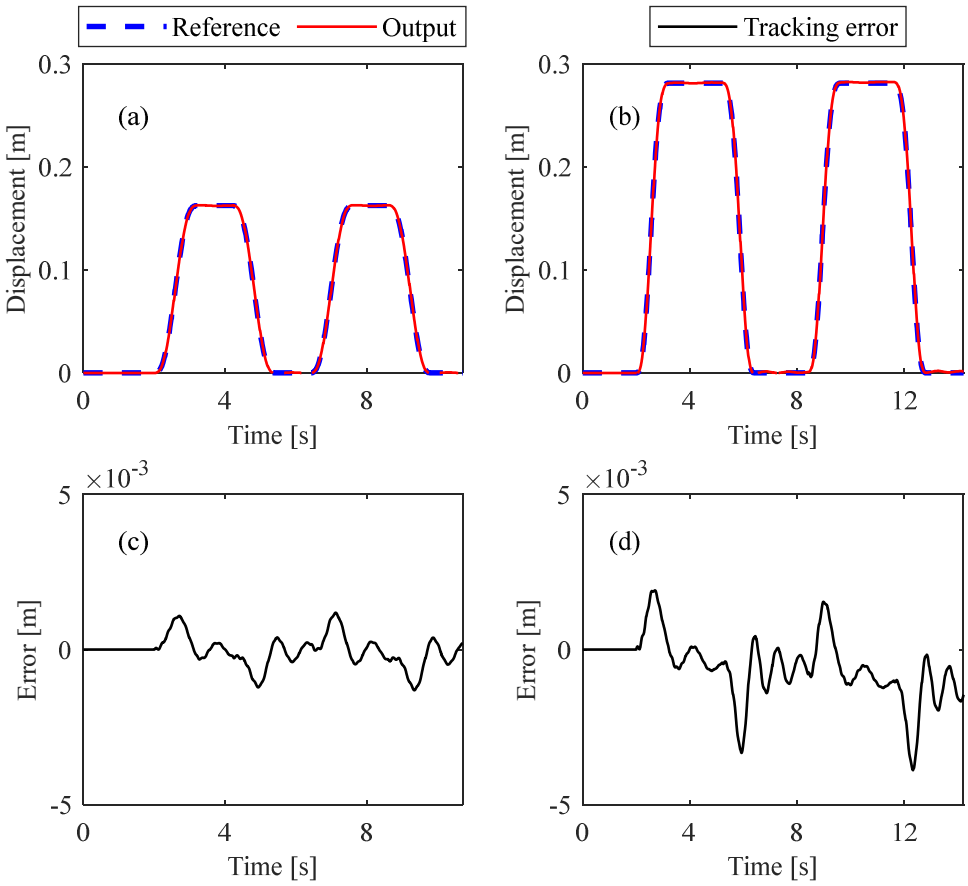


Figure 53. Numerical trajectory tracking responses with their relative tracking errors with different opening angles: 30° (a, c) and 50° (b, d).

A clearer view is provided by the analysis of the tracking error for both the tests and through benchmarking: the outcomes of the proposed method have been compared with those sported by the methods proposed in [2] and in [6], that are assumed as the benchmarks for this test case. In the application of [2], the same output redefinition is adopted to stabilize the internal dynamics. The results are shown in Figure 54, where the tracking errors are detailed to corroborate the effectiveness of the method proposed in this Chapter and the advancement it provides compared with the state-of-the-art methods. In the case of opening angle equal to 30°, the proposed method leads to a maximum error equal to 1.3 mm and to an RMS error equal to 0.45 mm; the benchmark I, i.e. the method proposed in [2], leads to a maximum error equal to 2.1 mm (+62 %) and to an RMS error equal to 0.9 mm (+100 %). Even larger errors are sported by benchmark II, i.e. the method proposed in [6]: the maximum error is equal to 17.6 mm (+1254 %) and the RMS error is 7.1 mm (+1478 %), with visible residual oscillations.

In the presence of the 50° opening angle, the proposed method still outperforms the benchmarks since it leads to a maximum error equal to 2.5 mm and to an RMS error equal to 0.8 mm; benchmark I leads to a maximum error equal to 3 mm (+20 %) and to an RMS error equal to 1.2 mm (+50 %); again, remarkably larger errors are sported by benchmark II: the maximum error is equal to 54.2 mm (+2068 %) and the RMS error is 28.1 mm (+3412 %).

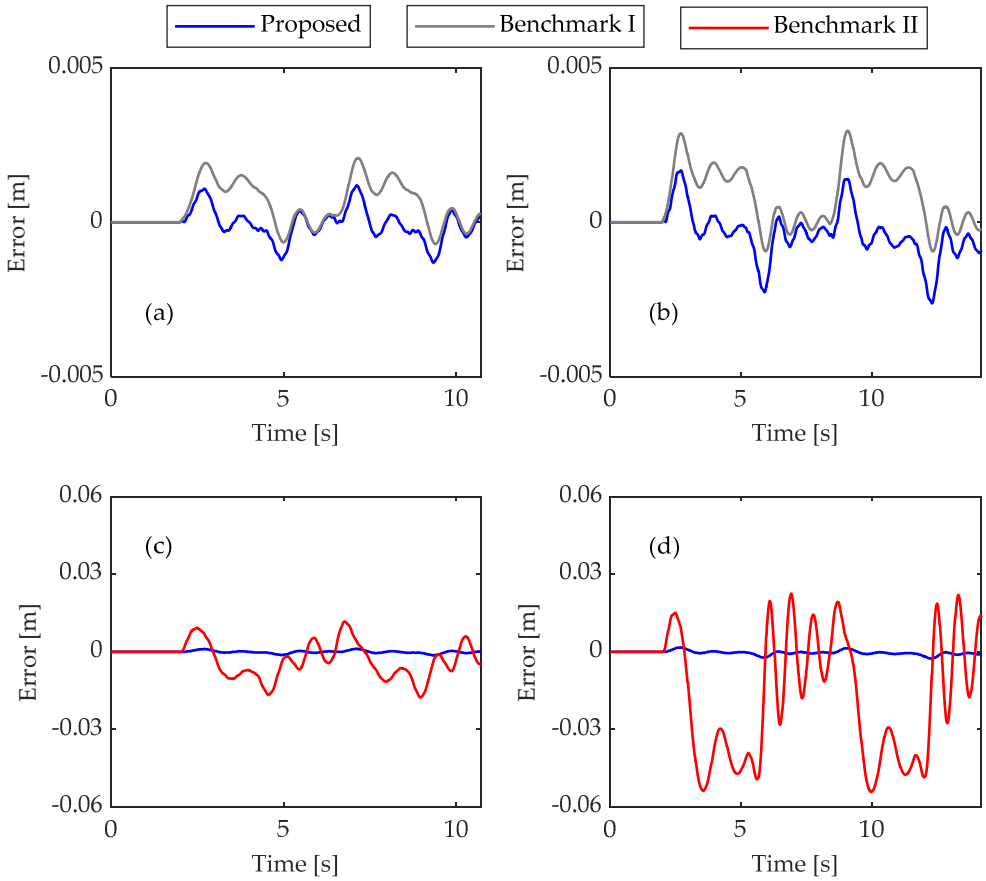


Figure 54. Comparison of the tracking errors of the proposed method and the two benchmarks, with different opening angles: 30° (a, c) and 50° (b, d).

Finally, the comparison of the computed torques is provided in Figure 55: very similar values are provided since all these methods are aimed at best approximating the exact torques; in particular, benchmark I, that is the most accurate one between the two benchmarks, is almost overlapped to the proposed method.

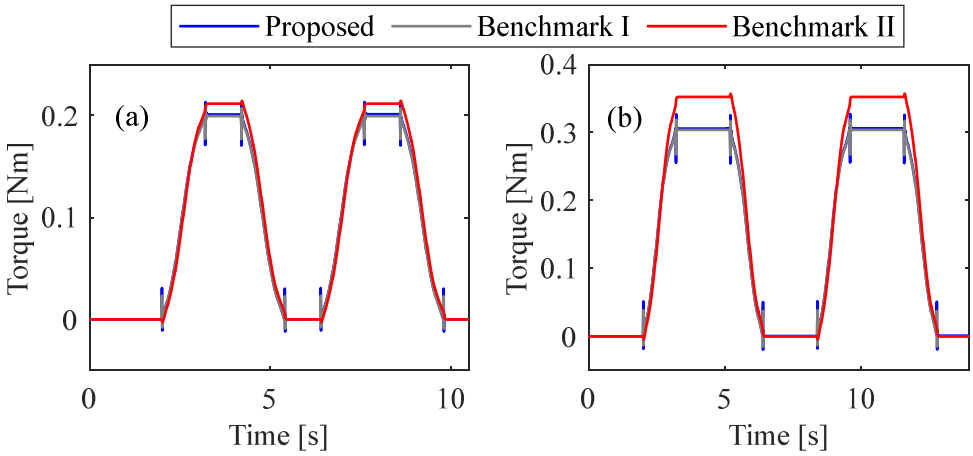


Figure 55. Comparison of the torques computed by proposed method and the two benchmarks, with different opening angles: 30° (a) and 50° (b).

6.5.4. Robustness considerations against model uncertainties

Since dynamic models are always affected by uncertainties respect to the real setups, the robustness of the results achieved by proposed method are assessed in the following by applying some perturbations to the dynamic model emulating the real system, to which feedforward torques computed through the nominal model are applied. The uncertain parameters of the dynamic models are the motor moment of inertia, the masses and lengths of the links (and hence the related moment of inertia), the stiffness of the rotational spring and the entries of the damping matrix. Firstly, the decoupled effect of the uncertainty of just one parameter at a time, by keeping the other ones to their nominal values, has been evaluated to assess which is the most sensitive parameter in terms of trajectory tracking error. The results of this analysis are reported in Figure 56, for both the opening angles considered (30° and 50°), where $\pm$10% variations of the nominal parameters are assumed. It can be noticed that the lengths of the two links are the most influential parameter; this is due mainly due to the error in the kinematics equation (134), that would also severely affect the inverse dynamic in a rigid link system. Among the two opening angles, smaller errors are obtained in the case of 30°, because of the smaller accelerations.

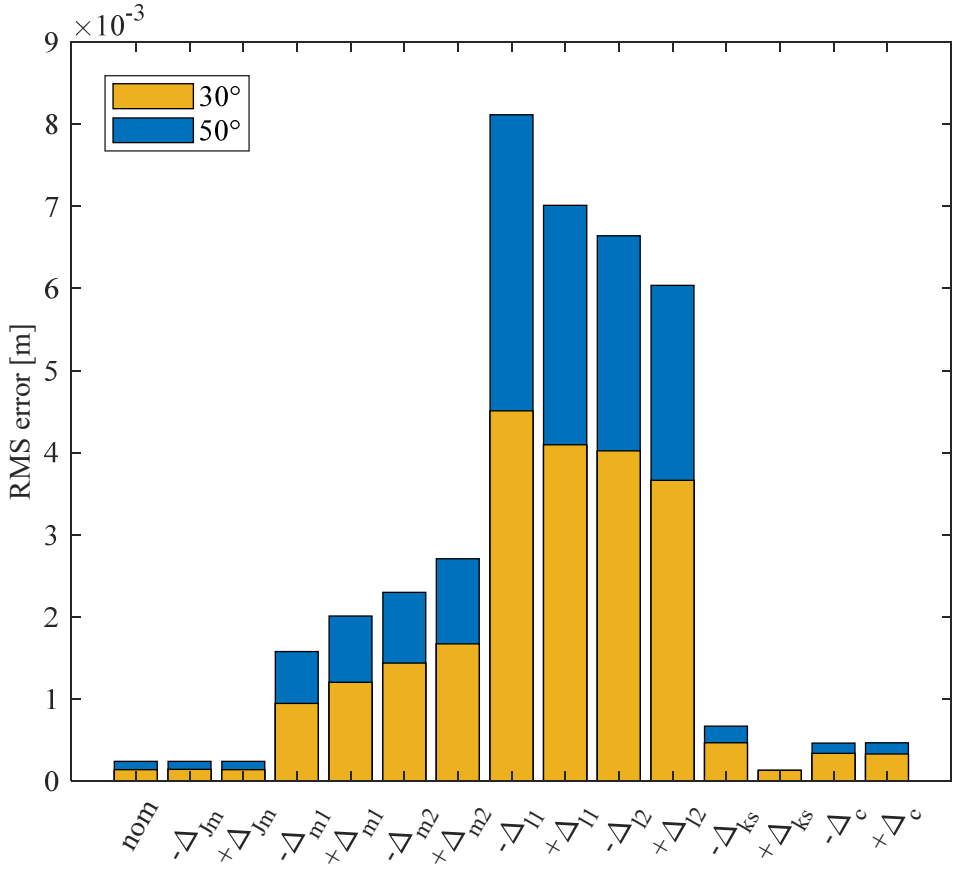


Figure 56. RMS trajectory tracking errors in the presence of model mismatch, when each +10% perturbations affect one parameter at a time.

A second robustness analysis is done with respect of the uncertainty on the motor torque loop dynamics. In this Chapter, as well as in all the papers on inverse dynamics of multibody and robotic systems quoted in the references, model inversion is done by assuming ideal response of the actuator current loop, i.e., infinite bandwidth of such a loop. This assumption is often reasonable since modern servomotors can reach up to 5000 Hz bandwidth [152]. On the other hand, "cheap" or old motors might have slower dynamic response. To evaluate this issue, robustness analysis is also performed by simulating the system response in the presence of a current loop modeled as a first order system

with a bandwidth ranging from 500 Hz to 5000 Hz. The results are summarized in Figure 57 that shows the RMS error as a function of the bandwidth (by assuming no uncertainty on the other parameters). It is evident that the tracking error approaches the ideal one as the bandwidth increases, and, for the case under investigation, bandwidth greater than 1500 Hz just causes a negligible increase. In contrast, the harmonic distortion and the lag in reproducing the commanded torque profile remarkable amplify the tracking error in the case of 500 Hz bandwidth.

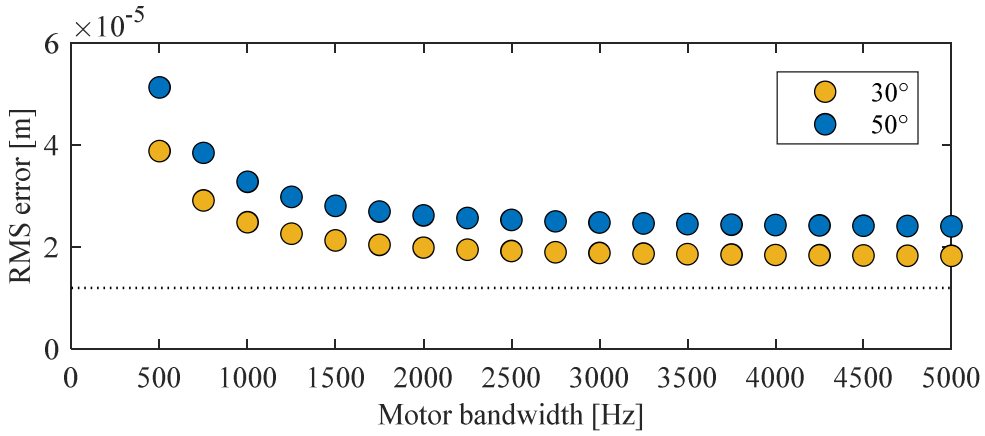


Figure 57. RMS trajectory tracking errors in the presence of non-ideal motor torque loop dynamics.

To further assess the robustness, a Monte Carlo analysis (with 2000 random tests) has been performed by considering the simultaneous perturbations of all the above-mentioned parameters. As for the current loop bandwidth, a uniform distribution in the interval [500 Hz, 5000 Hz] has been assumed. As for the other parameters, Gaussian distributions have been assumed, with a standard deviation equal to the 5 % of the nominal value. The average RMS value of the tracking error over a cycle is equal to 1.0 mm when the test case with the opening angle equal to 30° is considered, while the maximum RMS error achieved in such an analysis is 3.0 mm. In the case of a 50° opening angle, the average RMS value is 1.8 mm while the worst-case simulated leads to 5.6 mm. The Monte Carlo analysis is summarized through the trajectory tracking responses reported in Figure 58, to highlight the bands of uncertainty for both the opening angles evaluated, when the proposed method is adopted. The same perturbations have been applied to Benchmark I, by obtaining worse results because of the higher approximation it exploits in model inversion: the average and maximum RMS errors are, respectively, 1.6 mm (+60% compared with the proposed method) and 3.8 mm (+27%) with the 30° opening angle, and 2.5 mm (+38%) and 5.9 mm (+6%).

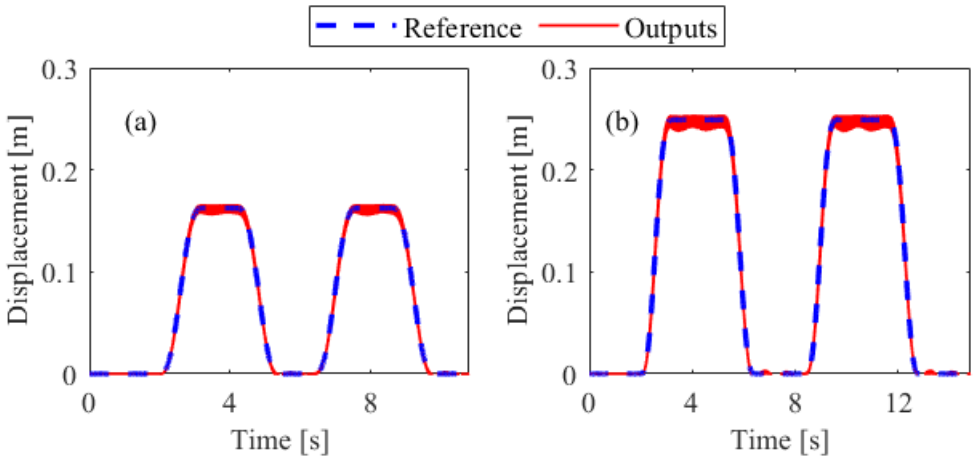


Figure 58. Trajectory tracking responses in the presence of random model mismatches when perturbations affect all parameters simultaneously (proposed method): 30° (a) and 50° (b).

6.5.5. Experimental application

The motor torques shown in Figure 52, and computed through the method proposed in this Chapter, by exploiting the nominal model parameters, have been applied to the experimental setup. As far as friction is concerned, the viscous term is considered in Eq.(131), and hence it affects also the internal dynamics. Additionally, a Coulomb term related to the friction at the active joint (and hence including the contribution of the motor rotor) and equal to ± 0.01 Nm has been included in the algebraic equation (116) by adding the term $0.01\,\mathrm{sign}\left(\dot{\theta}_1\right)$ Nm. Such a value has been estimated through experimental identification. More complicated friction model can be employed in the algebraic part, by taking advantage of the wide literature on friction modelling and compensation; this goes, however, beyond the scope of this work. No feedback terms are adopted, and therefore the showed results just rely on open loop control through model inversion, thus providing a fair and severe evaluation of its effectiveness.

