UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

PH.D. THESIS IN

INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

MOTION PLANNING AND CONTROL METHODS FOR NONPREHENSILE MANIPULATION AND MULTI-CONTACT LOCOMOTION TASKS

DIANA SERRA

TUTOR: PROF. VINCENZO LIPPIELLO
CO-TUTOR: DR. ING. FABIO RUGGIERO

XXIX CICLO

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Summary

Many existing works in the robotic literature deal with the problem of nonprehensile dynamic manipulation. However, a unified control framework does not exist so far. One of the ambitious goals of this Thesis is to contribute to identify planning and control frameworks solving classes of nonprehensile dynamic manipulation tasks, dealing with the non linearity of their dynamic models and, consequently, with the inherited design complexity. Besides, while passing through a number of connections between dynamic nonprehensile manipulation and legged locomotion, the Thesis presents novel methods for generating walking motions in multi-contact situations. A short summary of each Chapter follows.

- Chapter 1 contains a general introduction about the problems related to motion planning and control for dynamic nonprehensile manipulation and legged locomotion tasks. The methodologies used to cope with such problems for some classes of dynamic manipulations are described.

- Chapter 2 is focused on robot control for nonprehensile planar rolling manipulation, where an actuated manipulator of a given shape, referred to as hand, manipulates an object without grasping it through rotations. The goal of this Chapter is the synthesis of passivity-based control laws for planar nonprehensile rolling manipulations between two arbitrary shapes. This control method exploits the energy and interconnection properties of the physical system taking advantage of the nonlinearities of the system; it allows to include the gravitational and the Coriolis effects in the control action, thus avoiding the use of approximate models. In addition, a systematic approach to simplify the control design to shape the energy of the closed-loop system is proposed. Both separable (with constant mass matrix) and non-separable (with non constant mass matrix) Hamiltonian systems are considered as case studies. Numerical tests are presented to show the performance of the control laws described in this Chapter.

- Chapter 3 is devoted to the optimal planning of robot dynamic manipulation tasks involving impacts between the manipulator and the manipulated object, such as the ball batting and the ball juggling tasks. The planning algorithms presented in this Chapter can be roughly divided in three main phases: the prediction of the motion of the ball, the selection of the configuration of the paddle at impact time, and the trajectory planning for the paddle. In the first two phases, the hybrid dynamics of these tasks is properly taken into account, together with nonlinear estimation methods to intercept the ball in time. Whereas, in the third phase a minimum acceleration planner in SE(3), based on the theory of differential geometry, is used to generate an optimal path for the manipulator. Simulations of different case studies validate the approach presented in this Chapter, along with a comparison with state-of-the-art methods.

- Chapter 4 motivates the connection between nonprehensile manipulation and bipedal legged locomotion. Since, for the dynamic walking task, natural motion results from the minimization of the centroidal angular momentum, a comparison between different approaches to deal with such minimization is presented. Successively, the Chapter introduces the main open problems related to multi-contact walking. When generating standard walking motions, different simplifying assumptions are valid, and allow to keep the model linear and well suited for preview based control. Nevertheless, while dealing with more complex tasks, like walking motion in multi-contact configurations, several simplifying assumptions are no more valid. Such a problem is addressed in this Chapter, by proposing a Newton method based on carefully crafted linear approximate models. This ensures that the dynamic feasibility is always satisfied exactly through a nonlinear model predictive control. Simulation results show that walking motions up and down stairs with an additional hand support can be efficiently computed with the proposed approach.

- Chapter 5 discusses the main contributions, remarks, and proposals for possible future developments of the results presented in the manuscript. Some ideas to investigate new lines of research related to the connections between dynamic nonprehensile manipulation and legged multi-contact locomotion modeling and control are detailed.

Preface

Some of the research and results described in this manuscript have undergone peer review and have been published in, or at the date of this printing is being considered for publication in, scientific conferences, journals, or books. In the following all the papers developed during my research work as Ph.D. student are listed.

- D. Serra, A. C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, *Time–optimal paths for a robotic batting task*, Springer Lecture Notes in Electrical Engineering, 2017, to appear.

- D. Serra, F. Ruggiero, V. Lippiello, and B. Siciliano, *A nonlinear least squares approach for nonprehensile dual-hand robotic ball juggling*, World Congress of the International Federation of Automatic Control (Tolouse, FR), 2017, to appear.

- L. R. Buonocore, J. Cacace, A. Donaire, F. Ficuciello, G. A. Fontanelli, V. Lippiello, A. Petit, F. Ruggiero, A. C. Satici, D. Serra, B. Siciliano, L. Villani, *Nonprehensile manipulation of deformable objects: The RoDyMan project*, 2017, submitted.

- D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P.B. Wieber, *A Newton method with always feasible iterates for nonlinear model predictive control of walking in a multi-contact situation*, IEEE–RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 932–937.

- D. Serra, A.C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, *An optimal trajectory planner for a robotic batting task: The table tennis example*, International Conference on Informatics in Control, Automation and Robotics (Lisbon, PT), 2016, pp. 90–101. Finalist for Best Student Paper Award.