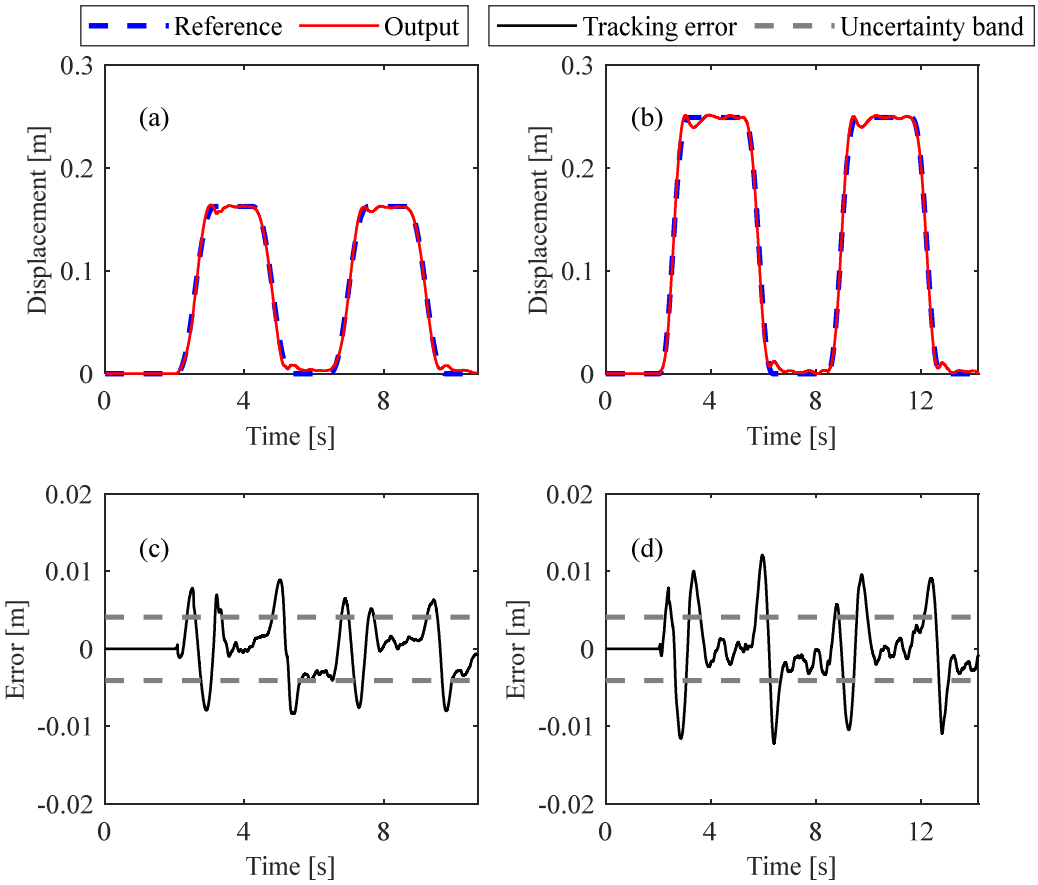


Figure 59. Experimental trajectory tracking responses with their relative tracking errors with different opening angles: 30° (a, c) and 50° (b, d).

The achieved displacements are reported in Figure 59, for each opening angle, and have been measured through a high-resolution Vicon motion capture exploiting 4 cameras Vicon Vantage V5. A precise tracking of the references is obtained, although the tracking error is greater than the one expected by the numerical analysis: the RMS values of the error are equal to 3.5 mm with the 30° opening angle, and 4.3 mm with the 50° one. This aspect is mainly due the unavoidable presence of model mismatches between the nominal one, used to compute the torque, and the real setup. In particular, among the parameters investigated through the Monte Carlo analysis, it should be noted that the current loop bandwidth of the actuator is about 500 Hz, thus approaching the worst-case scenario for such a parameter. A second relevant uncertainty source is friction, that has been remarkably simplified in the model and whose compensation has been performed in a very simple algebraic way. Finally, the deformation of the wire of the encoder placed at the passive joint significantly perturbs the system dynamics both in terms of mass distribution, by creating an equivalent stiffness (that slightly changes as the wire moves, since the diameter is 5 mm) and by acting as an external disturbance force acting on the system itself. Nonetheless, by considering that no feedback is adopted, tracking is accurate if compared with the resolution sported by the two encoders mounted on the mechanism, that is equal to 4.1 mm.

6.6. Conclusions

This Chapter proposes an enhanced inverse dynamics techniques for non-differentially flat, nonminimum-phase underactuated multibody systems, to obtain a feedforward control term without pre-actuation. The method is suitable for mechanisms and robots with passive joints or flexible links

and allow boosting the design of closed-loop control schemes with high robustness, thanks to the possibility of lowering the feedback gains.

By taking advantage of the idea already proposed in the literature, of properly approximating the desired output as a linearly combined output, and by performing the output redefinition to stabilize the internal dynamics, stable inversion of such a dynamic is obtained. Then, by exploiting the exact model of the actuated sub-system, the computation of the feedforward forces is obtained with just minor approximation. This possibility is allowed by a general approach that exploits the *QR-decomposition* of the input matrix to partition the independent coordinates into actuated and unactuated ones, thus simplifying the definition of the internal dynamics and hence its stabilization.

The concurrent use of these tools, results into a comprehensive, and computationally simple, method for precise and accurate inverse dynamics, that can be applied to several underactuated systems where linear (or piecewise-linear) output approximations are reasonable, by improving the existing state-of-the-art.

The effectiveness of the proposed method has been assessed numerically with three different test cases, highlighting different features of the method also by means of comparison with several state-of-the art techniques. In all the studied cases, the proposed method has showed higher accuracy in tracking the desired trajectory with tracking errors that are smaller than all the benchmarks. Furthermore, the proposed technique has been experimentally applied on a 2-DOF underactuated robotic arm, made by two rigid links and a passive joint. The experimental results highlight the enhanced performances achieved through the proposed method even in the presence of plant uncertainties and model mismatches. As a result, the mean experimental tracking errors lay inside the uncertainty band defined by the resolution of the encoders mounted on the mechanism.

The absence of pre-actuation and the simple proposed formulation, which exploits some algebraic calculations and the numerical integration of the equation of motions of the stabilized internal dynamics without the need of optimization procedures, make the method suitable for applications where no pre-calculation is allowed and, therefore, where a real-time calculation of feedforward is required.

7. Model Inversion for Precise Path and Trajectory Tracking in a Underactuated, Nonminimum-phase, spatial overhead crane

The idea, that has been proposed in the previous Chapter, is here considered with a new interpretation, in particular with the goal to offer a motion planning technique for underactuated systems. Usually, industrial robots are characterized by co-located controllers and, since the load is not measured, the only way to achieve good performances in terms of path and trajectory tracking of the load is by implementing a proper motion planning, above all when low inertia ratios are present. Therefore, the approximation of the linearly combined output technique, even though already very effective, has been improved by exploiting the formulation of the separable output. This formulation could be directly exploited to further improve the feedforward computation presented in the previous Chapter, even though it is already highly effective, however, focusing the attention on industrial robots, this Chapter proposes a motion planning technique for precise path and trajectory tracking in an underactuated, nonminimum-phase, spatial overhead crane. Besides having a number of independent actuators that is smaller than the number of degrees of freedom, tip control on this system has to an unstable internal dynamics that leads to divergent solution of the inverse dynamic problem. As previously mentioned, this Chapter exploits the representation of the controlled output as a separable function of the actuated (i.e., the platform translations) and unactuated (i.e., the swing angles) coordinates to easily formulate the internal dynamics, without any approximation, and to study its stability. Then, output redefinition is adopted within the internal dynamics to stabilize it, leading to stable and causal reference commands for the platform translations. Besides proposing the theoretical formulation of this novel method, this Chapter includes the numerical validation and the experimental application on a laboratory setup. Comparison with the state-of-the-art input shaping is also proposed. The results clearly show that almost exact tracking is obtained also in the experiments.

7.1. Introduction

7.1.1. Motivations and state-of-the-art

Underactuated multibody and robotic systems are those where the number of independent control forces is smaller than the number of degrees of freedom (DOFs). Underactuation is caused by the presence of flexible links, passive joints, as well as in case of actuator failure. Accurate and precise trajectory tracking in underactuated systems relies on the presence of effective control schemes, with feedback and feedforward terms, and an optimized motion planning accounting for the presence of some unactuated coordinates and to the related vibrational dynamics. Both feedforward control and motion planning usually exploit the inversion of the dynamic model, that is not straightforward as in fully actuated systems because of the rectangular force distribution matrix. The algebraic solution of the inverse dynamic problem is possible if the system is differentially flat [139]. Unfortunately, flatness is a system property that is not straightforward to assess and finding a flat output definition is not trivial and impose a lot of algebraic manipulations [104] since no systematic approach exists. Additionally, the use of flatness in underactuated systems imposes very smooth reference trajectories, usually with four continuous derivatives, that are often not desirable because of high values of the maximum and average speed and accelerations [153].
The difficulties in model inversion are exacerbated in the case of nonminimum-phase systems that have also unstable internal dynamics. In the case of linear systems, this is related to the presence of zeros with positive real part: if the model is inverted these zeros become right half-plane poles, making diverging the control actions. Generally speaking, also with reference to nonlinear systems, inversion of the dynamic model, either for computing the feedforward forces or the commanded motion of the actuated coordinates, is usually solved through approximate or non-causal solutions.

The literature proposes several approaches to model inversion in underactuated systems, and some papers also address the more challenging nonminimum-phase occurrence. More attention is, however, paid to inverse dynamics for computing the control forces, while motion planning is less investigated. In [41], an inversion-based approach to nonlinear output tracking control is proposed, together with a feedback term to stabilize the system along the desired trajectory; the solution is achieved by applying the Byrnes-Isidori regulator to a specific trajectory, leading to a noncausal solution that requires pre-actuation.
Another interesting approach is the method of servo constraints, that been proposed in [154] and in the related papers. It transforms the system model into a set of DAE where the reference to track is properly represented through algebraic constraints. The method, in its original formulation, does not formulate the internal dynamics and therefore is not suitable in those case of nonminimum-phase systems where it should be properly stabilized to obtain non diverging results. Such a method has been also applied to various systems, such as a crane with hoisting in [155], and in [140] for motion planning in a two-disk system.
Two further feedforward control techniques are proposed in [145]: in the first one, the model inversion is performed through the concept of nonlinear input-output normal form, which is achieved thanks to a coordinate transformation, while the second one exploits servo-constraints, leading to solve a set of differential algebraic equations (DAEs). Both techniques are exploited in an optimization problem. An interesting comparison is presented in [4], considering two different strategies that are present in the literature: the standard stable inversion method, which is solved through the formulation of a two-point boundary value problem, and an alternative optimization problem formulation, which is defined without any boundary conditions. This comparison shows that both approaches can be formulated through DAEs and, additionally, numerical results are presented considering manipulators with passive and compliant joints.
Optimization based formulations exploiting two-point boundary value problems have been also adopted, either for inverse dynamics [156–159] or motion planning [96]. In the usual formulation of two-point boundary value problems, no specifications are set on the path connecting the boundary points, while a function cost is minimized to accomplish secondary tasks, such as increasing robustness.
The techniques discussed so far, are devoted to general underactuated systems. In the specific case of cranes, the most famous motion planning technique is the Input Shaping, that has been mainly developed for two-DOF cranes (i.e., with just one vibrational mode) [160]. Basically, the reference trajectory is convolved with a baseline of impulses, computed on the basis of a simplified system model, to cancellate the residual vibration of the unactuated load. Input shaping in spatial crane has been mainly applied to rest-to-rest linear motions [161], although some papers [162,163] also apply it to path and trajectory tracking. In motion planning and control of spatial cranes, a lot of attention has been paid to feedback control (see e.g., as some relevant examples, the schemes proposed in [164–171]), also including techniques that perform closed-loop modifications of the reference trolley trajectory to ensure control of the load swing [151] or accurate path tracking [172].

7.1.2. Contributions of the Chapter

In this Chapter, a stable model inversion technique suitable for nonminimum-phase underactuated multibody systems is exploited to motion planning in a 4-DOF spatial overhead crane. The goal is to compute the reference trajectories for the actuated coordinates ensuring that the controlled output tracks the desired reference, in term of both trajectory in time and spatial path. No feedback control on the load side is exploited, and hence path and trajectory control, as well as load swing reduction, are just executed through an optimized motion planning of the trolley.
Firstly, the crane dynamic model is partitioned into two parts: an actuated subsystem and an unactuated one, that is useful to obtain the internal dynamics and to disregard the equations related to

the actuated subsystem that would be of interest just for the computation of the control forces. Secondly, the desired output trajectory of the system is described as a nonlinear separable function of the actuated and unactuated coordinates, to allow for a simple formulation of the internal dynamics. Output redefinition is then exploited to stabilize and solve it, to compute the time history of the internal dynamics in the trajectory execution. Finally, the commanded trajectory for the actuated coordinates is computed through kinematic nonlinear inversion of the output equation.
The proposed method relies on the numerical integration of the obtained ordinary differential equations (ODE), without requiring optimization as many methods in the literature do, and with a reduced computational effort. The obtained solution is causal since it does not require pre-actuation. This Chapter proposes the theory, that is itself a new contribution in the literature, the numerical validation and, finally, the experimental application through a laboratory setup, in the execution of several trajectories. Comparison with the well-known input shaping technique [163], as well with the unshaped command is provided, to demonstrate the effectiveness.

7.2. System model and method description

7.2.1. System dynamic model

Let us consider the dynamic model of the 4-DOF overhead cartesian crane under investigation sketched in Figure 45. As mentioned in the previous Chapter, the independent coordinates include two absolute translations describing the planar motion of the platform (also denoted as the trolley) of the crane, that are the two actuated coordinates $\mathbf{q}_\mathbf{A} = \begin{bmatrix} x_P & y_P \end{bmatrix}^T \in \mathbb{R}^2$, and two unactuated components of the swing angle $\mathbf{q}_\mathbf{U} = \begin{bmatrix} \theta_X & \theta_Y \end{bmatrix}^T \in \mathbb{R}^2$. θ_X is the swing angle projected on the XZ plane, while θ_Y is the swing angle measured from the XZ plane. Indeed, by assuming that the cable is taut with tensile stress and that hoisting is not allowed (i.e., the cable length h is constant), the four nonlinear ordinary differential equations (ODEs) describing the system dynamic behaviour are the ones reported in Eq.(126). All the system parameters are the same of the previous Chapter and therefore their values are the ones reported in Table 8.
Eq.(126) can be written in a concise form as the typical model of a nonlinear multibody system, where $\mathbf{M}(\mathbf{q}_\mathbf{U}) \in \mathbb{R}^{4\times4}$ is the position-dependent mass matrix, $\mathbf{C}(\mathbf{q}_\mathbf{U}, \dot{\mathbf{q}}_\mathbf{U})$ is the vector of the speed-dependent contributions (both inertial terms and viscous friction terms), $\mathbf{F}(\mathbf{q}_\mathbf{U}) \in \mathbb{R}^4$ is the vector of forces due to gravity acceleration, $\mathbf{B} \in \mathbb{R}^{4\times2}$ is the control forces distribution matrix and $\mathbf{u} = \begin{bmatrix} u_x & u_y \end{bmatrix}^T \in \mathbb{R}^2$ collects the control forces.

$$\mathbf{M}(\mathbf{q}_\mathbf{U}) \begin{bmatrix} \mathbf{q}_\mathbf{A} \\ \mathbf{q}_\mathbf{U} \end{bmatrix} + \mathbf{C}(\mathbf{q}_\mathbf{U}, \dot{\mathbf{q}}_\mathbf{U}) = \mathbf{F}(\mathbf{q}_\mathbf{U}) + \mathbf{B}\mathbf{u} \tag{141}$$

The goal of motion planning is to precisely make the suspended load track a desired path (both in in space and in time). Since just two control forces are applied, the motion of just two output coordinates is imposed. The output vector $\mathbf{y} \in \mathbb{R}^2$ is therefore defined as the following nonlinear algebraic function of the four independent coordinates, by exploiting trigonometric kinematic relations:

$$\mathbf{y} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} x_P + h \sin\theta_X \cos\theta_Y \\ y_P + h \sin\theta_Y \end{bmatrix} \tag{142}$$

7.2.2. Formulation and stabilization of the internal dynamics

The dynamic model in Eq.(126) and Eq. (141) can be partitioned into actuated and unactuated coordinates:

$$\begin{bmatrix} \mathbf{M}_{\mathbf{AA}} & \mathbf{M}_{\mathbf{AU}} \\ \mathbf{M}_{\mathbf{AU}}^T & \mathbf{M}_{\mathbf{UU}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\mathbf{A}} \\ \ddot{\mathbf{q}}_{\mathbf{U}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mathbf{A}} \\ \mathbf{C}_{\mathbf{U}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\mathbf{A}} \\ \mathbf{F}_{\mathbf{U}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{\mathbf{A}} \\ \mathbf{0} \end{bmatrix} \mathbf{u} \tag{143}$$

By considering just the unactuated subsystem, the following set of two nonlinear ODEs is obtained:

$$\mathbf{M}_{\mathbf{AU}}^T \ddot{\mathbf{q}}_{\mathbf{A}} + \mathbf{M}_{\mathbf{UU}} \ddot{\mathbf{q}}_{\mathbf{U}} + \mathbf{C}_{\mathbf{U}} = \mathbf{F}_{\mathbf{U}} \tag{144}$$

The trajectory design exploits the inversion of the equations of motion in Eq.(144): the input of the inverted equations will be the desired trajectory in time of the controlled output ($\mathbf{y}^{\mathbf{ref}}, \dot{\mathbf{y}}^{\mathbf{ref}}, \ddot{\mathbf{y}}^{\mathbf{ref}}$), while the final output to be computed are the commanded values of the actuated variables ensuring that $\mathbf{y}$ tracks such a reference.

In this work, it is exploited the formulation of the desired output, and hence of the reference trajectory, as a nonlinear, separable function of $\mathbf{q}_{\mathbf{A}}$ and $\mathbf{q}_{\mathbf{U}}$. Indeed, it can be easily recognized that the following form of Eq. (142) can be formulated, with $\mathbf{g}: \mathbb{R}^2 \mapsto \mathbb{R}^2$, $\mathbf{g}(\mathbf{q}_{\mathbf{U}}) = h \begin{bmatrix} \sin\theta_X \cos\theta_Y \\ \sin\theta_Y \end{bmatrix}$, as a nonlinear function:

$$\mathbf{y} = \mathbf{q}_{\mathbf{A}} + \mathbf{g}(\mathbf{q}_{\mathbf{U}}) \tag{145}$$

The second derivative of the output is, in turn:

$$\begin{aligned} \ddot{\mathbf{y}} = \begin{bmatrix} \ddot{x}_P \\ \ddot{y}_P \end{bmatrix} + h \begin{bmatrix} \cos\theta_X \cos\theta_Y & -\sin\theta_X \sin\theta_Y \\ 0 & \cos\theta_Y \end{bmatrix} \begin{bmatrix} \ddot{\theta}_X \\ \ddot{\theta}_Y \end{bmatrix} + \\ h \begin{bmatrix} -\sin\theta_X \cos\theta_Y \dot{\theta}_X{}^2 - 2\cos\theta_X \sin\theta_Y \dot{\theta}_X \dot{\theta}_Y - \sin\theta_X \cos\theta_Y \dot{\theta}_Y{}^2 \\ -\sin\theta_Y \dot{\theta}_Y{}^2 \end{bmatrix} \end{aligned} \tag{146}$$

Therefore, the desired output acceleration can be written in the following concise form,

$$\ddot{\mathbf{y}}^{\mathbf{ref}} = \ddot{\mathbf{q}}_{\mathbf{A}} + \mathbf{J}_{\mathbf{g}}(\mathbf{q}_{\mathbf{U}}) \ddot{\mathbf{q}}_{\mathbf{U}} + \boldsymbol{\gamma}_{\mathbf{g}}(\mathbf{q}_{\mathbf{U}}, \dot{\mathbf{q}}_{\mathbf{U}}) \tag{147}$$

where $\mathbf{J}_{\mathbf{g}} \in \mathbb{R}^{2\times 2}$ is the Jacobian matrix of $\mathbf{g}$, and $\boldsymbol{\gamma}_{\mathbf{g}} \in \mathbb{R}^2$ contains the centripetal terms. In writing Eq. (147), it is assumed that the trajectory $\mathbf{y}^{\mathbf{ref}}(t)$ is, at least, twice differentiable. This is an obvious requirement for any mechanical system, also in the case of fully actuated ones, where finite accelerations (and hence finite control forces) are required [173].