- D. Serra, *Robot control for nonprehensile dynamic manipulation tasks*, International Conference on Informatics in Control, Automation and Robotics, Doctoral Constortium (Lisbon,PT), 2016, pp. 3–12. Best Ph.D. Project Award.

- V. Lippiello, F. Ruggiero, D. Serra, *Emergency landing for a quadrotor in case of a propeller failure: A PID based approach*, IEEE International Symposium on Safety, Security, and Rescue Robotics (Hokkaido, J), 2014, pp. 1–7.

- V. Lippiello, F. Ruggiero, D. Serra, *Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Chicago, IL, USA), 2014, pp. 4782 - 4788.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Dynamic Nonprehensile Manipulation

The robotic manipulation aims at finding a set of suitable controls to change the state of an object from an initial to a desired configuration. A manipulation task/action is said to be *nonprehensile* when the object is not directly caged between the fingertips or the palm of the hand. The force closure constraint does not hold during a nonprehensile manipulation action, and the grasp is then performed exploiting only unilateral constraints, allowing the object to roll, slide, and break the contacts with the manipulator [78]. Examples of nonprehensile manipulation tasks are in everyday life such as pushing objects, folding clothes, bringing wineglass on a tray, cooking in a pan, and many others. Nonprehensile manipulation is classified as *dynamic* when the dynamics of both the object and the robot are essential to successfully accomplish the task.

The analysis of objects with multiple frictional contacts poses two interesting problems. The forward problem, predicts the motion of an object given the applied force. Solving this is essential for simulation aspects. The inverse problem, predicts the applied forces producing a desired object motion, or the set of applied forces producing a desired contact mode. Solving this is essential for planning and control aspects. This Thesis is mainly devoted to the inverse problem.

In the last decades, with the increase of powerful technology in both sensing and actuation speed, it has become possible the use of robots to manipulate an object in a very fast way. Indeed, the robotic community began to deal with nonprehensile dynamic manipulation tasks. Planning and control methods for such kind of tasks were firstly introduced in [68] and [73]. Nowadays, this class

of manipulation problems is still rather far from being fully solved and applied in robotic applications. In such kind of manipulation, a kinematic chain cannot be always closed, with the drawback of not having a direct kinematics available. Moreover, when one of more contacts change their status, the dynamics of the system changes in a non-smooth manner making the design of the controller more involved. Since the object can perform a large variety of motions, most of nonprehensile systems are underactuated, arising controllability issues. Nevertheless, without adding complexity to the mechanical design, nonprehensile dynamic manipulation offers several advantages:

- increase of available robot actions;
- enlargement of the operative workspace;
- reduction of task execution time;
- improvement of the dexterity in dynamic tasks.

Applications of nonprehensile dynamic manipulation through robots span industrial, surgical, humanoid and service robotics. Examples of nonprehensile dynamic manipulation tasks can be found in the control of vibratory platforms, usually employed in those applications where it is not directly possible to manipulate small or damageable objects which cannot be grasped. The problems of pushing, orienting and assembling parts have been extensively investigated for factory automation, where commercial bowl feeders, or parts orienting systems, are employed. Also in surgical robotics, some nonprehensile manipulation tasks are performed, such as pushing away arteries and reshaping muscles or organs. In service robotics, the development of humanoid robots assisting elderly people in everyday tasks can be sped up with the extension of the set of available robot movements. In addition, there exists many similarities between dexterous nonprehensile manipulation and robotic walking. Therefore, the research in the field of robotic dynamic manipulation may have repercussions in the design of advanced legged robots, actuated prosthesis, or active exoskeletons, in the same way as, for example, the framework of frictional grasping can be applied to compensate disturbance and balance a legged robot [85]. The following Section introduces the motivation behind the study of methods for motion planning and control of legged locomotion.

1.1.1 From Nonprehensile Manipulation to Multi-Contact Walking

Several analogies exist between the dynamic nonprehensile manipulation and walking tasks within locomotion applications. In particular, walking shares some hybrid contact/non-contact conditions with the manipulation task. In fact, methods for grasp analysis deal with the same constraints on contact forces and center of mass (CoM) position that arise for legged robots on irregular and steep terrain, since during manipulation the surface of the object is rarely flat and horizontal [14, 19, 85]. In the same way as grasp taxonomies are used to understand how the hand can hold an object, also taxonomies of the whole body pose for balancing are proposed in [17], as tools for autonomous decision making. This classification of multi-contact whole body poses highlights that almost all the so-called *whole-body grasps* are nonprehensile [10]. Such manipulation intrinsically involves multiple contacts and interaction between complex shapes, and then it can be related to legged locomotion in *multi-contact* situations [27, 104]. For this reason, the generation of motion for walking on uneven terrain in a multi-contact situation is inspected in this Thesis. A state constrained optimal control is designed with the aim to identify the set of possible - *dynamically feasible* - movements for a bipedal walking robot in a set of multi-contact configurations. A reference motion in case of interaction of a humanoid robot with external supports while walking on stairs is generated, extending the work [18]. Nonlinear transformations of the system dynamics are designed with the aim to keep the computational burden compatible with the real-time [105].