The formulation of the controlled output as a separable function is a more general formulation of the linearly combined output proposed in [174], as well as in [2,145], for the solution of the inverse dynamic problem, and allows effectively handling larger elastic displacements than a linear representation.

Eq. (147) enables writing the internal dynamics as just a function of $\mathbf{q}_{\mathbf{U}}$ (and its derivatives) and $\ddot{\mathbf{y}}^{\mathbf{ref}}$, by removing the dependence of Eq. (144) on $\ddot{\mathbf{q}}_{\mathbf{A}}$, leading to the following nonlinear ODEs:

$$\left[\mathbf{M}_{\mathbf{UU}} - \mathbf{M}_{\mathbf{AU}}^T \mathbf{J}_{\mathbf{g}}\right] \ddot{\mathbf{q}}_{\mathbf{U}} + \left[\mathbf{C}_{\mathbf{U}} - \mathbf{M}_{\mathbf{AU}}^T \boldsymbol{\gamma}_{\mathbf{g}}\right] = \mathbf{F}_{\mathbf{U}} - \mathbf{M}_{\mathbf{AU}}^T \ddot{\mathbf{y}}^{\mathbf{ref}} \tag{148}$$

The stability of the internal dynamics can be effectively studied through the associated zero-dynamics [2], i.e., the internal dynamics with the output constrained to be zero for all the time, that is computed by setting $\mathbf{y}^{\mathbf{ref}} = \dot{\mathbf{y}}^{\mathbf{ref}} = \ddot{\mathbf{y}}^{\mathbf{ref}} = \mathbf{0}$ in Eq. (148). In this way, the effect of the input reference trajectory is removed. Then, the local asymptotic stability of the equilibrium point $\mathbf{q}_{\mathbf{U}} = \dot{\mathbf{q}}_{\mathbf{U}} = \mathbf{0}$ of the nonlinear system is assessed: if some eigenvalues have positive real parts, the internal dynamics is locally unstable [145] and therefore divergent integration of Eq. (148) is obtained, leading to unbounded solutions and hence the impossibility to compute a feasible, exact and causal system inversion. These systems are said to by nonminimum-phase.
This evaluation shows that the internal dynamics is unstable when $\mathbf{y}$ is defined as the load position, thus leading to divergent integration of the ODEs in Eq. (148). In this work, it is therefore exploited the concept of "output redefinition" within the internal dynamics, i.e., it is assumed a fictitious output $\tilde{\mathbf{y}}$ to be controlled, that is placed along the cable at a distance $\tilde{h}$ from the attachment point of the cable:

$$\tilde{\mathbf{y}} = \begin{bmatrix} x_P \\ y_P \end{bmatrix} + \tilde{h} \begin{bmatrix} \sin\theta_X \cos\theta_Y \\ \sin\theta_Y \end{bmatrix} \tag{149}$$

$\tilde{h}$ should be chosen to ensure stability of the internal dynamics and accurate model inversion.
By considering Eqs. (146) and (147), the desired output acceleration of the fictitious output can be written as follows:

$$\ddot{\tilde{\mathbf{y}}}^{\mathbf{ref}} = \ddot{\mathbf{q}}_{\mathbf{A}} + \beta \mathbf{J}_{\mathbf{g}}(\mathbf{q}_{\mathbf{U}})\, \ddot{\mathbf{q}}_{\mathbf{U}} + \beta \boldsymbol{\gamma}_{\mathbf{g}}(\mathbf{q}_{\mathbf{U}}, \dot{\mathbf{q}}_{\mathbf{U}}) \tag{150}$$

where $\beta = \dfrac{\tilde{h}}{h}$.
The stabilized internal dynamics, to be solved for computing the trajectory of the unactuated coordinates in tracking the desired reference, is therefore:

$$\left[\mathbf{M}_{\mathbf{UU}} - \beta \mathbf{M}_{\mathbf{AU}}^T \mathbf{J}_{\mathbf{g}}\right] \ddot{\mathbf{q}}_{\mathbf{U}} + \left[\mathbf{C}_{\mathbf{U}} - \beta \mathbf{M}_{\mathbf{AU}}^T \boldsymbol{\gamma}_{\mathbf{g}}\right] = \mathbf{F}_{\mathbf{U}} - \mathbf{M}_{\mathbf{AU}}^T \ddot{\tilde{\mathbf{y}}}^{\mathbf{ref}} \tag{151}$$

The eigenvalue analysis of the zero dynamics associated to Eq. (151) reveals that stability is achieved if and only if $0 < \beta < 1$. The parameter β should be chosen as close as possible to the unitary value to ensure a good reconstruction of the desired output trajectory, thanks to a redefined output close to the actual one, while assuring stable integration. Indeed, it should be noted that the actual stability of Eq. (151) is affected by the numerical integration schemes adopted in the solution, that might cause numerical instability even if the theoretical eigenvalues have negative, though small, real parts. For example, conditionally stable and low order algorithms such as Euler's scheme might be unstable for values of β approaching 1, for large sample times and in the presence of roundoff errors. A relevant computational feature of the proposed approach is that it does not impose a specific integration scheme, and therefore the ones leading to more accurate results can be chosen among those usually adopted in simulating multibody systems [100].
As far as $\ddot{\tilde{\mathbf{y}}}^{\mathbf{des}}$ is concerned, since β approaches 1, it can be reasonably set $\ddot{\tilde{\mathbf{y}}}^{\mathbf{ref}} \equiv \ddot{\mathbf{y}}^{\mathbf{ref}}$, i.e., no reference correction is necessary to compensate for the use of the redefined output in solving the internal dynamics.

7.2.3. Computation of the command values of the actuated coordinates

Once the internal dynamics is stable, $\mathbf{q}_{\mathbf{U}}$, $\dot{\mathbf{q}}_{\mathbf{U}}$ and $\ddot{\mathbf{q}}_{\mathbf{U}}$ are computed by integrating the differential equations in Eq. (151) over the whole cycle time (and will be henceforth denoted as $\mathbf{q}_{\mathbf{U}}^*$, $\dot{\mathbf{q}}_{\mathbf{U}}^*$, $\ddot{\mathbf{q}}_{\mathbf{U}}^*$).

Then, the reference trajectory for the actuated coordinates ($\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}$, $\dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}}$, $\ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}}$) is obtained by exploiting the following equations, inferred from Eq.(147):

$$\begin{aligned}\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}} &= \mathbf{y}^{\mathrm{ref}} - \mathbf{g}\left(\mathbf{q}_{\mathrm{U}}^{*}\right)\\ \dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}} &= \dot{\mathbf{y}}^{\mathrm{ref}} - \mathbf{J}_{\mathrm{g}}\left(\mathbf{q}_{\mathrm{U}}^{*}\right)\dot{\mathbf{q}}_{\mathrm{U}}\\ \ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}} &= \ddot{\mathbf{y}}^{\mathrm{ref}} - \mathbf{J}_{\mathrm{g}}\left(\mathbf{q}_{\mathrm{U}}^{*}\right)\ddot{\mathbf{q}}_{\mathrm{U}}^{*} - \boldsymbol{\gamma}_{\mathrm{g}}\left(\mathbf{q}_{\mathrm{U}}^{*},\dot{\mathbf{q}}_{\mathrm{U}}^{*}\right)\end{aligned} \tag{152}$$

The computation of the forces necessary to execute such a motion for the trolley relies on the controller of the trolley, that can be based on any feedback or feedforward scheme ensuring precise tracking of $\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}$.

7.2.4. Some comments on the required command forces

Although it is not among the goals of the proposed work, that is focused on trajectory planning, the developments proposed up to now can be also exploited to compute the command forces for the actuators driving the actuated coordinates, that can be adopted for feedforward control. By exploiting $\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}$, $\dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}}$, $\ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}}$ computed in Eq. (152) and $\mathbf{q}_{\mathrm{U}}^{*}$, $\dot{\mathbf{q}}_{\mathrm{U}}^{*}$, $\ddot{\mathbf{q}}_{\mathrm{U}}^{*}$ computed in Eq. (151), such forces are obtained through the upper part of Eq. (143) that becomes an algebraic equation:

$$\mathbf{u} = \mathbf{B}_{\mathrm{A}}^{-1}\left(\mathbf{M}_{\mathrm{AA}}\ddot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}} + \mathbf{M}_{\mathrm{AU}}\ddot{\mathbf{q}}_{\mathrm{U}}^{*} + \mathbf{C}_{\mathrm{A}}\left(\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}},\mathbf{q}_{\mathrm{U}}^{*},\dot{\mathbf{q}}_{\mathrm{A}}^{\mathrm{ref}},\dot{\mathbf{q}}_{\mathrm{U}}^{*}\right) - \mathbf{F}_{\mathrm{A}}\left(\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}},\mathbf{q}_{\mathrm{U}}^{*}\right)\right) \tag{153}$$

7.3. Experimental application

7.3.1. Setup description

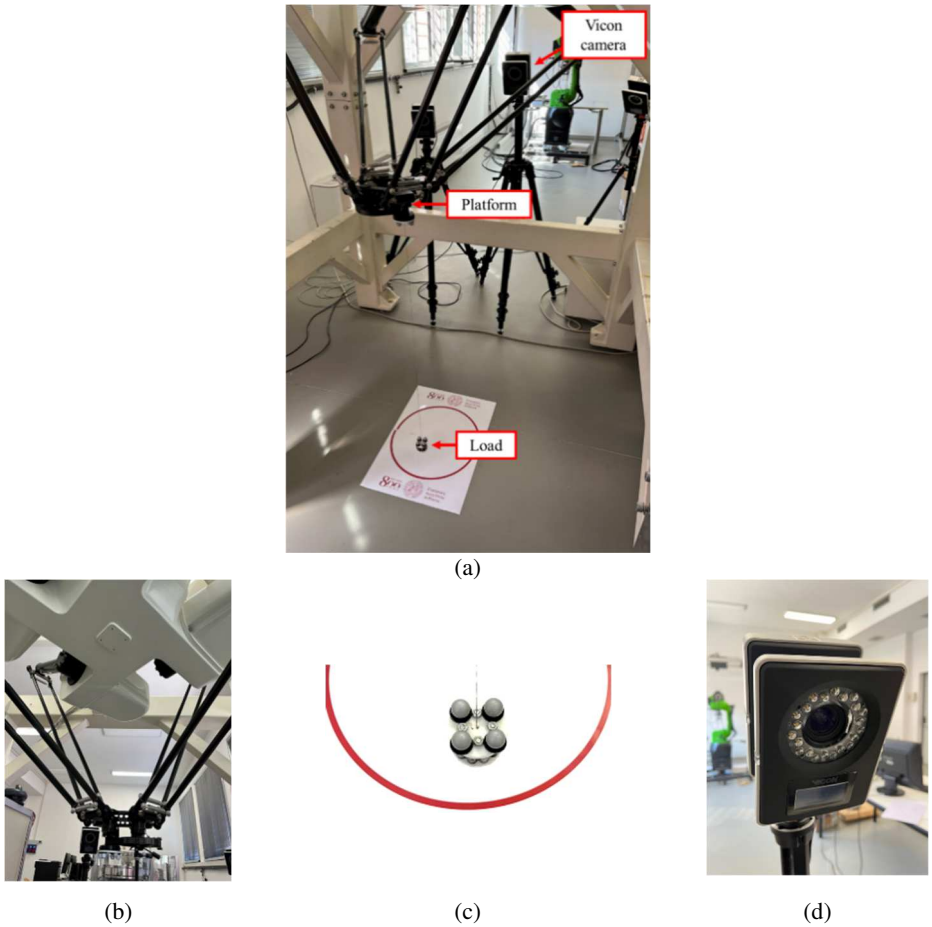
The overhead crane is mimicked through the end-effector of an Adept Quattro s650 h parallel robot, whose planar Cartesian translations represents the motion of the actuated platform $\mathbf{q}_{\mathrm{A}} = \begin{bmatrix} x_P & y_P \end{bmatrix}^T$. The values of the system parameters are summarized in Table 10.

Table 10. System parameters.

Parameter	Value
m : load mass [kg]	0.2011
h : pendulum length [m]	0.9624
c_θ : pendulum friction coefficient [Nms]	0.001

A picture of the manipulator is shown in Figure 60. The robot is controlled by the proprietary Adept SmartController. The commanded trajectories, in term of position $\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}$, are computed offline through a Matlab code, and loaded through an array of points in time in the Adept SmartController that computes the command torques for the axis motor through its native controller. Since such a control scheme is purely collocated and the inertia ratio approaches zero, the load oscillations are not visible to the platform side; therefore, no closed-loop control is made on the load motion. It should be noted that such a proprietary controller cannot be modified, and therefore motion and vibration control in such a system must just rely on enhanced motion planning, as the method proposed in this Chapter. The solution of the ODEs of the internal dynamics in Eq. (151) has been performed through the 4th order Runge-Kutta integrator, with a sampling time equal to 1 ms, and by setting $\beta = 0.9999$ since

it provides stable and accurate solutions. Then, decimation is performed to generate $\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}$ with the larger sample time of $16\ \mathrm{ms}$, as required by the robot controller. As a result, the reference trajectory that is actually commanded to the robot is slightly perturbed, compared to the theoretical one, thus leading to slightly greater error than those expected.
The motion of the pendulum is measured through a Vicon motion capture system exploiting six high-speed cameras to reconstruct the position of some markers located on the robot end-effector and on the load. The nominal accuracy of the vision system, declared by the manufacturer, is $1.5\ \mathrm{mm}$. In practice, the uncertainty band of the measures is higher, due to small errors that can occur in the positioning of the calibration wand and of the marker, as well as due to the environmental drift, i.e. the uncertainty due to the change of environmental factors after calibration such as temperature and unwanted light sources.
Three sample paths are shown in four tests, although the method can be applied to any path: a planar circle, with two different motion times, a planar rhombus and a planar spiral. Comparison is done with the input shaping Zero-Vibration (henceforth denoted as ZV) approach [160], and by setting $\mathbf{q}_{\mathrm{A}}^{\mathrm{ref}}(t) = \mathbf{y}^{\mathrm{ref}}(t)$, i.e. by treating the system as if no vibrational dynamics affects its motion without any shaping to compensate for vibrations (henceforth denoted as “unshaped”).



(a) (b) (c) (d)

Figure 60. The experimental test-bed: overview (a), the platform (b), the load with markers (c), a Vicon camera (d).

In the analyses of the experimental results, the following variables are introduced and displayed: $\left(x_P^{\mathrm{ref}}, y_P^{\mathrm{ref}}\right)$ are the Cartesian reference values for the platform, while $\left(x_P^{\mathrm{exp}}, y_P^{\mathrm{exp}}\right)$ are the experimentally measured ones. The Cartesian reference values for the suspended load $\mathbf{y}^{\mathrm{ref}}$ are $\left(x_L^{\mathrm{ref}}, y_L^{\mathrm{ref}}\right)$, while $\left(x_L^{\mathrm{exp}}, y_L^{\mathrm{exp}}\right)$ are the experimentally measured ones. Performances are assessed by

showing details on the tracking error on the load side $\left(e_{t,L}^{x}, e_{t,L}^{y}\right)$ and on the platform $\left(e_{t,P}^{x}, e_{t,P}^{y}\right)$ in both the *X* and *Y* directions. The contour error during motion, i.e., the actual difference in distance between the reference path and the actual path, is also evaluated through its root mean square (RMS) value, both for the load ($e_{c,L}$) and for the platform ($e_{c,P}$).

7.3.2. Test case 1: planar circular path with 8 s motion time

7.3.2.1. Numerical results

In the first test, the suspended mass should track a 0.2 m radius circumference in 8 s, by assuming a 7th-degree polynomial motion law. The motion starts at 2 s. Numerical results are first briefly proposed, just for this test case, to prove how the proposed method is effective in inverting the model of a nonminimum-phase system for accurate trajectory and path tracking.
Figure 61 shows the simulated and the reference load displacements in the Cartesian plane, as well as the commanded motion of the platform. The motion of the load is almost perfectly overlapped with the reference, with just some negligible errors. It is interesting to notice how the proposed method shapes the motion of the platform, i.e., of the actuated coordinates: centrifugal effects are compensated by a suitably smaller radius. Additionally, the load oscillations are weakly excited, leading to negligible residual vibrations as corroborated by the tracking errors shown in Figure 62: the maximum value is 0.008 mm, at the end of the motion, and the RMS values are 0.004 mm along both the *X* and *Y* directions.
These results confirm the correctness of the proposed approach, that is able of inverting almost exactly the dynamic model despite its nonminimum-phase nature.

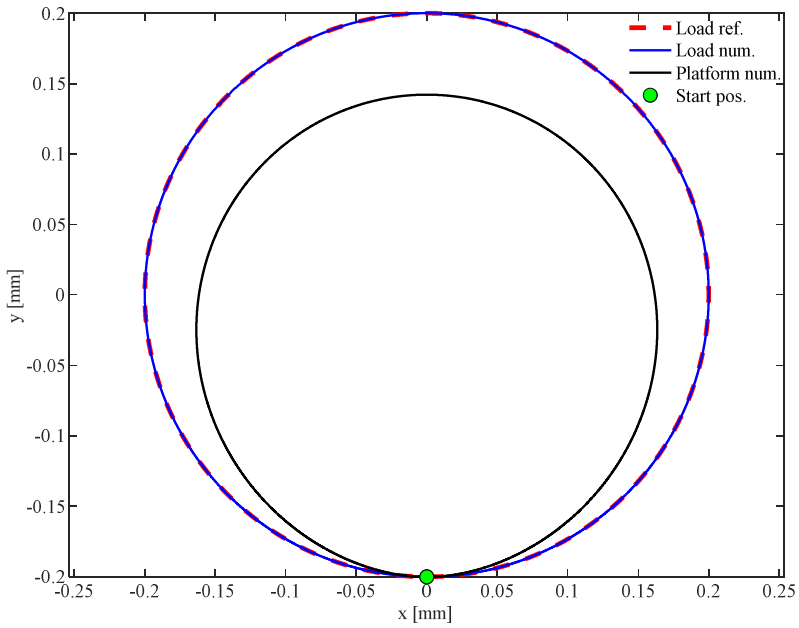


Figure 61. Test case 1, numerical results. Platform and load displacements in the Cartesian plane with the proposed method.

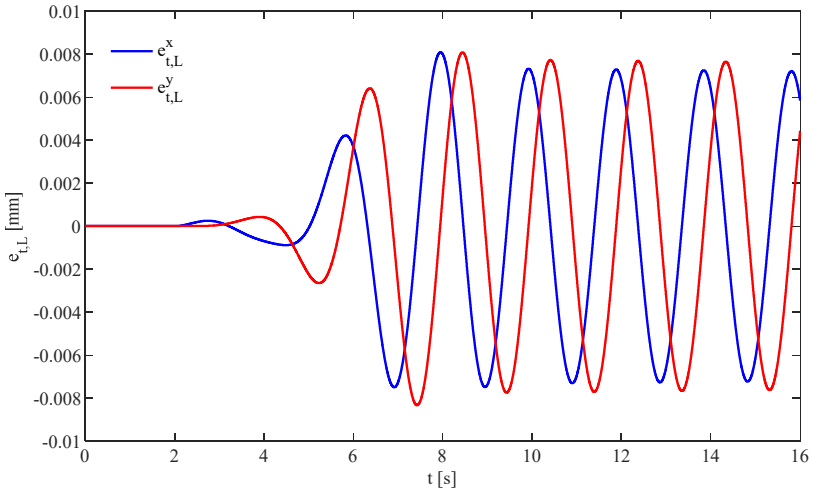


Figure 62. Test case 1, numerical results. Time history of the tracking errors.

7.3.2.2. Experimental results

The platform reference motion computed in Section 7.3.2.1 is here fed to the robot controller to evaluate the experimental performances. Figure 63.a and Figure 63.b show the time histories of the references and of the measured motions of both the platform and the load: the effectiveness of the proposed method is evident since the references and the actual measurements are, again, almost overlapped. A closer look is provided by the tracking errors in Figure 63.c and Figure 63.d. First, it can be observed that the tracking error of the robotic platform is not zero, as assumed in the numerical simulations of Section7.3.2.1, thus distorting the optimal trajectory computed through the proposed method. This perturbation contributes to slightly bigger errors on the load side both in transient and after the end of the motion, if compared to the theoretical expectations. Additionally, some undesired parasitic motions increase the measured errors. For example, initial conditions are never exactly zero, as it can be observed in Figure 63.c and Figure 63.d, where oscillations are evident when $t < 2$ s (i.e., before the motion start). Since the control scheme is purely collocated and the inertia ratio approaches zero, such oscillations sum up with the forced response and can be seen at the end of the motion as well (due to the negligible damping). Furthermore, the unbalance of the load mass introduces a spinning-like motion of the load itself about the cable. Nonetheless, the tracking errors are small, and tracking is accurate: RMS tracking errors for the load are 0.68 mm and 0.92 mm along the *X* and *Y* directions, respectively, while the maximum residual error at the motion completion is 0.80 mm in both the directions.

Further evidence is provided by Figure 64, that represents the results in the Cartesian plane. Again, the reference and the actual displacement of the loads are almost overlapped, with a maximum contour error for the load equal to 2.12 mm, and an RMS value equal to 0.93 mm. As for the robotic platform, it also is affected by contour error whose maximum and RMS values are 1.70 mm and 0.55 mm respectively.

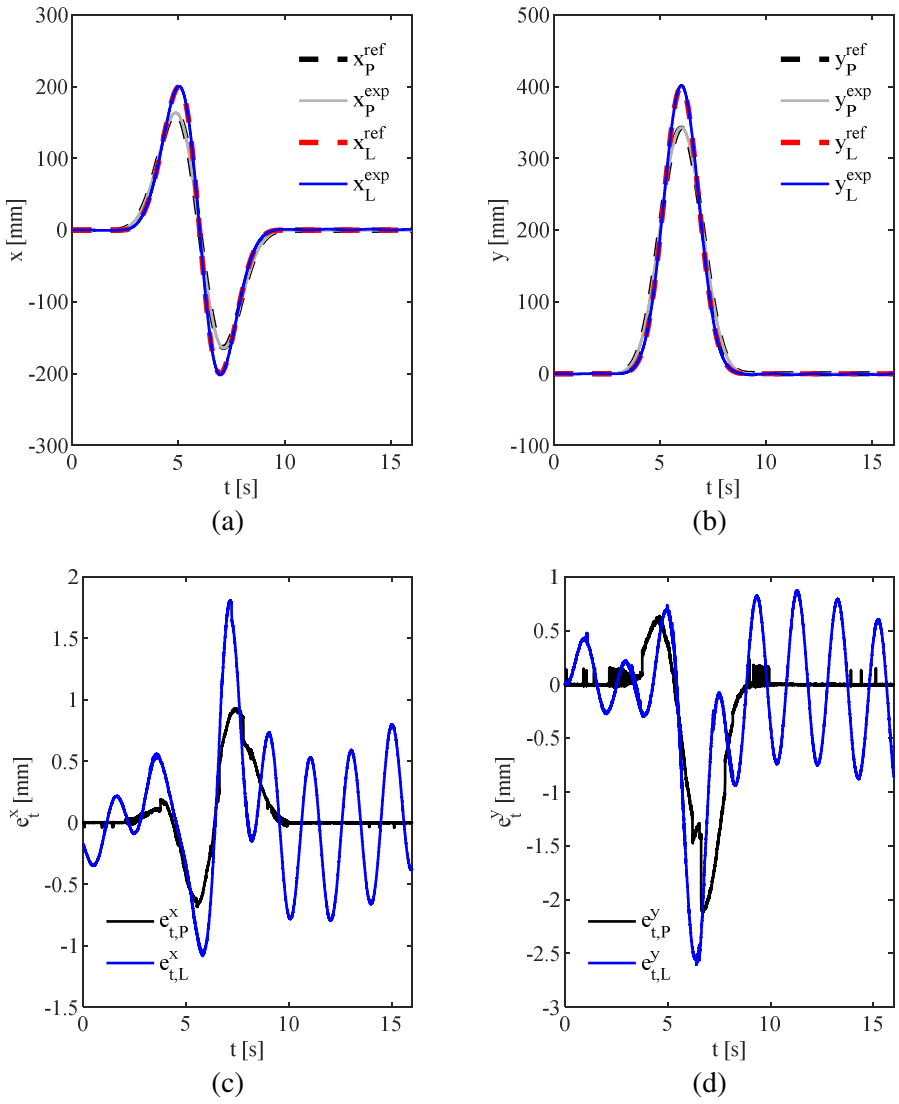


Figure 63. Test case 1. Time histories of the reference and experimental platform and load displacements in the Cartesian directions, (a) and (b), and the respective tracking errors, (c) and (d), with the proposed method.

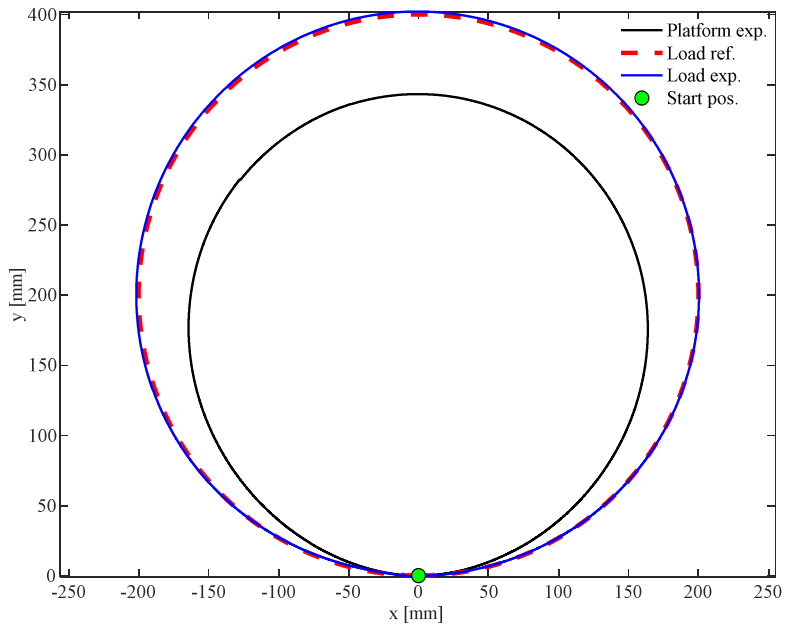


Figure 64. Test case 1. Reference and experimental platform and load displacements in the Cartesian plane, with the proposed method.

The effectiveness of the proposed method is further corroborated by comparison with the two benchmarks. Figure 65 shows the results obtained by commanding the platform with the unshaped reference: as the motion begins, the load starts an almost-uncontrolled oscillation with an amplitude of more than 400 mm. Figure 66 shows the results sported by the ZV input shaping method: while oscillations are properly controlled, the trajectory of the load is remarkably perturbated. Indeed, the transfer function of the shaper creates harmonic distortion of the shaped references compared to the unshaped one, thus resulting in a smaller desired radius for the load path. The lower the motion time, and the higher such a perturbation is. The tracking and contour errors sported by the input shaping are therefore remarkable larger than those achieved through the proposed method, as stated in Table 11, where the errors sported by the proposed method and the ZV are summarized.

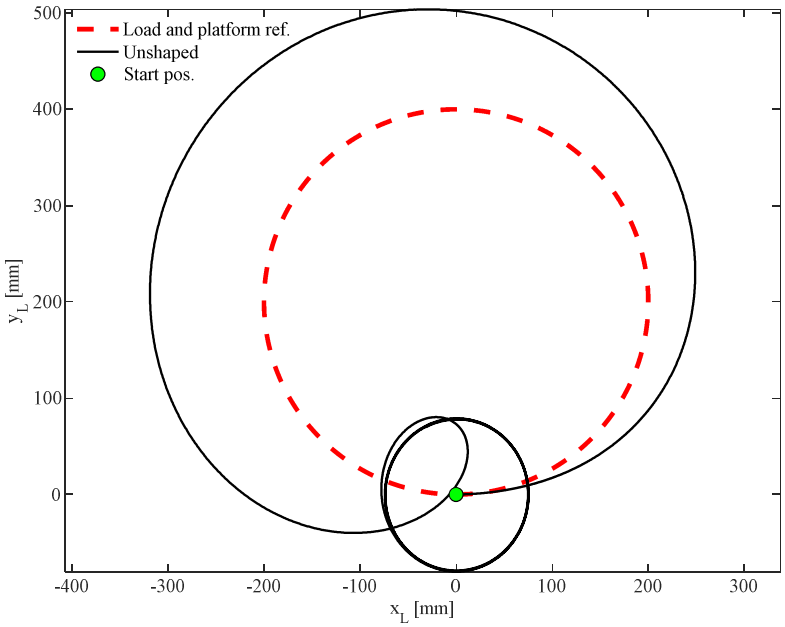


Figure 65. Test case 1. Reference and experimental load displacements in the Cartesian plane with the unshaped command.

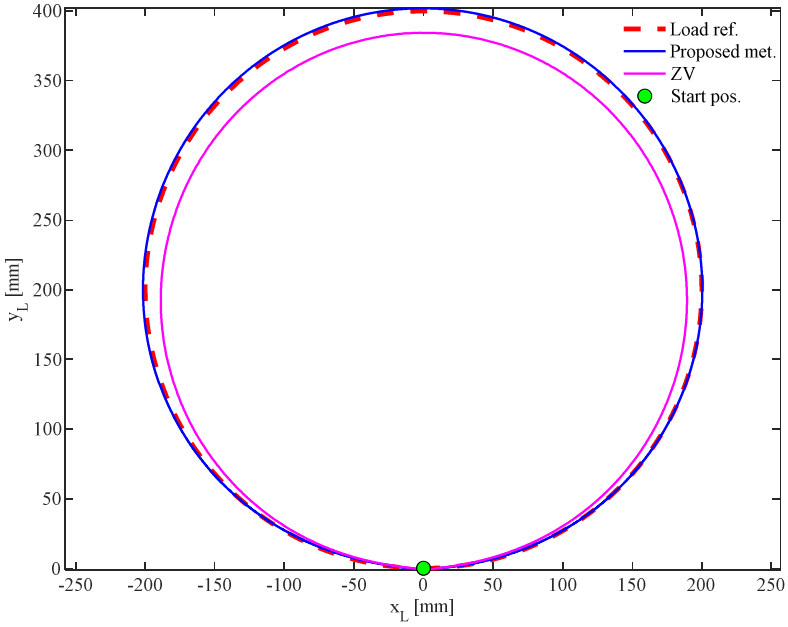


Figure 66. Test case 1. Reference and experimental load displacements in the Cartesian plane with the ZV input shaping and the proposed method.

Table 11. Test case 1. tracking and contour errors for the platform and the load.

Parameter	Proposed method	ZV
$e^{x}_{t,P_{RMS}}$ [mm]	0.42	0.62
$e^{y}_{t,P_{RMS}}$ [mm]	0.80	0.44
$e^{x}_{t,L_{RMS}}$ [mm]	0.68	16.98
$e^{y}_{t,L_{RMS}}$ [mm]	0.92	17.40
$e_{c,P_{RMS}}$ [mm]	0.55	0.44
$e_{c,L_{RMS}}$ [mm]	0.93	16.34

7.3.3. Test case 2: planar circular path with 4 s motion time – experimental results

The second test consists of another planar circular path, to be executed with the remarkable smaller motion time of 4 s; the same radius as in Section 7.3.2 is commanded, and again, a 7th-degree polynomial motion law is employed. The goal of this test is to highlight the method effectiveness in a more severe trajectory: as the motion time halves, therefore accelerations quadruple and hence higher oscillations are excited.

Figure 67.a and Figure 67.b show the time histories of the references and of the measured motion of both the robotic platform and the load: despite high accelerations and speed, the references and the actual measurements are almost overlapped. Clearly, slightly larger errors are obtained, both for the platform and consequently for the load. Indeed, reducing the motion time increase the harmonic contents of the motion law to be tracked by the robot and, therefore, larger harmonic distortion is obtained due to control bandwidth and interpolations. The robot platform motion has a maximum tracking error equal to 1.39 mm, that contributes to RMS values of the load tracking errors equal to 0.36 mm and 0.65 mm along the *X* and *Y* directions, respectively (it should be noted that such values are also affected by the initial oscillations that are not zero). Nonetheless, by considering the complexity of the test, the proposed method should be considered as very effective.

The representation in the Cartesian plane in Figure 69 corroborates the method effectiveness: the maximum contour error of the load is 5.21 mm, while its RMS value is 2.29 mm. In Figure 69 it is interesting to observe that the motion of the robotic platform resembles a *limaçon of Pascal*.
Finally, comparison with the unshaped command (Figure 68) and with the ZV input shaping (Figure 70) are proposed for benchmarking: the errors sported by both these approaches are remarkably larger than those of the proposed method, and therefore the motion of the load visibly does not approach the desired one.
Table 12 summarizes the errors obtained in this test.

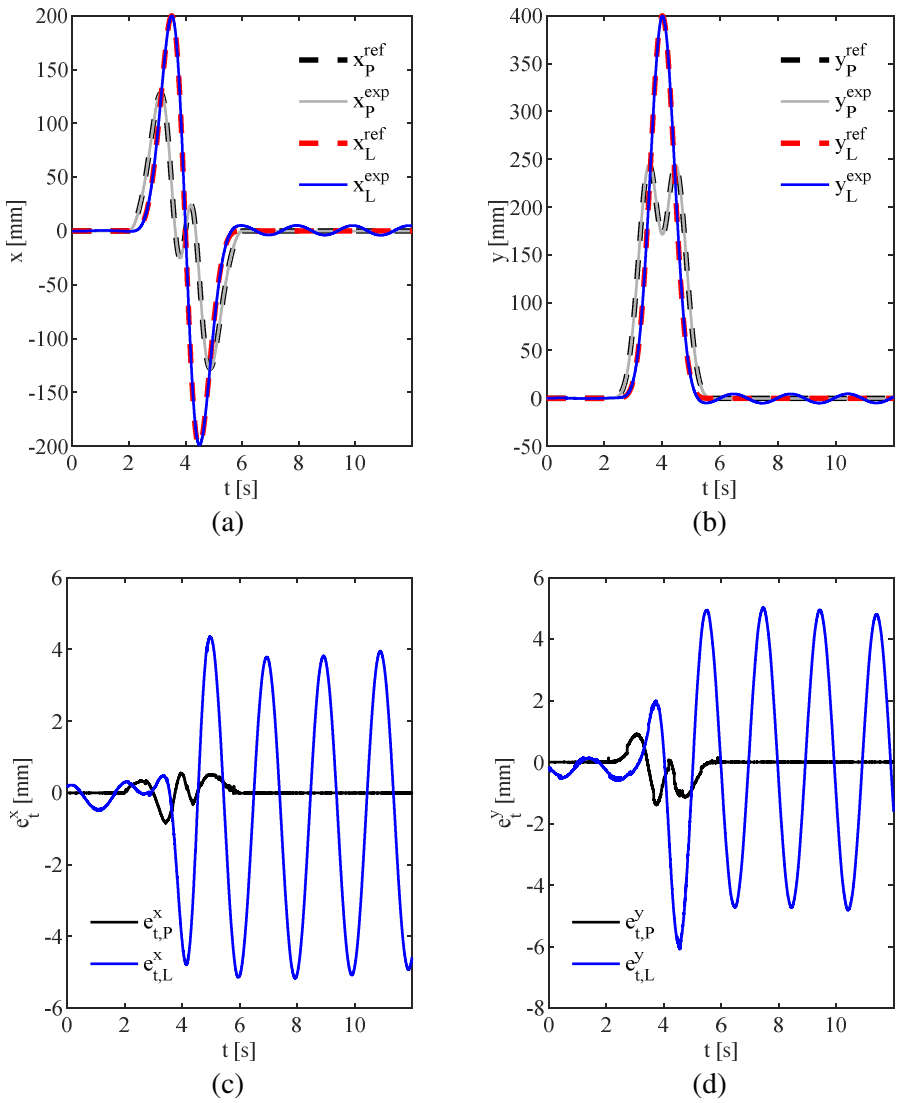


Figure 67. Test case 2. Time histories of the reference and experimental platform and load displacements in the Cartesian directions, (a) and (b), and the respective tracking errors, (c) and (d), with the proposed method.

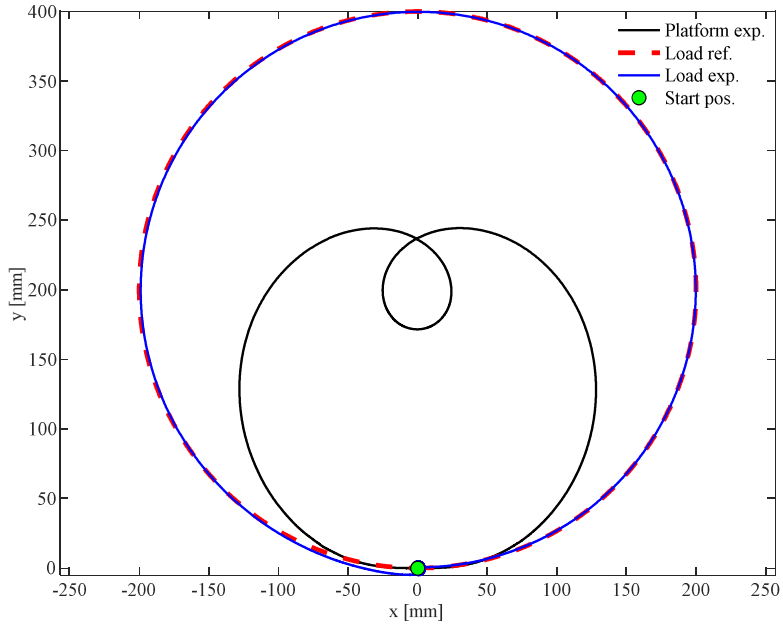


Figure 68. Test case 2. Reference and experimental platform and load displacements in the Cartesian plane, with the proposed method.