1.2 Methodology

In order to understand and recognize the dynamic effects playing a relevant role in nonprehensile manipulation and legged locomotion tasks, the robot is firstly modeled as an ideal system with a simplified dynamics and with available perceptual information. Under these hypotheses, it becomes easier to design planners and controllers where dynamic manipulation issues can be brought back. In a subsequent stage, the presence of the robot dynamics can be considered. The effects of the robot dynamics on the accomplishment of tasks can be evaluated, with the resulting adjustment of the corresponding implementation. Joint-space controllers can be designed to consider also the effects of motor constraints in the whole system. Additionally, the issues related to the perceptual information retrieved from sensors can be considered. It is remarkable that the real-time re-

Figure 1.1: Bottom-up approach to develop a unified control methodology for dynamic manipulation.

quirement posed by the fast dynamics involved in this kind of dynamic tasks is essential.

A bottom-up methodology is employed to deal with the generalization of the proposed planning and control methods to classes of dynamic nonprehensile manipulation tasks. The Fig. 1.1 schematizes such approach, in which the class of nonprehensile sliding manipulation [69] has been included, even if this is out of the scope of this Thesis. As it is possible to figure out so far, a nonprehensile manipulation action is a complex, skillful and dexterous task. It can be usually undertaken by splitting in simpler subtasks, usually called *primitives*, such as rolling, pushing, throwing, batting, dynamic catching, and so on. A supervisory control is then required to detect, identify and switch between the available primitives to perform the original complex dynamic nonprehensile manipulation task. The pursued methodology is then to study step by step each nonprehensile manipulation primitive, equipping this last with the proper motion planner and controller. The primitives considered in this Thesis are related to:

- rolling tasks, involving a rolling constraint for balancing;
- impact tasks, which exhibit a hybrid dynamics due to intermittent contacts between the object and the robot.

This feature of impact manipulation resembles the dynamics of walking robots. Therefore, due to their similarities, walking task is also considered and regarded as a dynamic nonprehensile manipulation task.

In this Thesis, passivity-based approaches are considered for the class of nonprehensile planar rolling. Some benchmarks within the class of rolling tasks are the ball and beam system [42, 46, 83, 96], the circular ball and beam system [7, 8, 99], the disk on disk [94, 95], and the butterfly robot [71, 113]. Besides the classical Euler-Lagrange modeling approach to robotic systems, also the Port-Hamiltonian systems formalism has been used for such tasks. Both Euler-Lagrange and Port-Hamiltonian formalism are energy-based representations exposing in complementary manners different physical properties of the systems related to the ways of energy processing, power flow and interconnection structure. This allows to exploit them for the design of control algorithms for tasks with non-negligible dynamics. Passivity-based control is chosen with the aim to exploit its potentiality to stabilize the system with the contribution of gravitational force, or centrifugal forces, avoiding the cancellation of the nonlinear dynamics of the system, that for dynamic manipulation tasks might play a relevant role. Shaping the energy of a mechanical system permits to deal with not just stabilization, but also performance objectives [33]. In particular, the method of *interconnection and damping assignment passivity-based control* (IDA-PBC) is considered [84]. The IDA-PBC method provides analytic control laws, taking advantage of the Lyapunov control theory to deal with the stability issues. However, a bottleneck of the approach is the solution of the so-called *matching equations*, which are typically a set of partial differential equations. In fact, a control design using IDA-PBC is quite challenging to apply to the problem of re-configuring a 3D rolling sphere on a plate [15, 61], or for impact manipulation tasks [16, 98, 116]. Even if in the literature the control of rolling is treated in detail also in the 3D situation [75, 34], in this Thesis the focus is on the 2D case to better the IDA-PBC framework, and future investigation might lead to the 3D extension.

Optimal planning and control theories are investigated for the class of impact manipulation and for legged locomotion task, respectively. Regarding planning and control of impact tasks, such as juggling [16, 38, 81, 88, 89, 98, 116], dribbling [12, 13, 43], batting [103, 121], etc. Such tasks require so high velocities and precision that robotic companies take them as examples to display the high performances of their products. For instance, Kuka has chosen the table tennis game to promote its wares in a thrilling commercial spot[1], showing the potential

[1] `https://youtu.be/tIIJME8-au8`

abilities of robots. The Omron automation company has also broadcast a video showing its parallel Delta robot playing table tennis and coaching humans at CEATEC Japan 2015 exhibition[2]. In this Thesis, exploiting the emerging development of fast solvers of linear and nonlinear problems [37, 66], and relying on the technological advancement of the measurement systems, robust optimal motion planning techniques are proposed to deal with nonprehensile manipulation and locomotion tasks. For impact manipulation tasks [106, 107, 108], the proposed methods are inspired by the works [26] and [120].