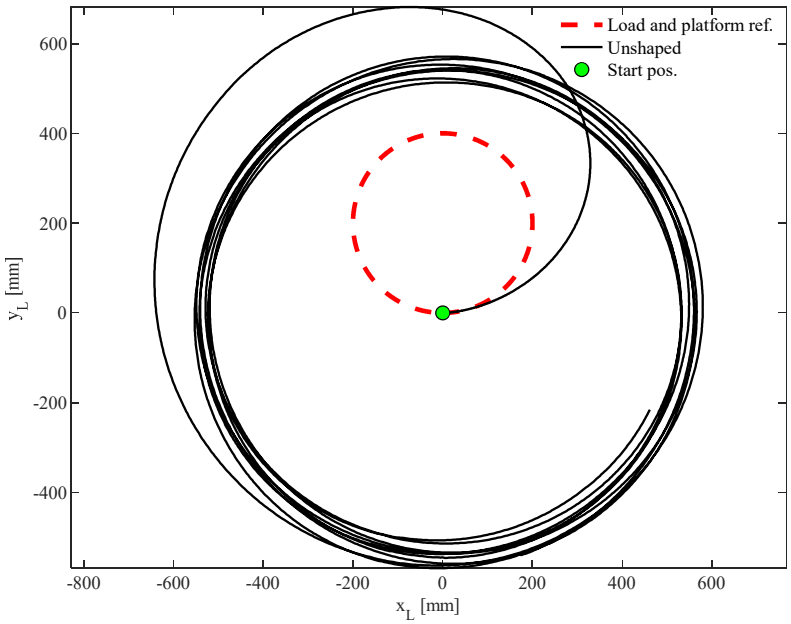


Figure 69. Test case 2. Reference and experimental load displacements in the Cartesian plane, with the unshaped command.

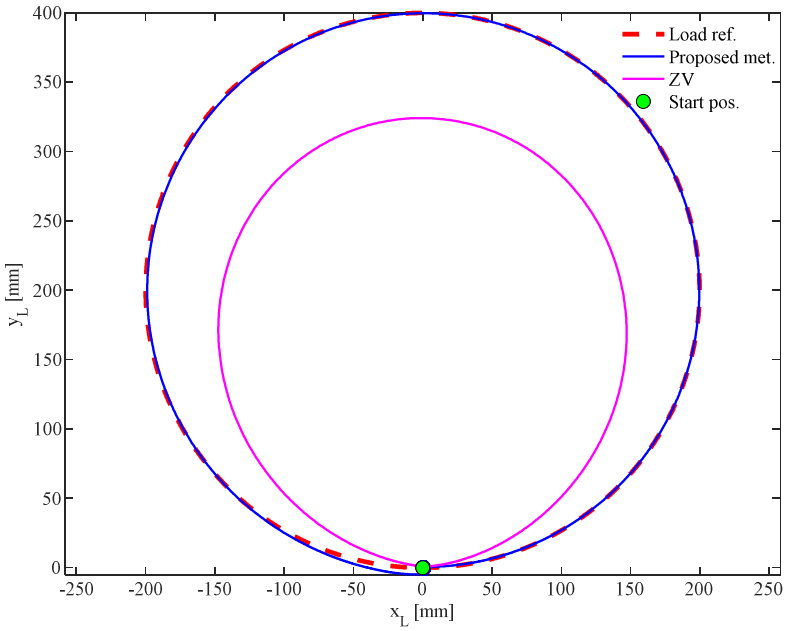


Figure 70. Test case 2. Reference and experimental load displacements in the Cartesian plane with the ZV input shaping versus the proposed method.

Table 12. Test case 2. tracking and contour errors for the platform and the load.

Parameter	Proposed method	ZV
$e^{x}_{t,P_{RMS}}$ [mm]	0.36	0.28
$e^{y}_{t,P_{RMS}}$ [mm]	0.65	0.24
$e^{x}_{t,L_{RMS}}$ [mm]	2.58	36.97
$e^{y}_{t,L_{RMS}}$ [mm]	2.75	37.48
$e_{c,P_{RMS}}$ [mm]	0.41	0.30
$e_{c,L_{RMS}}$ [mm]	2.29	41.21

7.3.4. Test case 3: planar rhombus path – experimental results

The third test case considers the task of tracking a planar rhombus whose sides are equal to 0.3 m. The motion of each side is performed through a 7th-degree polynomial, rest-to-rest motion law, with an overall motion time equal to 6 s (the motion starts at t = 2 s). This test is assumed for further benchmarking with the ZV input shaping: indeed, this spatial path can be executed by input shapers without distortion (at least in ideal conditions), since it consists of a sequence of 4 rest-to-rest motions, to be shaped independently. Figure 71 shows the tracking in time, through the time histories along the X and Y directions (Figure 71.a, Figure 71.b), and their respective tracking errors (Figure 71.c, Figure 71.d). Although the tracking errors of the load are not zero, and some residual oscillations remain uncompensated after the motion end, the result is valuable: the RMS tracking errors are 1.49 mm and 1.88 mm along the x and y directions, respectively, while the maximum residual oscillation after the motion end is equal to 3.80 mm and 4.52 mm respectively. The same figures shows that the platform tracking error is not zero during motion, as well, because of the already mentioned distortion and presence of some parasitic motion. The representation in the Cartesian plane is shown in Figure 72, which confirms the accuracy of path tracking: the maximum and RMS contour errors for the load are 4.11 mm and 2.11 mm, respectively.

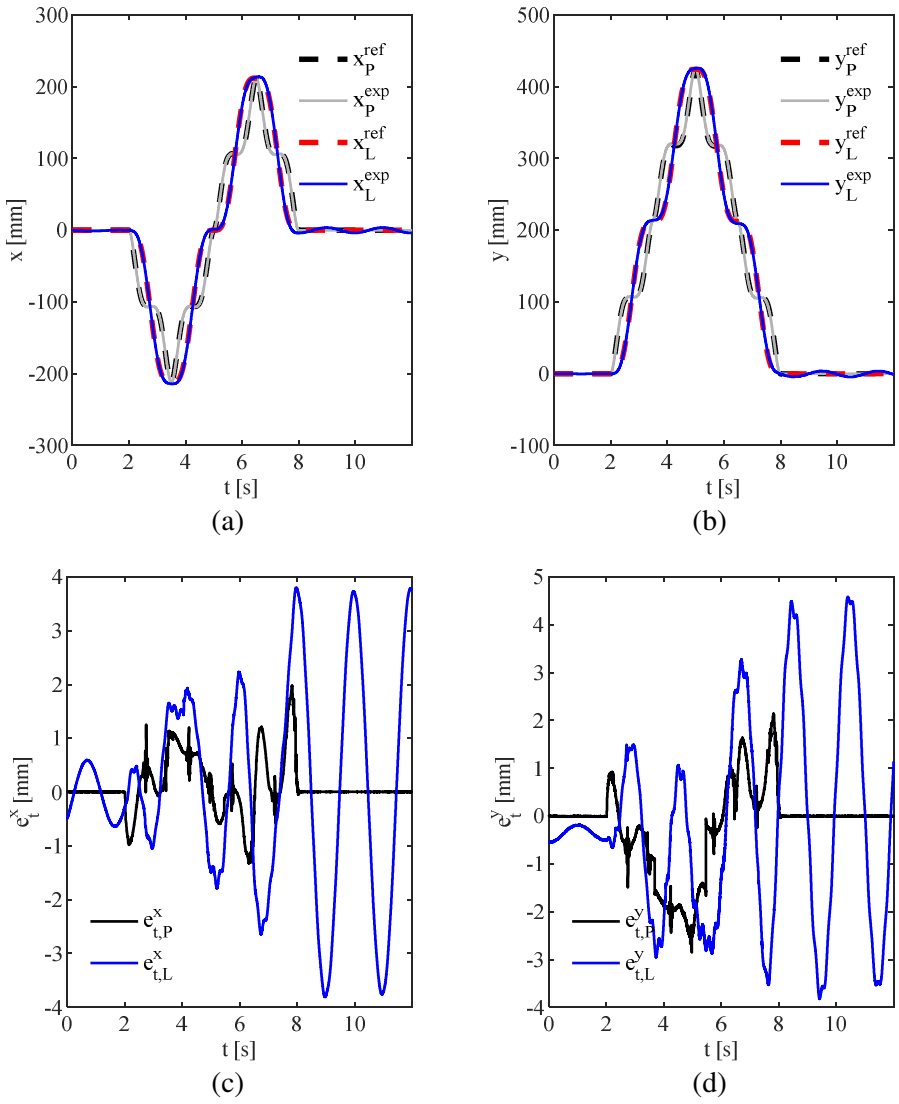


Figure 71. Test case 3. Reference and experimental platform and load displacements in the Cartesian directions, (a) and (b), and the respective trajectory tracking errors, (c) and (d), with the proposed method.

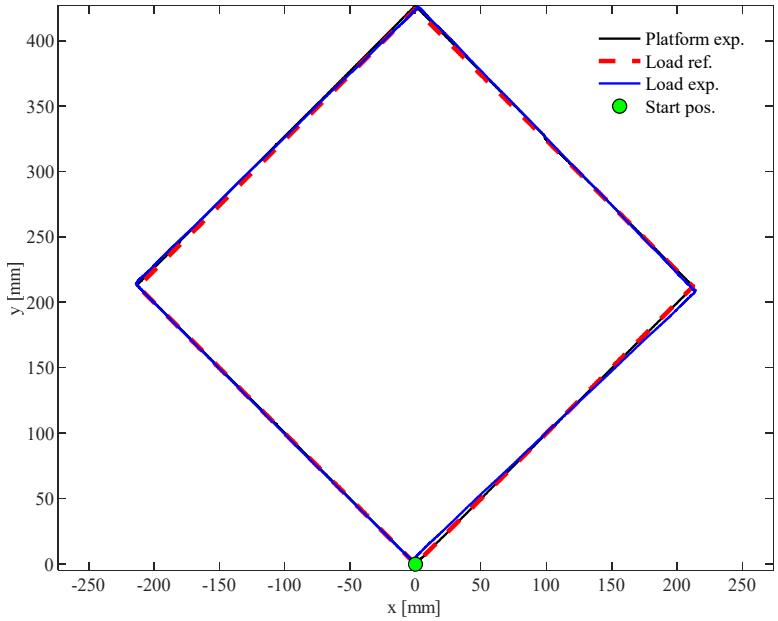
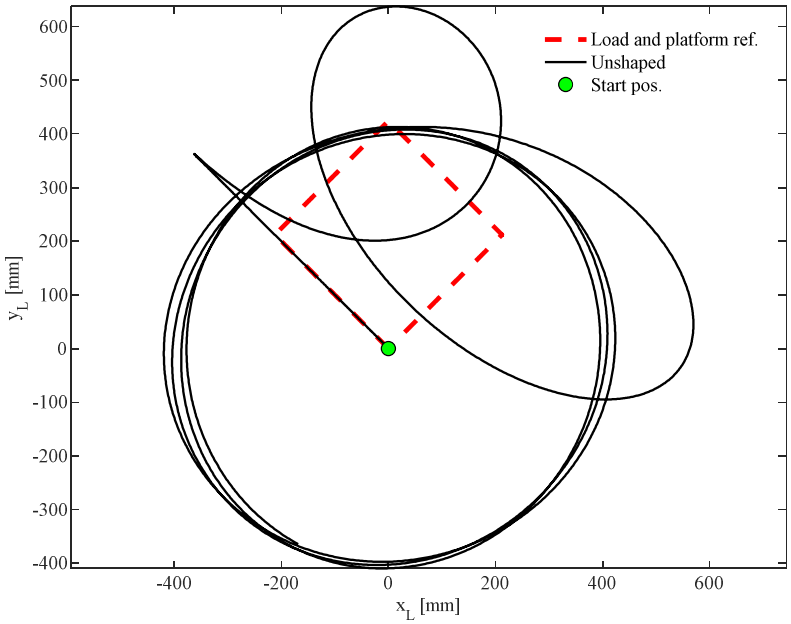
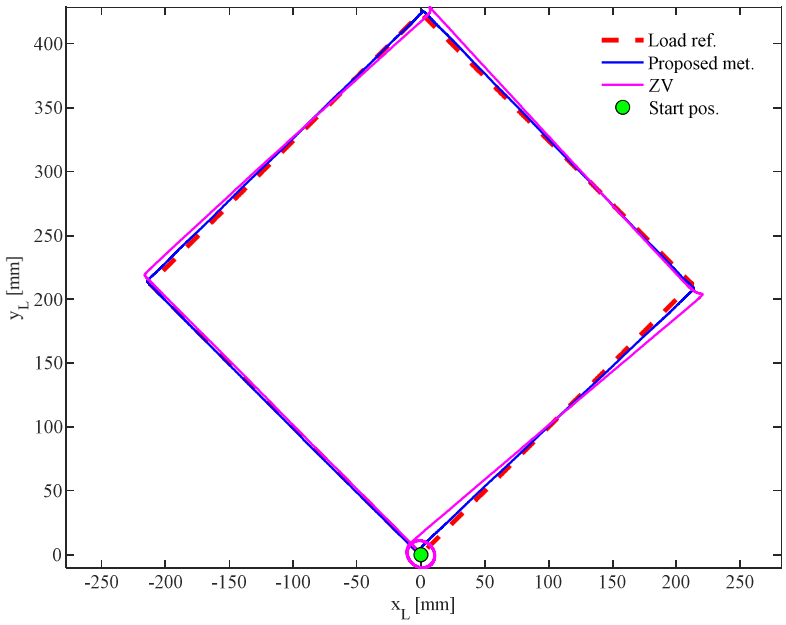


Figure 72. Test case 3. Reference and experimental platform and load displacements in the Cartesian plane, with the proposed method.

The effectiveness of the proposed approach is corroborated by the comparison with the two benchmarks in Figure 73, that is provided through the representation in the Cartesian plane. If no shaping is performed, very large load oscillations are excited, leading to errors up to 800 mm. If the motion profiles are shaped through the ZV input shaping, visible errors are obtained. Figure 74 provides a closer look to the tracking errors sported by the ZV: the maximum residual error is 14.46 mm, i.e. 3.8 times than the errors of the proposed method. This is caused by the unavoidable model uncertainty and by the distortion of the platform trajectory (that also affect the results of the proposed method). Such residual oscillations, that increase after each side of the rhombus, causes some motion components orthogonal to the desired path that create a visible difference between the commanded and the actual load path in Figure 73.b.

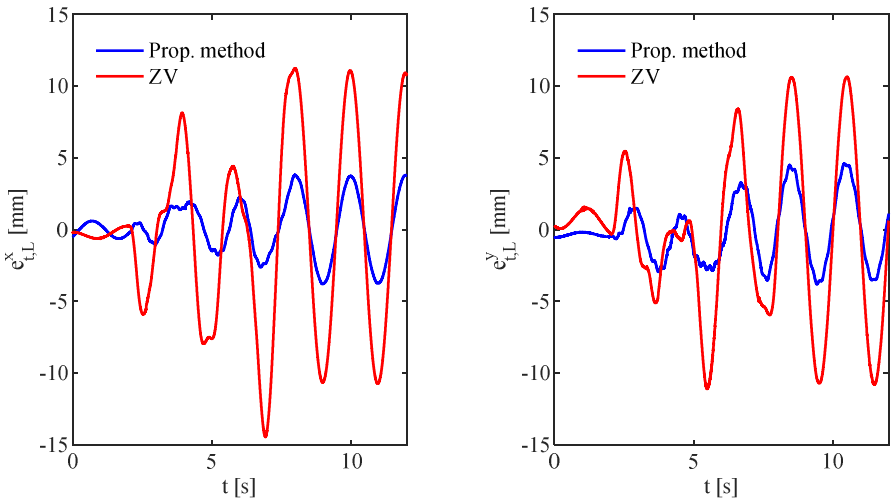


(a)



(b)

Figure 73. Test case 3. Reference and experimental load displacements in the Cartesian plane with different methods: unshaped command (a), ZV versus the proposed method (b).



(a) (b)

Figure 74. Test case 3. Experimental load tracking errors, (a) and (b): ZV versus the proposed method.

Table 13. Test case 3. tracking and contour errors for the platform and the load.

Parameter	Proposed method	ZV
$e^{x}_{t,P_{RMS}}$ [mm]	0.71	1.60
$e^{y}_{t,P_{RMS}}$ [mm]	1.25	1.54
$e^{x}_{t,L_{RMS}}$ [mm]	1.49	6.28
$e^{y}_{t,L_{RMS}}$ [mm]	1.88	4.66
$e_{c,P_{RMS}}$ [mm]	1.02	1.03

$e_{c,L_{RMS}}$ [mm]	2.11	7.65

7.3.5. Test case 4: planar spiral path – experimental results

The fourth test consists of a planar spiral path formulated as an Archimedean spiral, where the radius is defined as a function of the rotating angle. The starting value of the radius is set equal to 0.2 m while the ending one is 0. The motion time is equal to 6 s (the motion starts at t =2 s) and a 7th-degree polynomial motion law is assumed to execute 2 spatial revolutions.
The robot and payload time histories in the cartesian directions *x* and *y* are shown in Figure 75.a and Figure 75.b. The platform motion is tracked with a good accuracy even if the complexity of the path yields to a non-negligible maximum tracking error which is equal to 3.49 mm as shown in Figure 75.c. As a consequence, the maximum tracking errors for the load are respectively equal to 5.26 mm and 5.08 mm along the *X* and *Y* directions, as evidenced by Figure 75.c and Figure 75.d. The RMS tracking errors are listed in Table 14. Figure 76 confirms the effectiveness of the proposed method in path tracking: the RMS contour error for the load is just equal to 3.38 mm. In contrast, the application of the unshaped command shown in Figure 77.a leads to uncontrolled oscillations as the motion starts, whose amplitude is about 500 mm. If the ZV is adopted, the load oscillations are properly controlled; however, trajectory and path tracking are not satisfactory since the shapers remarkable modify the desired path leading to visible tracking and contour errors, as summarized also in Table 14: the RMS load contour error is equal to 37.98 mm, i.e. about 11 times the one obtained by the proposed method.

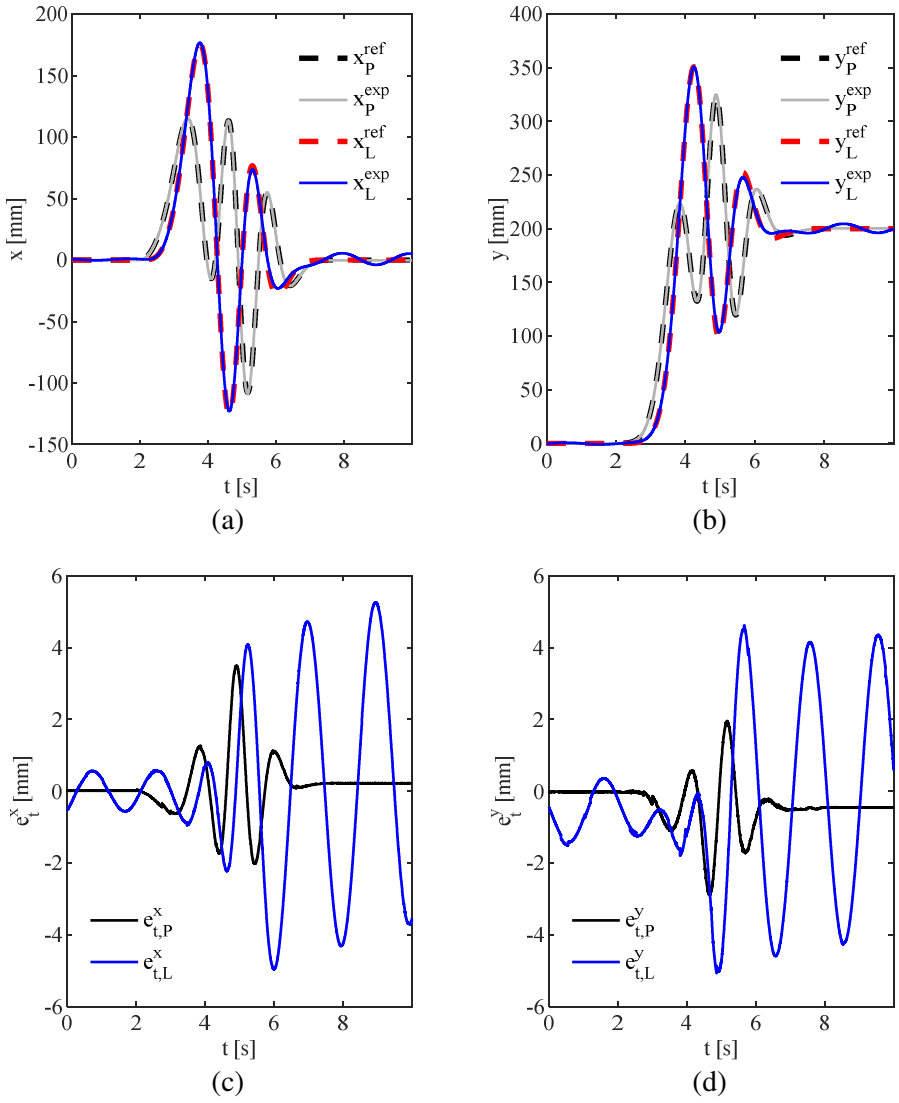


Figure 75. Test case 4. Reference and experimental platform and load displacements in the Cartesian directions, (a) and (b), and the respective trajectory tracking errors, (c) and (d), with the proposed method.

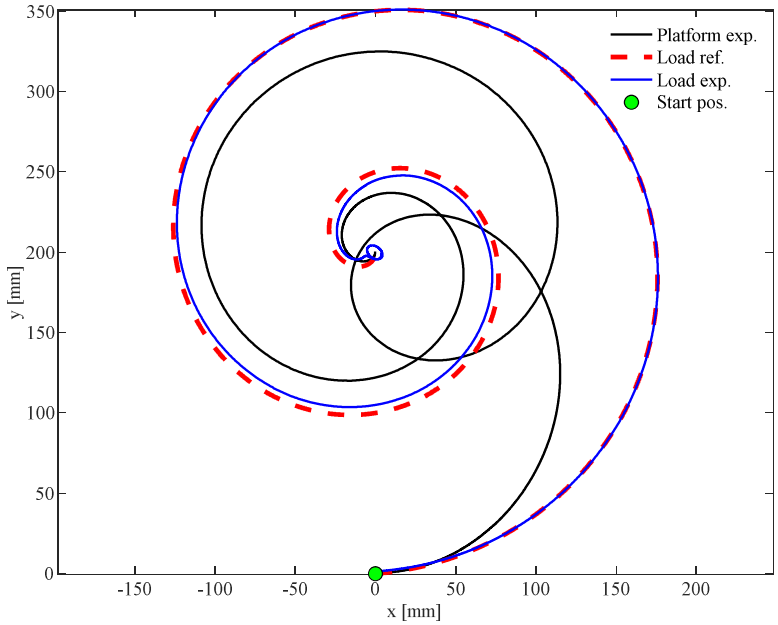


Figure 76. Test case 4. Reference and experimental platform and load displacements in the Cartesian plane, with the proposed method.

Table 14. Test case 4. tracking and contour errors for the platform and the load.

Parameter	Proposed method	ZV
$e^{x}_{t,P_{RMS}}$ [mm]	1.04	0.99
$e^{y}_{t,P_{RMS}}$ [mm]	0.94	1.25
$e^{x}_{t,L_{RMS}}$ [mm]	2.43	31.56
$e^{y}_{t,L_{RMS}}$ [mm]	2.57	34.34
$e_{c,P_{RMS}}$ [mm]	0.41	0.36
$e_{c,L_{RMS}}$ [mm]	3.38	37.98

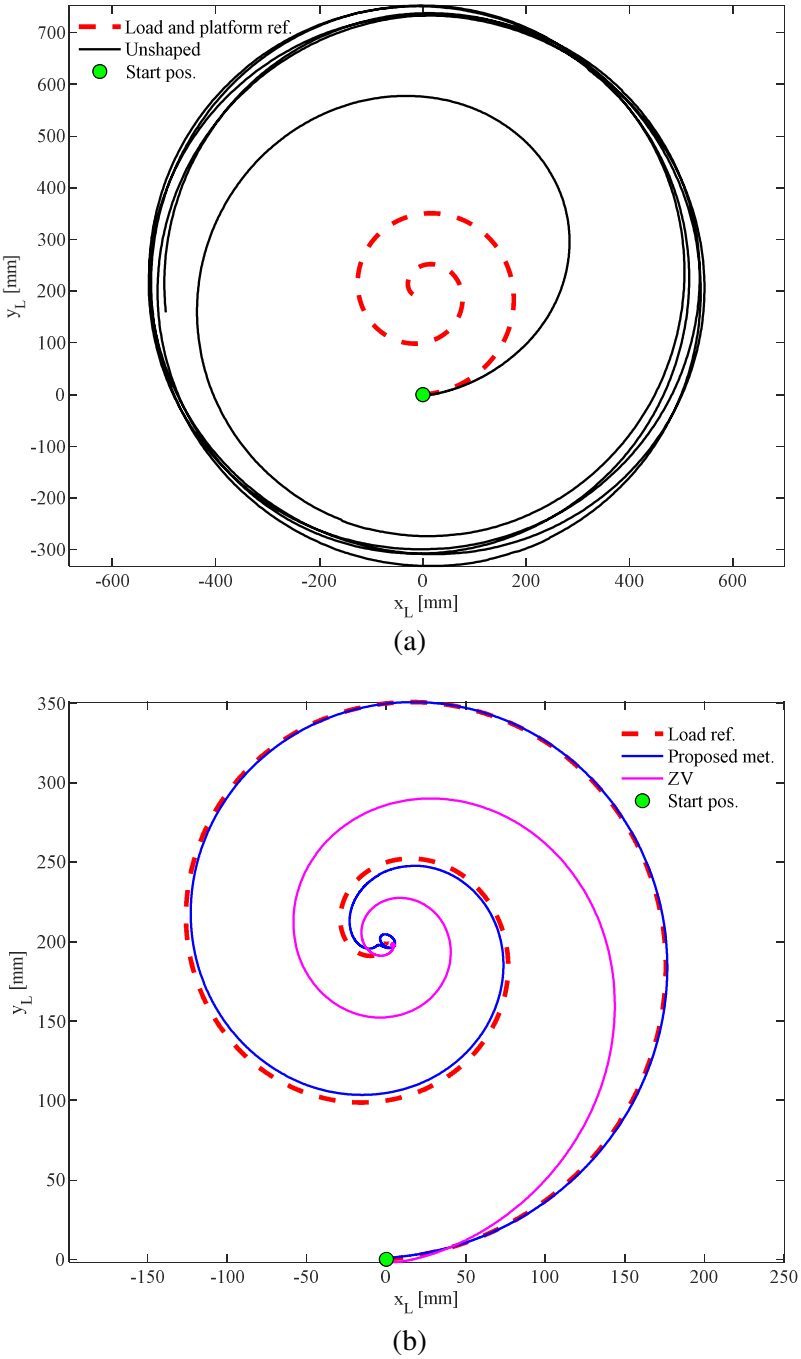


Figure 77. Test case 4. Reference and experimental load displacements in the Cartesian plane with different methods: unshaped command (a), ZV versus the proposed method (b).

7.4. Conclusions

This Chapter proposes a novel technique to path and trajectory planning in an underactuated, nonminimum-phase 4-DOF overhead crane. The method exploits stable inversion of the dynamic model governing the internal dynamics of the system, by leading to the commanded values of the actuated coordinates ensuring that the controlled output tracks the desired reference.
The method relies on the following steps:

- Partitioning of the model into actuated and unactuated coordinates.
- Definition of the controlled output and its representation as a nonlinear separable function in the actuated and unactuated coordinates.

- Formulation of the internal dynamics in term of unactuated coordinates and desired output.
- Stabilization of the internal dynamics through the concept of output redefinition.
- Integration of the stabilized equation of motions of the internal dynamics for computing the time-history of the unactuated coordinates.
- Computation of the commanded values of the actuated coordinates through nonlinear kinematic inversion.

The computation of the commanded forces to be exerted by the actuators driving the actuated coordinates is then demanded to the control scheme of such actuators, thus allowing the implementation of the proposed technique in proprietary controllers (as those of industrial robots) where the control scheme cannot be modified.

The method steps lead to a simple numerical solution of the motion planning problem, that is solved by adopting any scheme for solving ODEs, without requiring optimization problems. The obtained commanded trajectory is, finally, causal since it does not require pre-actuation.

Numerical results show that almost exact model inversion is performed, with negligible transient and steady state errors. The experimental application through a laboratory setup corroborates its effectiveness: tracking and contour errors are negligible in all the four tests, with different shapes and motion time. The comparison with the Zero Vibration input shaping and with the unshaped trajectories further proves the benefits of using the proposed technique.

8. Conclusion

This thesis has addressed the control of underactuated multibody systems with the goal to extend the current state-of-the-art with novel approaches and formulations, both in terms of feedback control and feedforward one. The main results can be therefore summarized as follows.

In Chapter 2 a new approach has been proposed to precise tip tracking in underactuated, nonminimum-phase multibody systems, by means of a new architecture of Model Predictive Control (MPC) that allows tracking time-varying references with negligible error and delay. Additionally, due to the reference embedding, it does not require implementing feedforward controllers, that are usually difficult to design in the case of underactuated, nonminimum-phase systems. The effectiveness of the proposed MPC-ERD is experimentally validated through a laboratory setup of a two-link underactuated, nonminimum-phase planar manipulator, showing the capability to ensure precise and zero-delay tracking of the tip time-varying reference, even in the presence of position measurement with a rough discretization. The experimental implementation and result corroborate the system stability and the controller feasibility due to a small computational effort.

The same underactuated system of Chapter 2 is adopted in Chapter 3 as test case for a different control paradigm, that is Pole Placement. More precisely, a robust pole placement is proposed in Chapter 3 and it is assessed in flexible linear systems with time delay. The method exploits state feedback control to perform the partial pole placement of the desired system poles and the robustness is quantified through the sensitivity function of the loop gain transfer function. The effectiveness and usefulness of the proposed method has been experimentally assessed against a challenging nonlinear and uncertain two-link flexible robot arm, whose flexibility arises due to the passive joint. Two controllers have been tuned and experimentally applied to show different features of the method, showing excellent results and agreement with the expected numerical results. Besides effectively handling the severe time delays assumed, the imposed robustness has allowed the controller to get rid of the unavoidable uncertainties.

Chapter 4 has proposed a novel controller architecture for path tracking control in Cable Driven Parallel Robots, that exploits the idea of Model Predictive Control (MPC). To handle the highly nonlinear dynamic model of a CDPR, two sequential actions have been included. The first action is used to evaluate the optimal cable tensions, while the second control action translates these optimal tensions into the required motor torques and hence into commanded currents for the motor current-loops. Three numerical test-cases have been proposed through a three-cable spatial CSPR with a lumped end-effector, showing good performances in path tracking tasks and ensuring the positiveness of the cable tensions. By looking at the contour error, it has been noticed that MPC-EI has been capable to reduce it by one order of magnitude, ensuring at the same time the verification of the boundaries on all cable tensions and, therefore, suggesting superior performance. The application of the controller to the model of a CSPR with very elastic cables, whose axial stiffness has been taken by the literature, shows that an almost-negligible deterioration of the path tracking performances is achieved, thanks to the wise controller design, highlighting that MPC algorithms represent good candidates for path and trajectory tracking with CSPRs. The analysis of the computational effort has shown that the CPU time at each time step is remarkably smaller than the time step of the control loop

The outcomes of Chapter 4 highlight that MPC algorithms represent a very good candidate for achieving high performances in Cable Driven Parallel Robots, also in the presence of highly flexible cables, therefore a Cable Suspended Parallel Robot is still considered in Chapter 5, but in this case with rigid cables, since axial flexibility of cables has resulted to lead to negligible effects. In order to keep cable tensions greater than a minimum positive boundary and, therefore, avoiding the system under investigation to become an underactuated one, in Chapter 5 the problem of trajectory tracking in the presence of time-varying references with Cable Driven Parallel Robots (CDPRs) has been considered and a two-stage controller has been proposed based on the previously presented MPC-ERD algorithm and the dynamic model of each actuator. The effectiveness of the proposed controller

has been validated numerically considering a CSPR in the presence of different test cases and a comparison with both MPC with just the embedded integrator and classic MPC, without embedded integrator, is performed, showing the supremacy of the novel MPC-ERD algorithm. In addition, the operating limits of the motors in terms of torque and speed have been embedded as well into the controller design, by considering their characteristic curve, acting as time-varying constraints for the cable tensions. The negligible delays and errors achieved with the numerical validation highlighted the benefits of the proposed approach.
Unless advanced controllers, such as the proposed MPC-ERD, are used, feedforward control must be considered with the goal to further improve the performance in terms of path and trajectory tracking. Therefore, Chapter 6 has proposed an enhanced inverse dynamics techniques for non-differentially flat, nonminimum-phase underactuated multibody systems, to obtain a feedforward control term without pre-actuation. By taking advantage of the idea already proposed in the literature, of properly approximating the desired output as a linearly combined output, and by performing the output redefinition to stabilize the internal dynamics, stable inversion of such a dynamic has been obtained. Then, by exploiting the exact model of the actuated sub-system, the computation of the feedforward forces has been obtained with just minor approximation. The effectiveness of the proposed method has been numerically assessed with three different test cases, highlighting different features of the method. In all the studied cases, the proposed method has showed higher accuracy in tracking the desired trajectory with tracking errors that are smaller than all the benchmarks. Furthermore, the proposed technique has been experimentally applied on a 2-DOF underactuated robotic arm, made by two rigid links and a passive joint. The absence of pre-actuation and the simple proposed formulation makes the method suitable for applications where no pre-calculation is allowed and, therefore, where a real-time calculation of feedforward is required.
Finally, since industrial robots are characterized by co-located controllers, where there is no direct measurement regarding the load, the only way to ensure good performances in terms of path and trajectory tracking of the load is by implementing a proper motion planning. With this goal, the idea proposed in Chapter 6 is exploited and re-interpreted in Chapter 7, where a novel technique to path and trajectory planning in an underactuated, nonminimum-phase 4-DOF overhead crane is proposed. The method exploits stable inversion of the dynamic model governing the internal dynamics of the system, by leading to the commanded values of the actuated coordinates ensuring that the controlled output tracks the desired reference. The computation of the commanded forces to be exerted by the actuators driving the actuated coordinates has been then demanded to the control scheme of such actuators, thus allowing the implementation of the proposed technique in proprietary controllers (as those of industrial robots) where the control scheme cannot be modified. The obtained commanded trajectory is, finally, causal since it does not require pre-actuation. Numerical results have shown that almost exact model inversion is performed, with negligible transient and steady state errors. The experimental application through a laboratory setup has corroborated its effectiveness.

BIBLIOGRAPHY

[1] M. Vidyasagar, On undershoot and nonminimum phase zeros, IEEE Trans. Automat. Contr. 31 (1986) 440–440. https://doi.org/10.1109/TAC.1986.1104289.

[2] A. De Luca, P. Lucibello, A.G. Ulivi, Inversion Techniques for Trajectory Control of Flexible Robot Arms, J. Robot. Syst. 6 (1989) 325–344. https://doi.org/10.1002/rob.4620060403.

[3] R. Seifried, Integrated mechanical and control design of underactuated multibody systems, Nonlinear Dyn. 67 (2012) 1539–1557. https://doi.org/10.1007/s11071-011-0087-2.

[4] G. Bastos, R. Seifried, O. Brüls, Analysis of stable model inversion methods for constrained underactuated mechanical systems, Mech. Mach. Theory. 111 (2017) 99–117. https://doi.org/10.1016/j.mechmachtheory.2017.01.011.

[5] Q. Tu, J. Rastegar, J.R. Singh, Trajectory synthesis and inverse dynamics model formulation and control of tip motion of a high performance flexible positioning system, Mech. Mach. Theory. 29 (1994) 959–968. https://doi.org/10.1016/0094-114X(94)90064-7.

[6] X. Wang, D. Chen, Output Tracking Control of a One-Link Flexible Manipulator via Causal Inversion, IEEE Trans. Control Syst. Technol. 14 (2006) 141–148. https://doi.org/10.1109/TCST.2005.859628.

[7] S.K. Ider, M.K. Özgören, V. Ay, Trajectory tracking control of robots with flexible links, Mech. Mach. Theory. 37 (2002) 1377–1394. https://doi.org/10.1016/S0094-114X(02)00043-5.

[8] Y. Tang, F. Sun, Z. Sun, Neural network control of flexible-link manipulators using sliding mode, Neurocomputing. 70 (2006) 288–295. https://doi.org/10.1016/j.neucom.2006.01.030.

[9] V. Feliu, J.A. Somolinos, A. García, Inverse dynamics based control system for a three-degree-of-freedom flexible arm, IEEE Trans. Robot. Autom. 19 (2003) 1007–1014. https://doi.org/10.1109/TRA.2003.819596.

[10] M.J. Shirazi, H. Salarieh, A. Alasty, R. Shabani, Tip tracking control of a micro-cantilever Timoshenko beam via piezoelectric actuator, JVC/Journal Vib. Control. 19 (2013) 1561–1574. https://doi.org/10.1177/1077546312447837.

[11] M.T. Burkhardt, P.S. Holzwarth, R. Seifried, Inversion Based Trajectory Tracking Control for a Parallel Kinematic Manipulator With Flexible Links, 11th Int. Conf. Vib. Probl. (2013) 9–12.

[12] C.F. Castillo-Berrio, S.N. Engin, V. Feliu-Batlle, A study on the tip tracking control of a single flexible beam, Trans. Inst. Meas. Control. 38 (2016). https://doi.org/10.1177/0142331216632869.