For the legged locomotion tasks, *model predictive control* (MPC) is the framework employed in this Thesis. The MPC approach is an optimal control method which, at each sampling time, requires the three following steps: measurement of the actual state; computation of the control that optimizes a given state-dependent cost function, on a finite horizon, starting from the current discrete time; and, finally, application of the control input at the first-time index only [74]. A main drawback of the standard approach is that the available theoretical results of stability are presently limited to the linear case or to very particular classes of nonlinear systems. For example, the *nonlinear model predictive control* (NMPC) presents stability properties under certain assumptions. However, it has some ability to handle constraints, which makes it well suited to the problem of walking pattern synthesis and then control of bipedal robots, since they are typically subject to unilateral constraints [53].

The first Sections of the Chapters 2, 3 and 4 briefly introduce the research context, the motivations, and the main objectives of each investigated research topic. Whereas, the corresponding last Sections are focused on the explanation of the main scientific results and contribution of each Chapter.

[2] `https://youtu.be/6MRxwPHH0Fc`

Chapter 2

Passivity-based Control of Nonprehensile Planar Rolling Manipulation

2.1 Introduction

This Chapter is focused on the nonprehensile planar rolling primitive, where an actuated manipulator of a given shape, referred to as hand, manipulates an object without grasping it, through rotations. The object can only roll upon the hand's shape, and the hand is actuated by a controlled torque, while the object is unactuated. The goal of this work is the synthesis of passivity-based control laws for planar rolling manipulation tasks between two arbitrary shapes. The control objective is to balance the object and drive the hand to a desired configuration. Planning and control of rolling on general curved shapes is also studied in [57].

The stabilization of a disk free to roll on an actuated disk (disk on disk system) is introduced in [94, 95]. The ball and beam is a benchmark system where a ball rolls on a one degree of freedom linear beam. It was extensively studied in the past years because of its peculiar feature: it fails to have a well defined relative degree, thus feedback linearization cannot be applied. The authors of [46] propose an approximate input-output linearization of this nonlinear systems. An output feedback controller is introduced in [115]. A flatness based approach with an exact feedforward linearization is introduced in [44]. In [83] a passivity-based control is applied to the problem of stabilization of some underactuated mechanical systems, like the ball and beam. The authors of [42] show

a technique for obtaining stable and robust oscillations for the ball and beam system, which consists of two steps. The first step aims at finding a control law such that the closed-loop of a reduced model of the dynamics is a second order Hamiltonian system which presents stable oscillations; in the second step, the controller is extended to the full system using backstepping. In [96] a control method for a redundant manipulator to balance the ball-beam system is showed. The force/torque sensor attached to the end-effector of the manipulator is used for estimating the ball position. Since it involves significant noise, a state feedback controller is employed along with an observer.

In the literature, the circular ball and beam has been proposed as variation of the ball and straight beam system where the lower disk (circular beam) has a decentralized center of mass. The stabilization of such a system is challenging, as compared to conventional ball and beam system, due to the presence of two unstable equilibrium points and the gyroscopic forces which should not be neglected in its dynamics. Different linear and nonlinear control approaches are proposed in the literature to deal with the balancing of this system [7, 8, 99]. This is a system originally presented in [8], where the Jordan form of its model is linearized near the unstable equilibrium to design a linear controller. A linear control approach is also used in the [7], where the limits of the beam actuator are taken into account. Interestingly, a geometric passivity-based control approach for this system is presented in [99]. Another complex nonprehensile planar rolling manipulation is the butterfly task, inspected by [25, 71, 113]. Nonprehensile rolling systems where the center of mass (CoM) of the object does not coincide with its geometric center are investigated in [50].

This Chapter provides an extension of the papers [32] and [64]. On one hand, a control framework based on standard feedback linearization for nonprehensile planar rolling manipulation is presented in [64]. The general model employed in [64] contains the arclength parametrization of the object/hand shapes. Given some assumptions on the shapes of both the object and the manipulator, a feedback linearization transforms the state-space system in a chain of integrators form. Linear control theory is then used to stabilize the closed-loop system. On the other hand, a passivity-based control for a single example of nonprehensile planar rolling system, the disk on disk, is presented in [32]. A particular kind of passivity-based control is proposed, related to interconnection and damping assignment, known as IDA-PBC. This approach exploits the energy and interconnection properties of the physical system taking advantage its nonlinearities. The IDA-PBC aims at finding a control law such that the closed-loop preserves the Hamiltonian structure, with a minimum of the potential energy at the desired

equilibrium, and a further damping injection to ensure asymptotic stability. The control law is obtained by matching the open and desired closed-loops, and processing that result in a set of partial differential equations. The IDA-PBC method is here applied to both *separable* Hamiltonian systems, i.e. with constant mass matrix, such as the disk on disk system, and *non-separable* Hamiltonian systems, i.e. with non constant mass matrix, such as the ball and beam and the circular ball and beam systems. The latter has relatively more complex dynamics with respect to the former, that is a commonly studied system.

Firstly, a general model for separable planar rolling systems is considered. The balancing of the disk on disk is used first as example to illustrate an application of the control approach. Subsequently, a general analytic expression of the desired potential energy is derived for the class of nonprehensile planar rolling systems, with constant mass matrix. This function is obviously a solution of the partial differential equations resulting from the matching equation. Removing the assumption of constant mass matrix, a constructive method to solve the matching equations for non-separable planar rolling systems is proposed with the aim to reduce the design complexity, while preserving the effectiveness of the IDA-PBC method. The methodology is inspired by the works [40, 83, 93]. The approach employs a target potential energy matching equation, depending on a parametrization of the desired closed-loop mass matrix, to simultaneously simplify the recognition of the desired mass matrix and select the desired energy function for the closed-loop system. In particular the straight ball and beam, and the circular ball and beam, are considered as further examples. The performances of the controllers are validated with numerical tests in the Matlab/Simulink environment.

2.2 Nonprehensile Planar Rolling Manipulation Dynamics

The dynamic model for nonprehensile planar rolling manipulation based on the works [64, 95], for symmetric[1] planar rolling, is reported in this Section, with an extension in case of asymmetric planar rolling manipulation, i.e. removing the assumption that the center of rotation of the hand corresponds to its geometric center. This class of rolling manipulation is represented as planar underactuated systems, with underactuation degree one.

[1]The term *symmetric* is here referred, with a slight abuse of language, to the correspondence between the center of rotation and the geometric center, without any reference to the symmetry of shapes.

2.2.1 Symmetric Planar Rolling

The dynamic model of the class of symmetric planar rolling manipulation is here derived.

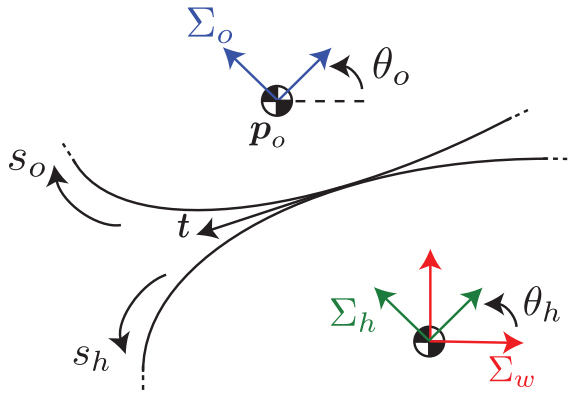


Figure 2.1: A general nonprehensile planar rolling manipulation system. In red the world fixed frame Σ_w. In green the hand frame Σ_h, while in blue the object frame Σ_o, placed at the respective centers of mass.

Referring to Fig. 2.1, let Σ_w be the inertia world fixed frame, while let Σ_h be the frame attached to the hand, and Σ_o the frame attached to the object. The last two frames are located at their respective centers of mass. Let $\theta_h \in \mathbb{R}$ be the angle of the hand in Σ_w, while $\boldsymbol{p}_o \in \mathbb{R}^2$ and $\theta_o \in \mathbb{R}$ are the position and the orientation, respectively, of Σ_o in Σ_w.

The shapes of both the object and the hand are represented through an arclength parametrization: $s_h \in \mathbb{R}$ and $s_o \in \mathbb{R}$ are the arclength parameters for the hand and the object, respectively. At least locally, the shapes should be of class $\mathcal{C}^2$. Locally, any point of the hand's shape is given by the chart $\boldsymbol{c}_h^h(s_h) = \begin{bmatrix} u_h(s_h) & v_h(s_h) \end{bmatrix}^T \in \mathbb{R}^2$, expressed with respect to Σ_h, while any point of the object's shape is given by $\boldsymbol{c}_o^o(s_o) = \begin{bmatrix} u_o(s_o) & v_o(s_o) \end{bmatrix}^T \in \mathbb{R}^2$, expressed with respect to Σ_o. Notice that s_h increases counterclockwise along the hand, while s_o increases clockwise along the object. With this choice, the pure rolling assumption is $\dot{s}_h = \dot{s}_o$. Without loss of generality, the frames Σ_w and Σ_h coincide at $\theta_h = 0$, the point $s_h = 0$ is at the intersection between the vertical (gravitational) axis of Σ_w and the hand's shape, i.e. $\boldsymbol{c}_h(0) = \begin{bmatrix} 0 & v_h(0) \end{bmatrix}^T$ in Σ_w, and thus $s_h = s_o$ at all times during rolling. Therefore, the contact location will be specified only by s_h throughout the remainder of the paper.

As first assumption, the hand and the object maintain pure rolling contact for all time. The arclength parametrization implies the property $\|\boldsymbol{c}_h^{h\prime}\| = 1$, with the symbol $'$ indicating the derivative with respect to the parameter s_h. The same holds for $\boldsymbol{c}_o^o(s_h)$. At the contact point $\boldsymbol{c}_h^h(s_h)$, the tangent vector to the shapes is expressed as $\boldsymbol{t}^h(s_h) = \boldsymbol{c}_h^{h\prime} \in \mathbb{R}^2$ forming an angle $\phi_h(s_h) = \text{atan2}(v_h'(s_h), u_h'(s_h))$ in Σ_h. The same tangent can be expressed with respect to Σ_o with an angle $\phi_o(s_h) = \text{atan2}(v_o'(s_h), u_o'(s_h))$. The signed curvatures of the shapes are defined as

$$\kappa_h(s_h) = \phi_h'(s_h) = u_h'(s_h)v_h''(s_h) - u_h''(s_h)v_h'(s_h), \tag{2.1a}$$
$$\kappa_o(s_h) = \phi_o'(s_h) = u_o'(s_h)v_o''(s_h) - u_o''(s_h)v_o'(s_h). \tag{2.1b}$$