[13] M. Morari, J. H. Lee, Model predictive control: Past, present and future, Comput. Chem. Eng. 23 (1999) 667–682. https://doi.org/10.1016/S0098-1354(98)00301-9.

[14] P.S. Agachi, M.V. Cristea, A.A. Csavdari, B. Szilagyi, 2. Model predictive control, Adv. Process Eng. Control. (2016) 32–74. https://doi.org/10.1515/9783110306637-003.

[15] N. Dini, V.J. Majd, An MPC-based two-dimensional push recovery of a quadruped robot in trotting gait using its reduced virtual model, Mech. Mach. Theory. 146 (2020). https://doi.org/10.1016/j.mechmachtheory.2019.103737.

[16] T. Fan, C.W. de Silva, Dynamic modelling and model predictive control of flexible-link manipulators, Int. J. Robot. Autom. 23 (2008) 227–234.

[17] P. Boscariol, A. Gasparetto, V. Zanotto, Active position and vibration control of a flexible links mechanism using model-based predictive control, J. Dyn. Syst. Meas. Control. Trans. ASME. 132 (2010) 1–4. https://doi.org/10.1115/1.4000658.

[18] P. Boscariol, A. Gasparetto, V. Zanotto, Model predictive control of a flexible links mechanism, J. Intell. Robot. Syst. Theory Appl. 58 (2010) 125–147. https://doi.org/10.1007/s10846-009-9347-5.

[19] G. Takács, B. Rohal'-Ilkiv, Model predictive control algorithms for active vibration control:

A study on timing, performance and implementation properties, JVC/Journal Vib. Control. 20 (2014) 2061–2080. https://doi.org/10.1177/1077546313479993.
[20] R. Brancati, G. Di Massa, S. Pagano, A. Petrillo, S. Santini, A combined neural network and model predictive control approach for ball transfer unit–magnetorheological elastomer–based vibration isolation of lightweight structures, JVC/Journal Vib. Control. 26 (2020) 1668–1682. https://doi.org/10.1177/1077546320902316.
[21] A.G. Wills, D. Bates, A.J. Fleming, B. Ninness, S.O.R. Moheimani, Model predictive control applied to constraint handling in active noise and vibration control, IEEE Trans. Control Syst. Technol. 16 (2008) 3–12. https://doi.org/10.1109/TCST.2007.903062.
[22] N. Rahimi, T. Binazadeh, Robust model predictive control of heterogeneous time-delay multi-agent systems with polytopic uncertainties and input amplitude constraints, JVC/Journal Vib. Control. 27 (2021) 1098–1112. https://doi.org/10.1177/1077546320937222.
[23] C.Y. Lin, Y.H. Huang, Enhancing vibration suppression in a periodically excited flexible beam by using a repetitive model predictive control strategy, JVC/Journal Vib. Control. 22 (2016) 3518–3531. https://doi.org/10.1177/1077546314564451.
[24] D.Q. Mayne, J.B. Rawlings, C. V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica. 36 (2000) 789–814. https://doi.org/10.1016/S0005-1098(99)00214-9.
[25] A.B. Fontes, C.E.T. Dórea, M.R. Márcio, An iterative algorithm for constrained MPC with stability of bilinear systems, 2008 Mediterr. Conf. Control Autom. - Conf. Proceedings, MED'08. (2008) 1526–1531. https://doi.org/10.1109/MED.2008.4602048.
[26] P. Boscariol, A. Gasparetto, V. Zanotto, Vibration reduction in a flexible link mechanism through the synthesis of an MPC controller, IEEE 2009 Int. Conf. Mechatronics, ICM 2009. (2009). https://doi.org/10.1109/ICMECH.2009.4957182.
[27] S. Xiang, H. Gao, Z. Liu, C. Gosselin, Dynamic transition trajectory planning of three-DOF cable-suspended parallel robots via linear time-varying MPC, Mech. Mach. Theory. 146 (2020) 103715. https://doi.org/10.1016/j.mechmachtheory.2019.103715.
[28] J. Bettega, D. Richiedei, A. Trevisani, Using Pose-Dependent Model Predictive Control for Path Tracking with Bounded Tensions in a 3-DOF Spatial Cable Suspended Parallel Robot, Machines. 10 (2022) 453. https://doi.org/10.3390/machines10060453.
[29] J.N. Kutz, S.L. Brunton, B.W. Brunton, J.L. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, SIAM, 2016. https://doi.org/10.1137/1.9781611974508.
[30] H. Ouyang, D. Richiedei, A. Trevisani, Pole assignment for control of flexible link mechanisms, J. Sound Vib. 332 (2013) 2884–2899. https://doi.org/10.1016/j.jsv.2013.01.004.
[31] E.F. Camacho, C. Bordons Alba, Model Predictive Control, Springer-Verlag, London, 2007.
[32] L. Wang, Model predictive control system design and implementation using MATLAB®, Springer, London, 2009. https://doi.org/10.1007/978-1-84882-331-0.
[33] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28. https://doi.org/10.1017/S0022112010001217.
[34] M.S. Hemati, M.O. Williams, C.W. Rowley, Dynamic mode decomposition for large and streaming datasets, Phys. Fluids. 26 (2014) 1–6. https://doi.org/10.1063/1.4901016.
[35] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S.L. Brunton, J.N. Kutz, On dynamic mode decomposition: Theory and applications, J. Comput. Dyn. 1 (2014) 391–421. https://doi.org/10.3934/jcd.2014.1.391.
[36] D. Richiedei, A. Trevisani, Simultaneous active and passive control for eigenstructure assignment in lightly damped systems, Mech. Syst. Signal Process. 85 (2017) 556–566. https://doi.org/10.1016/j.ymssp.2016.08.046.
[37] R. Belotti, R. Caracciolo, I. Palomba, D. Richiedei, A. Trevisani, An Updating Method for Finite Element Models of Flexible-Link Mechanisms Based on an Equivalent Rigid-Link

System, Shock Vib. 2018 (2018). https://doi.org/10.1155/2018/1797506.
[38] L. Gualtieri, E. Rauch, R. Vidoni, Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review, Robot. Comput. Integr. Manuf. 67 (2021) 101998. https://doi.org/10.1016/j.rcim.2020.101998.
[39] D.G. Luenberger, Optimization by Vector Space Methods, Wiley Professional, 1969.
[40] D.A. Wismer, R. Chattergy, Introduction to nonlinear optimization: a problem solving approach, North-Holland, 1979.
[41] S. Devasia, D. Chen, B. Paden, Nonlinear Inversion-Based Output Tracking, IEEE Trans. Automat. Contr. 41 (1996) 930–942. https://doi.org/10.1109/9.508898.
[42] P. Seiler, A. Packard, P. Gahinet, An Introduction to Disk Margins [Lecture Notes], IEEE Control Syst. 40 (2020) 78–95. https://doi.org/10.1109/MCS.2020.3005277.
[43] R. Ariyatanapol, Y.P. Xiong, H. Ouyang, Partial pole assignment with time delays for asymmetric systems, Acta Mech. 229 (2018) 2619–2629. https://doi.org/10.1007/s00707-018-2118-2.
[44] J.J. Sinou, B. Chomette, Active vibration control and stability analysis of a time-delay system subjected to friction-induced vibration, J. Sound Vib. 500 (2021) 116013. https://doi.org/10.1016/j.jsv.2021.116013.
[45] N. Olgac, R. Sipahi, Dynamics and stability of variable-pitch milling, JVC/Journal Vib. Control. 13 (2007) 1031–1043. https://doi.org/10.1177/1077546307078754.
[46] K. Gu, S.I. Niculescu, Survey on recent results in the stability and control of time-delay systems, J. Dyn. Syst. Meas. Control. Trans. ASME. 125 (2003) 158–165. https://doi.org/10.1115/1.1569950.
[47] S. Li, C. Zhu, Q. Mao, J. Su, J. Li, Active disturbance rejection vibration control for an all-clamped piezoelectric plate with delay, Control Eng. Pract. 108 (2021) 104719. https://doi.org/10.1016/j.conengprac.2020.104719.
[48] I. Birs, C. Muresan, I. Nascu, C. Ionescu, A Survey of Recent Advances in Fractional Order Control for Time Delay Systems, IEEE Access. 7 (2019) 30951–30965. https://doi.org/10.1109/ACCESS.2019.2902567.
[49] T. Shi, P. Shi, S. Wang, Robust sampled-data model predictive control for networked systems with time-varying delay, Int. J. Robust Nonlinear Control. 29 (2019) 1758–1768. https://doi.org/10.1002/rnc.4483.
[50] M. Wu, J. Cheng, C. Lu, L. Chen, X. Chen, W. Cao, X. Lai, Disturbance estimator and smith predictor-based active rejection of stick–slip vibrations in drill-string systems, Int. J. Syst. Sci. 51 (2020) 826–838. https://doi.org/10.1080/00207721.2020.1744046.
[51] J.M. Araujo, T.L.M. Santos, Control of second-order asymmetric systems with time delay: Smith predictor approach, Mech. Syst. Signal Process. 137 (2020) 106355. https://doi.org/10.1016/j.ymssp.2019.106355.
[52] K. Natori, R. Oboe, K. Ohnishi, Stability analysis and practical design procedure of time delayed control systems with communication disturbance observer, IEEE Trans. Ind. Informatics. 4 (2008) 185–197. https://doi.org/10.1109/TII.2008.2002705.
[53] K. Natori, R. Oboe, K. Ohnishi, Robustness on Model Error of Time Delayed Control Systems with Communication Disturbance Observer, IEEJ Trans. Ind. Appl. 128 (2008) 709–717. https://doi.org/https://doi.org/10.1541/ieejias.128.709.
[54] W. Zhou, Y. Wang, Y. Liang, Sliding mode control for networked control systems: A brief survey, ISA Trans. 124 (2022) 249–259. https://doi.org/10.1016/j.isatra.2020.12.049.
[55] F. Nian, S. Shen, C. Zhang, G. Lv, Robust Switching Control for Force-reflecting Telerobotic with Time-varying Communication Delays, Proc. 32nd Chinese Control Decis. Conf. CCDC 2020. (2020) 1714–1719. https://doi.org/10.1109/CCDC49329.2020.9164407.
[56] J. Kautsky, N.K. Nichols, P. Van Dooren, Robust pole assignment in linear state feedback, Int. J. Control. 41 (1985) 1129–1155. https://doi.org/10.1080/0020718508961188.
[57] E.K. Chu, B.N. Datta, Numerically robust pole assignment for second-order systems, Int. J.

Control. 64 (1996) 1113–1127. https://doi.org/10.1080/00207179608921677.
[58] Y.M. Ram, J.E. Mottershead, Receptance method in active vibration control, AIAA J. 45 (2007) 562–567. https://doi.org/10.2514/1.24349.
[59] Y.M. Ram, A. Singh, J.E. Mottershead, State feedback control with time delay, Mech. Syst. Signal Process. 23 (2009) 1940–1945. https://doi.org/10.1016/j.ymssp.2008.04.012.
[60] Y.M. Ram, J.E. Mottershead, M.G. Tehrani, Partial pole placement with time delay in structures using the receptance and the system matrices, Linear Algebra Appl. 434 (2011) 1689–1696. https://doi.org/10.1016/j.laa.2010.07.014.
[61] J.M. Pratt, K. V. Singh, B.N. Datta, Quadratic partial eigenvalue assignment problem with time delay for active vibration control, J. Phys. Conf. Ser. 181 (2009). https://doi.org/10.1088/1742-6596/181/1/012092.
[62] K. V. Singh, R. Dey, B.N. Datta, Partial eigenvalue assignment and its stability in a time delayed system, Mech. Syst. Signal Process. 42 (2014) 247–257. https://doi.org/10.1016/j.ymssp.2013.08.028.
[63] R. Belotti, D. Richiedei, Pole assignment in vibrating systems with time delay: An approach embedding an a-priori stability condition based on Linear Matrix Inequality, Mech. Syst. Signal Process. 137 (2020) 106396. https://doi.org/10.1016/j.ymssp.2019.106396.
[64] N.J.B. Dantas, C.E.T. Dorea, J.M. Araujo, Partial pole assignment using rank-one control and receptance in second-order systems with time delay, Meccanica. 56 (2021) 287–302. https://doi.org/10.1007/s11012-020-01289-w.
[65] J.M. Araújo, Discussion on 'State feedback control with time delay,' Mech. Syst. Signal Process. 98 (2018) 368–370. https://doi.org/10.1016/j.ymssp.2017.05.004.
[66] J. Sherman, W.J. Morrison, Adjustment of an inverse matrix corresponding to a change in one element of a given matrix, Ann. Math. Stat. 21 (1950) 124–127. https://doi.org/10.1214/aoms/1177729893.
[67] R. Belotti, D. Richiedei, I. Tamellin, A. Trevisani, Pole assignment for active vibration control of linear vibrating systems through Linear Matrix Inequalities, Appl. Sci. 10 (2020). https://doi.org/10.3390/app10165494.
[68] D. Richiedei, I. Tamellin, Active control of linear vibrating systems for antiresonance assignment with regional pole placement, J. Sound Vib. 494 (2021) 115858. https://doi.org/10.1016/j.jsv.2020.115858.
[69] K.J. Astroem, R.M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, 2nd. ed, Princeton University Press, 2021.
[70] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control: Analysis and Design, Wiley: New York, 2007.
[71] H. Nyquist, Regeneration Theory, Nokia Bell Labs, 1932. https://doi.org/10.1002/j.1538-7305.1932.tb02344.x.
[72] T.S. Franklin, J.M. Araújo, T.L.M. Santos, Receptance-based robust stability criteria for second-order linear systems with time-varying delay and unstructured uncertainties, Mech. Syst. Signal Process. 149 (2021) 107191. https://doi.org/10.1016/j.ymssp.2020.107191.
[73] X. Zhao, Y. Li, P. Boonen, Intelligent optimization algorithm of non-convex function based on genetic algorithm, J. Intell. Fuzzy Syst. 35 (2018) 4289–4297. https://doi.org/10.3233/JIFS-169749.
[74] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st ed., Addison-Wesley Longman Publishing Co., 1989.
[75] K. Engelborghs, T. Luzyanina, D. Roose, Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL, ACM Trans. Math. Softw. 28 (2002) 1–21. https://doi.org/10.1145/513001.513002.
[76] S.B. Choi, M.S. Seong, S.H. Ha, Accurate position control of a flexible arm using a piezoactuator associated with a hysteresis compensator, Smart Mater. Struct. 22 (2013). https://doi.org/10.1088/0964-1726/22/4/045009.

[77] L. Scalera, P. Boscariol, G. Carabin, R. Vidoni, A. Gasparetto, Enhancing energy efficiency of a 4-DOF parallel robot through task-related analysis, Machines. (2020) 1–14. https://doi.org/10.3390/machines8010010.
[78] J. Vaughan, K.C.C. Peng, W. Singhose, W. Seering, Influence of remote-operation time delay on crane operator performance, in: IFAC Proc. Vol., IFAC, 2012: pp. 85–90. https://doi.org/10.3182/20120622-3-US-4021.00048.
[79] T. Slama, A. Trevisani, D. Aubry, R. Oboe, F. Kratz, Experimental analysis of an Internet-based bilateral teleoperation system with motion and force scaling using a model predictive controller, IEEE Trans. Ind. Electron. 55 (2008) 3290–3299. https://doi.org/10.1109/TIE.2008.928120.
[80] A. Trevisani, Underconstrained planar cable-direct-driven robots: A trajectory planning method ensuring positive and bounded cable tensions, Mechatronics. 20 (2010) 113–127. https://doi.org/10.1016/j.mechatronics.2009.09.011.
[81] S. Behzadipour, A. Khajepour, A New Cable-Based Parallel Robot with Three Degrees of Freedom, Multibody Syst. Dyn. 13 (2005) 371–383. https://doi.org/10.1007/s11044-005-3985-6.
[82] S.A. Khalilpour, R. Khorrambakht, H. Damirchi, H.D. Taghirad, P. Cardou, Tip-trajectory tracking control of a deployable cable-driven robot via output redefinition, Multibody Syst. Dyn. 52 (2021) 31–58. https://doi.org/10.1007/s11044-020-09761-x.
[83] E. Idà, T. Bruckmann, M. Carricato, Rest-to-Rest Trajectory Planning for Underactuated Cable-Driven Parallel Robots, IEEE Trans. Robot. 35 (2019) 1338–1351. https://doi.org/10.1109/TRO.2019.2931483.
[84] M.H. Korayem, H. Tourajizadeh, M. Bamdad, Dynamic load carrying capacity of flexible cable suspended robot: Robust feedback linearization control approach, J. Intell. Robot. Syst. Theory Appl. 60 (2010) 341–363. https://doi.org/10.1007/s10846-010-9423-x.
[85] M.A. Khosravi, H.D. Taghirad, Robust PID control of fully-constrained cable driven parallel robots, Mechatronics. 24 (2014) 87–97. https://doi.org/10.1016/j.mechatronics.2013.12.001.
[86] P. Boscariol, A. Gasparetto, V. Zanotto, Simultaneous position and vibration control system for flexible link mechanisms, Meccanica. 46 (2011) 723–737. https://doi.org/10.1007/s11012-010-9333-9.
[87] P. Boscariol, V. Zanotto, Design of a controller for trajectory tracking for compliant mechanisms with effective vibration suppression, Robotica. 30 (2012) 15–29. https://doi.org/10.1017/S0263574711000415.
[88] C. Vermillion, J. Sun, K. Butts, Model predictive control allocation for overactuated systems - Stability and performance, Proc. IEEE Conf. Decis. Control. (2007) 1251–1256. https://doi.org/10.1109/CDC.2007.4434722.
[89] M. Katliar, J. Fischer, G. Frison, M. Diehl, H. Teufel, H.H. Bülthoff, Nonlinear Model Predictive Control of a Cable-Robot-Based Motion Simulator, IFAC-PapersOnLine. 50 (2017) 9833–9839. https://doi.org/10.1016/j.ifacol.2017.08.901.
[90] R. Qi, M. Rushton, A. Khajepour, W.W. Melek, Decoupled modeling and model predictive control of a hybrid cable-driven robot (HCDR), Rob. Auton. Syst. 118 (2019) 1–12. https://doi.org/10.1016/j.robot.2019.04.013.
[91] J.C. Santos, A. Chemori, M. Gouttefarde, Redundancy Resolution Integrated Model Predictive Control of CDPRs: Concept, Implementation and Experiments, Proc. - IEEE Int. Conf. Robot. Autom. (2020) 3889–3895. https://doi.org/10.1109/ICRA40945.2020.9197271.
[92] A. Trevisani, Planning of dynamically feasible trajectories for translational, planar, and underconstrained cable-driven robots, J. Syst. Sci. Complex. 26 (2013) 695–717. https://doi.org/10.1007/s11424-013-3175-1.
[93] N. Zhang, W. Shang, S. Cong, Dynamic trajectory planning for a spatial 3-DoF cable-suspended parallel robot, Mech. Mach. Theory. 122 (2018) 177–196. https://doi.org/10.1016/j.mechmachtheory.2017.12.023.