The relative curvature at the contact point is given by

$$\kappa_r(s_h) = \kappa_h(s_h) - \kappa_o(s_h). \tag{2.2}$$

Notice that $\kappa_h(s_h) > 0$ and $\kappa_o(s_h) < 0$ denote convexity at the contact point for the hand and the object, respectively. Hence, $\kappa_r(s_h) > 0$ guarantees a single contact point at least locally [95]. The following constraint expresses the angle of the tangent $\boldsymbol{t}^h(s_h)$ with respect to Σ_w: $\theta_h + \phi_h(s_h) = \theta_o + \phi_o(s_h)$. Therefore, taking into account (2.2), the following relations hold

$$\theta_o = \theta_h + \phi_h(s_h) - \phi_o(s_h), \tag{2.3a}$$
$$\dot{\theta}_o = \dot{\theta}_h + \kappa_r(s_h)\dot{s}_h. \tag{2.3b}$$

The following constraint expresses instead the coincidence between the contact points on both the hand and the object

$$\boldsymbol{p}_h + \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^h(s_h) = \boldsymbol{p}_o + \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^o(s_h), \tag{2.4}$$

where $\boldsymbol{p}_h \in \mathbb{R}^2$ is the position of Σ_h in Σ_w, while $\boldsymbol{R}(\theta) \in SO(2)$ is the rotation matrix in the 2D space. Notice that the relation $\dot{\boldsymbol{R}}(\theta) = \boldsymbol{R}(\bar{\theta})\dot{\theta}$ holds with $\bar{\theta} = \theta + \frac{\pi}{2}$.

Another important assumption, for now, is that the hand can only rotate around its CoM. Therefore, without loss of generality, placing Σ_w at the hand's CoM and taking into account (2.4) yield

$$\boldsymbol{p}_o = \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^h(s_h) - \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^o(s_h), \tag{2.5}$$

and

$$\dot{\boldsymbol{p}}_o = \boldsymbol{\gamma}(\boldsymbol{q})\dot{\theta}_h + \boldsymbol{\eta}(\boldsymbol{q})\dot{s}_h = \begin{bmatrix}\boldsymbol{\gamma}(\boldsymbol{q}) & \boldsymbol{\eta}(\boldsymbol{q})\end{bmatrix}\dot{\boldsymbol{q}}, \tag{2.6}$$

with $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T = \begin{bmatrix} \theta_h & s_h \end{bmatrix}^T$, and

$$\boldsymbol{\gamma} = \boldsymbol{R}(\bar{\theta}_h)\boldsymbol{c}_h^h - \boldsymbol{R}(\bar{\theta}_o)\boldsymbol{c}_o^o, \tag{2.7a}$$

$$\boldsymbol{\eta} = \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^{h\prime} - \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^{o\prime} - \kappa_r \boldsymbol{R}(\bar{\theta}_o)\boldsymbol{c}_o^o, \tag{2.7b}$$

in which dependencies have been dropped, while (2.3b) is included and (2.3a) has to be plugged in.

Hence, considering the given assumption, the dynamic model is derived through the Euler-Lagrange formalism. The kinetic and potential energy for a nonprehensile planar rolling manipulation task are respectively given by

$$T(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\left(I_h \dot{\theta}_h^2 + m_o \dot{\boldsymbol{p}}_o^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{p}}_o(\boldsymbol{q}, \dot{\boldsymbol{q}}) + I_o \dot{\theta}_o^2(\boldsymbol{q})\right) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\mathbf{q}} \tag{2.8}$$

and

$$V(\boldsymbol{q}) = m_o g \begin{bmatrix} 0 & 1 \end{bmatrix}^T \boldsymbol{p}_o(\boldsymbol{q}), \tag{2.9}$$

where m_o is the object mass, I_h and I_o are the hand and object inertia, respectively, computed with respect to Σ_h and Σ_o, g is the gravity acceleration, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{2\times 2}$ is the symmetric and positive definite mass matrix whose elements are

$$b_{11}(\boldsymbol{q}) = I_h + I_o + m_o \boldsymbol{\gamma}^T(\boldsymbol{q})\boldsymbol{\gamma}(\boldsymbol{q}), \tag{2.10a}$$

$$b_{12}(\boldsymbol{q}) = b_{21}(\boldsymbol{q}) = I_o \kappa_r(s_h) + m_o \boldsymbol{\gamma}(\boldsymbol{q})^T \boldsymbol{\eta}(\boldsymbol{q}), \tag{2.10b}$$

$$b_{22}(\boldsymbol{q}) = I_o \kappa_r^2(s_h) + m_o \boldsymbol{\eta}(\boldsymbol{q})^T \boldsymbol{\eta}(\boldsymbol{q}). \tag{2.10c}$$

Consequently, the Lagrangian function is given by

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\mathbf{q}} - V(\boldsymbol{q}). \tag{2.11}$$

By computing the Lagrange equations [110], the dynamic model can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \nabla V(\boldsymbol{q}) = \boldsymbol{G}(\boldsymbol{q})\tau, \tag{2.12}$$

where τ is the actuating torque around the CoM of the hand,

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} g_1(\boldsymbol{q}) & g_2(\boldsymbol{q}) \end{bmatrix}^T = \nabla V(\boldsymbol{q})$$

and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{2\times 2}$ is a suitable matrix whose generic element is given by

$$c_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\sum_{k=1}^{2}\left(\frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k} + \frac{\partial b_{ik}(\boldsymbol{q})}{\partial q_j} + \frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i}\right)\dot{q}_k, \tag{2.13}$$

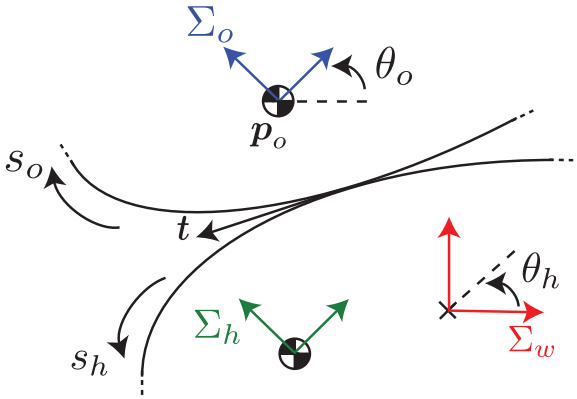


Figure 2.2: The general nonprehensile planar rolling manipulation system with the center of rotation of the hand (indicated by the $\times$ symbol) not corresponding to its geometric center. In red the world fixed frame Σ_w. In green the hand frame Σ_h, while in blue the object frame Σ_o, placed at the respective centers of mass.

with $i, j = 1, 2$. The dynamic model can be written in the following extended form, dropping the dependencies,

$$b_{11}\ddot{\theta}_h + b_{12}\ddot{s}_h + c_{11}\dot{\theta}_h + c_{12}\dot{s}_h + g_1 = \tau, \tag{2.14a}$$
$$b_{12}\ddot{\theta}_h + b_{22}\ddot{s}_h + c_{21}\dot{\theta}_h + c_{22}\dot{s}_h + g_2 = 0, \tag{2.14b}$$

with

$$g_1 = m_o g \left(-v_h s_{\theta_h} - v_o \frac{\partial c_{\theta_o}}{\partial \theta_h} + u_h c_{\theta_h} - u_o \frac{\partial s_{\theta_o}}{\partial \theta_h} \right), \tag{2.15a}$$
$$g_2 = m_o g (v'_h c_{\theta_h} - v'_o c_{\theta_o} - v_o c'_{\theta_o} + u'_h s_{\theta_h} - u'_o s_{\theta_o} - u_o s'_{\theta_o}), \tag{2.15b}$$

in which (2.3a) has to be plugged in, the elements of $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are omitted for brevity, and c_θ and s_θ are used instead of $\cos\theta$ and $\sin\theta$, respectively.

2.2.2 Asymmetric Planar Rolling

The dynamic model of the asymmetric version of a nonprehensile planar rolling system is now derived. In this asymmetric version of the system, the center of rotation of the hand does not correspond to its geometric center, therefore the assumption that the hand can only rotate around its CoM has been removed. Referring to Fig. 2.2, let Σ_w be the inertia world fixed frame, which is without

loss of generality attached at the holder where the hand is actuated (i.e., the center of rotation of the hand). Also, let Σ_h be the frame attached to the CoM of the hand, while Σ_o is the frame attached to the CoM of the object. Let $\theta_h \in \mathbb{R}$ be the angle of the hand in Σ_w, while $\boldsymbol{p}_o \in \mathbb{R}^2$ and $\theta_o \in \mathbb{R}$ are the position and the orientation, respectively, of Σ_o in Σ_w. The position of the CoM of the hand in Σ_w is denoted by $\boldsymbol{p}_h(\theta_h) = \begin{bmatrix} u_w(\theta_h) & v_w(\theta_h) \end{bmatrix}^T \in \mathbb{R}^2$. For the asymmetric version of the system, the coincidence between the contact points on both the hand and the object is expressed by

$$\boldsymbol{p}_h(\theta_h) + \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^h(s_h) = \boldsymbol{p}_o + \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^o(s_h), \tag{2.16}$$

yielding to

$$\boldsymbol{p}_o = \boldsymbol{p}_h(\theta_h) + \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^h(s_h) - \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^o(s_h), \tag{2.17}$$

and

$$\dot{\boldsymbol{p}}_o = \boldsymbol{\gamma}(\boldsymbol{q})\dot{\theta}_h + \boldsymbol{\eta}(\boldsymbol{q})\dot{s}_h = \begin{bmatrix} \boldsymbol{\gamma}(\boldsymbol{q}) & \boldsymbol{\eta}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}}, \tag{2.18}$$