[94] M.R. Homaeinezhad, S. Yaqubi, H.M. Gholyan, Simultaneous path-following and vibration control for uncertain nonlinear flexible mechanical systems without dependency on oscillatory mathematical model, Multibody Syst. Dyn. 51 (2021) 279–303. https://doi.org/10.1007/s11044-020-09757-7.
[95] W. Singhose, J. Vaughan, Reducing vibration by digital filtering and input shaping, IEEE Trans. Control Syst. Technol. 19 (2011) 1410–1420. https://doi.org/10.1109/TCST.2010.2093135.
[96] P. Boscariol, D. Richiedei, Robust point-to-point trajectory planning for nonlinear underactuated systems: Theory and experimental assessment, Robot. Comput. Integr. Manuf. 50 (2018) 256–265. https://doi.org/10.1016/j.rcim.2017.10.001.
[97] P. Boscariol, D. Richiedei, I. Tamellin, Residual vibration suppression in uncertain systems: A robust structural modification approach to trajectory planning, Robot. Comput. Integr. Manuf. 74 (2022) 102282. https://doi.org/10.1016/j.rcim.2021.102282.
[98] V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, R. Vidoni, Experimental validation of minimum time-jerk algorithms for industrial robots, J. Intell. Robot. Syst. Theory Appl. 64 (2011) 197–219. https://doi.org/10.1007/s10846-010-9533-5.
[99] P. Boscariol, A. Gasparetto, R. Vidoni, Planning Continuous-Jerk Trajectories for Industrial Manipulators, ASME 2012 11th Bienn. Conf. Eng. Syst. Des. Anal. ESDA 2012. 3 (2012) 127–136. https://doi.org/10.1115/ESDA2012-82103.
[100] J. García de Jalón, Kinematic and Dynamic Simulation of Multibody Systems The Real-Time Challenge, 1994. http://www.springerlink.com/index/10.1007/b118073.
[101] S.A. Khalilpour, A. Bourbour, R. Khorrambakht, S. Kariminasab, H.D. Taghirad, Forward Kinematics Resolution of A Deployable Cable Robot, 5th RSI Int. Conf. Robot. Mechatronics, IcRoM 2017. (2018) 27–32. https://doi.org/10.1109/ICRoM.2017.8466157.
[102] A. Berti, M. Gouttefarde, M. Carricato, Dynamic Recovery of Cable-Suspended Parallel Robots After a Cable Failure, Springer Proc. Adv. Robot. 4 (2018) 331–339. https://doi.org/10.1007/978-3-319-56802-7_35.
[103] X. Diao, O. Ma, Vibration analysis of cable-driven parallel manipulators, Multibody Syst. Dyn. 21 (2009) 347–360. https://doi.org/10.1007/s11044-008-9144-0.
[104] T. Heyden, C. Woernle, Dynamics and flatness-based control of a kinematically undetermined cable suspension manipulator, Multibody Syst. Dyn. 16 (2006) 155–177. https://doi.org/10.1007/s11044-006-9023-5.
[105] F. González, J. Kövecses, Use of penalty formulations in dynamic simulation and analysis of redundantly constrained multibody systems, Multibody Syst. Dyn. 29 (2013) 57–76. https://doi.org/10.1007/s11044-012-9322-y.
[106] D. Richiedei, I. Tamellin, A. Trevisani, A homotopy transformation method for interval-based model updating of uncertain vibrating systems, Mech. Mach. Theory. 160 (2021) 104288. https://doi.org/10.1016/j.mechmachtheory.2021.104288.
[107] J. Ros, A. Plaza, X. Iriarte, J. Ángeles, Exponential integration schemes in multibody dynamics, in: 2nd Jt. Int. Conf. Multibody Syst. Dyn., 2012.
[108] G.H. Golub, C.F. Van Loan, Matrix computations, 2013.
[109] Kollmorgen, AKD Electrical, (n.d.). https://www.kollmorgen.com/en-us/developer-network/akd-electrical/ (accessed March 7, 2022).
[110] A. Preumont, Collocated versus non-collocated control, (2011) 117–130. https://doi.org/10.1007/978-94-007-2033-6_6.
[111] G. Barrette, C.M. Gosselin, Determination of the dynamic workspace of cable-driven planar parallel mechanisms, J. Mech. Des. Trans. ASME. 127 (2005) 242–248. https://doi.org/10.1115/1.1830045.
[112] G.E.P. Box, Science and Statistics, J. Am. Stat. Assoc. 71 (1976) 791–799. https://doi.org/10.2307/2286841.
[113] R.L. Williams II, P. Gallina, Planar cable-direct-driven robots: Design for wrench exertion, J.

Intell. Robot. Syst. 35 (2002) 203–219. https://doi.org/10.1023/A:1021158804664.
[114] V. Mattioni, E. Ida', M. Carricato, Design of a Planar Cable-Driven Parallel Robot for Non-Contact Tasks, Appl. Sci. 11 (2021). https://doi.org/10.3390/app11209491.
[115] S. Seriani, P. Gallina, A. Wedler, A modular cable robot for inspection and light manipulation on celestial bodies, Acta Astronaut. 123 (2016) 145–153. https://doi.org/10.1016/j.actaastro.2016.03.020.
[116] R. V. Bostelman, J.S. Albus, N. Dagalakis, A. Jacoff, J. Gross, Applications of the NIST Robocrane, Robot. Manuf. 5 (1994) 403–410.
[117] J.P. Sousa, C.G. Palop, E. Moreira, A.M. Pinto, J. Lima, P. Costa, P. Costa, G. Veiga, A.P. Moreira, The SPIDERobot: A Cable-Robot System for On-site Construction in Architecture, Robot. Fabr. Archit. Art Des. 2016. (2016) 230–239. https://doi.org/10.1007/978-3-319-26378-6.
[118] S. Seriani, M. Seriani, P. Gallina, Workspace optimization for a planar cable-suspended direct-driven robot, Robot. Comput. Integr. Manuf. 34 (2015) 1–7. https://doi.org/10.1016/j.rcim.2015.01.004.
[119] X. Jiang, C. Gosselin, Dynamic point-to-point trajectory planning of a three-DOF cable-suspended parallel robot, IEEE Trans. Robot. 32 (2016) 285–288. https://doi.org/10.1109/digitalheritage.2015.7413883.
[120] S. Xiang, H. Gao, Z. Liu, C. Gosselin, Trajectory Optimization for a Six-DOF Cable-Suspended Parallel Robot with Dynamic Motions beyond the Static Workspace, Proc. - IEEE Int. Conf. Robot. Autom. (2020) 3903–3908. https://doi.org/10.1109/ICRA40945.2020.9196803.
[121] J. Bolboli, M.A. Khosravi, F. Abdollahi, Stiffness feasible workspace of cable-driven parallel robots with application to optimal design of a planar cable robot, Rob. Auton. Syst. 114 (2019) 19–28. https://doi.org/10.1016/j.robot.2019.01.012.
[122] G. Boschetti, R. Minto, A. Trevisani, Experimental investigation of a cable robot recovery strategy, Robotics. 10 (2021) 1–18. https://doi.org/10.3390/robotics10010035.
[123] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (2003) 733–764. https://doi.org/10.1016/S0967-0661(02)00186-7.
[124] B.P.M. Silva, B.A. Santana, T.L.M. Santos, M.A.F. Martins, An implementable stabilizing model predictive controller applied to a rotary flexible link: An experimental case study, Control Eng. Pract. 99 (2020). https://doi.org/10.1016/j.conengprac.2020.104396.
[125] J.C. Santos, A. Chemori, M. Gouttefarde, Model predictive control of large-dimension cable-driven parallel robots, Mech. Mach. Sci. 74 (2019) 221–232. https://doi.org/10.1007/978-3-030-20751-9_19.
[126] B. Zi, B.Y. Duan, J.L. Du, H. Bao, Dynamic modeling and active control of a cable-suspended parallel robot, Mechatronics. 18 (2008) 1–12. https://doi.org/10.1016/j.mechatronics.2007.09.004.
[127] C. Gosselin, P. Ren, S. Foucault, Dynamic trajectory planning of a two-DOF cable-suspended parallel robot, Proc. - IEEE Int. Conf. Robot. Autom. (2012) 1476–1481. https://doi.org/10.1109/ICRA.2012.6224683.
[128] J. Lamaury, M. Gouttefarde, Control of a large redundantly actuated cable-suspended parallel robot, Proc. - IEEE Int. Conf. Robot. Autom. (2013) 4659–4664. https://doi.org/10.1109/ICRA.2013.6631240.
[129] A. Aflakian, A. Safaryazdi, M. Tale Masouleh, A. Kalhor, Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose, Mechatronics. 50 (2018) 160–176. https://doi.org/10.1016/j.mechatronics.2018.02.005.
[130] L. Scalera, P. Gallina, S. Seriani, A. Gasparetto, Cable-Based Robotic Crane (CBRC): Design and Implementation of Overhead Traveling Cranes Based on Variable Radius Drums, IEEE Trans. Robot. 34 (2018) 474–485. https://doi.org/10.1109/TRO.2018.2791593.
[131] A. Zheng, M. Morari, Stability of Model Predictive Control with Mixed Constraints, IEEE

Trans. Automat. Contr. 40 (1995) 1818–1823. https://doi.org/10.1109/9.467664.
[132] E.C. Kerrigan, J.M. Maciejowski, Soft Constraints and Exact Penalty Functions in Model Predictive Control, Proc. UKACC Int. Conf. (2000).
[133] A. Richards, Fast Model Predictive Control with soft constraints, Eur. J. Control. 25 (2015) 51–59. https://doi.org/10.1016/j.ejcon.2015.05.003.
[134] D. Richiedei, Integrated selection of gearbox, gear ratio, and motor through scaling rules, Mech. Based Des. Struct. Mach. 46 (2018) 712–729. https://doi.org/10.1080/15397734.2018.1453366.
[135] R. Caracciolo, D. Richiedei, Optimal design of ball-screw driven servomechanisms through an integrated mechatronic approach, Mechatronics. 24 (2014) 819–832. https://doi.org/10.1016/j.mechatronics.2014.01.004.
[136] J. Eden, D. Lau, Y. Tan, D. Oetomo, On positive output controllability and cable driven parallel manipulators, 2015 Aust. Control Conf. AUCC 2015. (2015) 27–32.
[137] H. Ouyang, Pole assignment of friction-induced vibration for stabilisation through state-feedback control, J. Sound Vib. 329 (2010) 1985–1991. https://doi.org/10.1016/j.jsv.2009.12.027.
[138] J.M. Araújo, C.E.T. Dórea, L.M.G. Gonçalves, J.B.P. Carvalho, B.N. Datta, Robustness of the quadratic partial eigenvalue assignment using spectrum sensitivities for state and derivative feedback designs, J. Low Freq. Noise Vib. Act. Control. 37 (2018) 253–268. https://doi.org/10.1177/1461348418755614.
[139] M. Fliess, J. Levine, P. Martin, P. Rouchon, Flatness and defect of non-linear systems: Introductory theory and examples, Int. J. Control. 61 (1995) 1327–1361. https://doi.org/10.1080/00207179508921959.
[140] W. Blajer, K. Dziewiecki, K. Kołodziejczyk, Z. Mazur, Inverse dynamics of underactuated mechanical systems: A simple case study and experimental verification, Commun. Nonlinear Sci. Numer. Simul. 16 (2011) 2265–2272. https://doi.org/10.1016/j.cnsns.2010.04.047.
[141] L. Biagiotti, C. Melchiorri, Trajectory planning for automatic machines and robots, Springer Science & Business Media, 2008.
[142] D. Richiedei, A. Trevisani, Analytical computation of the energy-efficient optimal planning in rest-to-rest motion of constant inertia systems, Mechatronics. 39 (2016) 147–159. https://doi.org/10.1016/j.mechatronics.2016.05.004.
[143] S. Devasia, B. Paden, Stable Inversion for Nonlinear Nonminimum-Phase Time-Varying Systems, IEEE Trans. Automat. Contr. 43 (1998) 283–288. https://doi.org/10.1109/9.661082.
[144] J.A. Butterworth, L.Y. Pao, D.Y. Abramovitch, Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems, Mechatronics. 22 (2012) 577–587. https://doi.org/10.1016/j.mechatronics.2011.12.006.
[145] R. Seifried, Two approaches for feedforward control and optimal design of underactuated multibody systems, Multibody Syst. Dyn. 27 (2012) 75–93. https://doi.org/10.1007/s11044-011-9261-z.
[146] G. Oriolo, Y. Nakamura, Control of Mechanical Systems with Second-Order Nonholonomic Constraints: Underactuated Manipulators, in: 30th IEEE Conf. Decis. Control, IEEE, Brighton, UK, 1991. https://doi.org/10.1109/CDC.1991.261620.
[147] R. Belotti, D. Richiedei, I. Tamellin, A. Trevisani, Inverse structural modification for improving the design of harmonic excitation forces in underactuated vibration generators, in: ISMA, 2020.
[148] R. Belotti, D. Richiedei, I. Tamellin, A. Trevisani, Response optimization of underactuated vibration generators through dynamic structural modification and shaping of the excitation forces, Int. J. Adv. Manuf. Technol. 112 (2021) 505–524. https://doi.org/10.1007/s00170-020-06083-2.
[149] S. Okabe, Y. Kamiya, K. Tsujikado, Y. Yokoyama, Vibratory Feeding by Nonsinusoidal Vibration—Optimum Wave Form, J. Vib. Acoust. Trans. ASME. 107 (1985) 188–195.

https://doi.org/10.1115/1.3269243.
[150] R. Caracciolo, D. Richiedei, A. Trevisani, G. Zanardo, Designing vibratory linear feeders through an inverse dynamic structural modification approach, Int. J. Adv. Manuf. Technol. 80 (2015) 1587–1599. https://doi.org/10.1007/s00170-015-7096-0.
[151] D. Richiedei, A. Trevisani, Delayed-Reference Anti-Swing Control of Overhead Crane Systems, Int. Work. Adv. Motion Control. AMC. 1 (2008) 92–97. https://doi.org/10.1109/AMC.2008.4516047.
[152] Kollmorgen, S200 High Performance Compact Brushless Servo Drives, (n.d.). https://www.kollmorgen.com/sites/default/files/public_downloads/S200 Base Unit Reference Manual Rev C.pdf.
[153] L. Biagiotti, C. Melchiorri, Trajectory Planning for Automatic Machines and Robots, Springer-Verlag Berlin Heidelberg, 2008.
[154] W. Blajer, K. Kołodziejczyk, Control of underactuated mechanical systems with servo-constraints, Nonlinear Dyn. 50 (2007) 781–791. https://doi.org/10.1007/s11071-007-9231-4.
[155] W. Blajer, K. Kołodziejczyk, Motion planning and control of gantry cranes in cluttered work environment, IET Control Theory Appl. (2007). https://doi.org/10.1049/iet-cta.
[156] C.M. Pappalardo, D. Guida, Adjoint-based optimization procedure for active vibration control of nonlinear mechanical systems, J. Dyn. Syst. Meas. Control. Trans. ASME. 139 (2017). https://doi.org/10.1115/1.4035609.
[157] C.M. Pappalardo, D. Guida, Use of the adjoint method for controlling the mechanical vibrations of nonlinear systems, Machines. 6 (2018). https://doi.org/10.3390/machines6020019.
[158] P. Boscariol, A. Gasparetto, Robust model-based trajectory planning for nonlinear systems, JVC/Journal Vib. Control. 22 (2016) 3904–3915. https://doi.org/10.1177/1077546314566834.
[159] P. Boscariol, A. Gasparetto, Optimal trajectory planning for nonlinear systems: Robust and constrained solution, Robotica. 34 (2016) 1243–1259. https://doi.org/10.1017/S0263574714002239.
[160] W. Singhose, Command shaping for flexible systems: A review of the first 50 years, Int. J. Precis. Eng. Manuf. 10 (2009) 153–168. https://doi.org/10.1007/s12541-009-0084-2.
[161] S. Garrido, M. Abderrahim, A. Giménez, R. Diez, C. Balaguer, Anti-swinging input shaping control of an automatic construction crane, IEEE Trans. Autom. Sci. Eng. 5 (2008) 549–557. https://doi.org/10.1109/TASE.2007.909631.
[162] G. Peláez, G. Pelaez, J.M. Perez, A. Vizán, E. Bautista, Input shaping reference commands for trajectory following Cartesian machines, Control Eng. Pract. 13 (2005) 941–958. https://doi.org/10.1016/j.conengprac.2004.09.011.
[163] W.E. Singhose, N.C. Singer, Effects of input shaping on two-dimensional trajectory following, IEEE Trans. Robot. Autom. 12 (1996) 881–887. https://doi.org/10.1109/70.544771.
[164] X. Wu, X. He, Nonlinear Energy-Based Regulation Control of Three-Dimensional Overhead Cranes, IEEE Trans. Autom. Sci. Eng. 14 (2017) 1297–1308. https://doi.org/10.1109/TASE.2016.2542105.
[165] Y. Fang, W.E. Dixon, D.M. Dawson, E. Zergeroglu, Nonlinear coupling control laws for an underactuated overhead crane system, IEEE/ASME Trans. Mechatronics. 8 (2003) 418–423. https://doi.org/10.1109/TMECH.2003.816822.
[166] N. Sun, Y. Fang, X. Zhang, Energy coupling output feedback control of 4-DOF underactuated cranes with saturated inputs, Automatica. 49 (2013) 1318–1325. https://doi.org/10.1016/j.automatica.2013.01.039.
[167] X. Wu, X. He, Partial feedback linearization control for 3-D underactuated overhead crane systems, ISA Trans. 65 (2016) 361–370. https://doi.org/10.1016/j.isatra.2016.06.015.
[168] N. Sun, Y. Fang, H. Chen, B. Lu, Y. Fu, Slew/Translation Positioning and Swing

Suppression for 4-DOF Tower Cranes with Parametric Uncertainties: Design and Hardware Experimentation, IEEE Trans. Ind. Electron. 63 (2016) 6407–6418. https://doi.org/10.1109/TIE.2016.2587249.

[169] O. Boutalbi, K. Benmahammed, M. Boubezoula, Robust feedback linearizing controller design for an overhead crane system, 2017 6th Int. Conf. Syst. Control. ICSC 2017. (2017) 640–645. https://doi.org/10.1109/ICoSC.2017.7958723.

[170] W. O'Connor, H. Habibi, Gantry crane control of a double-pendulum, distributed-mass load, using mechanical wave concepts, Mech. Sci. 4 (2013) 251–261. https://doi.org/10.5194/ms-4-251-2013.

[171] H. Habibi, W. O'Connor, Payload motion control of rotary gantry and luffing cranes using mechanical wave concepts, Trans. Inst. Meas. Control. 39 (2017) 1649–1662. https://doi.org/10.1177/0142331216644044.

[172] G. Boschetti, R. Caracciolo, D. Richiedei, A. Trevisani, Moving the suspended load of an overhead crane along a pre-specified path: A non-time based approach, Robot. Comput. Integr. Manuf. 30 (2014) 256–264. https://doi.org/10.1016/j.rcim.2013.10.004.

[173] A. Gasparetto, P. Boscariol, A. Lanzutti, R. Vidoni, Path Planning and Trajectory Planning Algorithms: A General Overview, (2015). https://doi.org/10.1007/978-3-319-14705-5_1.

[174] J. Bettega, D. Richiedei, Feedforward control of a nonlinear underactuated multibody system, in: WMVC 2022, Mech. Mach. Sci., Springer, 2022.