with

$$\boldsymbol{\gamma} = \boldsymbol{p}_h^{\backprime} + \boldsymbol{R}(\bar{\theta}_h)\boldsymbol{c}_h^h - \boldsymbol{R}(\bar{\theta}_o)\boldsymbol{c}_o^o, \tag{2.19a}$$

$$\boldsymbol{\eta} = \boldsymbol{R}(\theta_h)\boldsymbol{c}_h^{h\prime} - \boldsymbol{R}(\theta_o)\boldsymbol{c}_o^{o\prime} - \kappa_r \boldsymbol{R}(\bar{\theta}_o)\boldsymbol{c}_o^o, \tag{2.19b}$$

in which dependencies have been dropped, while (2.3b) is included and (2.3a) has to be plugged in. The symbol $\backprime$ indicates the derivative with respect to θ_h. For this class of system, where the only assumption is the preservation of the pure rolling contact for all time between the hand and the object, the kinetic energy for an asymmetric planar rolling system becomes

$$T(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\left(I_h \dot{\theta}_h^2 + m_h \dot{\boldsymbol{p}}_h^{\mathrm{T}}(\theta_h)\dot{\boldsymbol{p}}_h(\theta_h) + m_o \dot{\boldsymbol{p}}_o^{\mathrm{T}}\dot{\boldsymbol{p}}_o + I_o \dot{\theta}_o^2 \right) = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{B}(\boldsymbol{q})\dot{\mathbf{q}}, \tag{2.20}$$

with the following elements for the mass matrix

$$b_{11}(\boldsymbol{q}) = I_h + I_o + m_h \boldsymbol{p}_h^{\backprime T}\boldsymbol{p}_h^{\backprime} + m_o \boldsymbol{\gamma}^T(\boldsymbol{q})\boldsymbol{\gamma}(\boldsymbol{q}), \tag{2.21a}$$

$$b_{12}(\boldsymbol{q}) = b_{21}(\boldsymbol{q}) = I_o \kappa_r(s_h) + m_o \boldsymbol{\gamma}(\boldsymbol{q})^T \boldsymbol{\eta}(\boldsymbol{q}), \tag{2.21b}$$

$$b_{22}(\boldsymbol{q}) = I_o \kappa_r^2(s_h) + m_o \boldsymbol{\eta}(\boldsymbol{q})^T \boldsymbol{\eta}(\boldsymbol{q}). \tag{2.21c}$$

The potential energy becomes

$$V(\boldsymbol{q}) = g \begin{bmatrix} 0 & 1 \end{bmatrix} (m_o \boldsymbol{p}_o(\boldsymbol{q}) + m_h \boldsymbol{p}_h(\boldsymbol{q})). \tag{2.22}$$

By computing the Lagrange equations, the dynamic model for the asymmetric version of the planar rolling manipulation can be written in the following form

$$b_{11}\ddot{\theta}_h + b_{12}\ddot{s}_h + c_{11}\dot{\theta}_h + c_{12}\dot{s}_h + g_1 = \tau_h, \tag{2.23a}$$
$$b_{12}\ddot{\theta}_h + b_{22}\ddot{s}_h + c_{21}\dot{\theta}_h + c_{22}\dot{s}_h + g_2 = 0, \tag{2.23b}$$

with

$$g_1 = g(m_o c_{\theta_h} u_h - m_o s_{\theta_h} v_h + (m_h + m_o)v_w^{\backprime} + m_o(s_{\theta_o} v_o - c_{\theta_o} u_o)\theta_o^{\backprime}), \tag{2.24a}$$
$$g_2 = m_o g(s_{\theta_h} u_h' + c_{\theta_h} v_h' - c_{\theta_o}(v_o' + u_o\theta_o') + s_{\theta_o}(-u_o' + v_o\theta_o')), \tag{2.24b}$$

in which (2.3a) has to be plugged in, the elements of $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are omitted for brevity.

2.3 IDA Passivity-based Control

The Port-Hamiltonian framework allows modeling of mechanical systems, preserving physical phenomenon information. The dynamical system is intended as an energy-transformation device and the action of a controller is understood in energy terms.

The dynamics of the nonprehensile planar rolling system in generalized coordinates considered in (2.12) is described using the Euler-Lagrange formalism. In classical mechanics, the conjugate generalized momentum, $\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^T \in \mathbb{R}^2$, is defined as $\boldsymbol{p} = \nabla_{\dot{q}}\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$. Using the momentum and the generalized coordinate vector, the Euler-Lagrange equations can be transformed, using the transformation of Legendre, into a set of first order differential equations known as Hamiltonian canonical equations of motion

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{2\times2} & \boldsymbol{I}_{2\times2} \\ -\boldsymbol{I}_{2\times2} & \boldsymbol{0}_{2\times2} \end{bmatrix} \nabla\mathcal{H} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{G} \end{bmatrix} \tau, \tag{2.25}$$

where $\boldsymbol{I}_{2\times2}, \boldsymbol{0}_{2\times2} \in \mathbb{R}^{2\times2}$ are respectively the identity and the zero matrix, and $\boldsymbol{0} \in \mathbb{R}^2$ is the zero vector. The Hamiltonian function, $\mathcal{H}$, represents the total energy of the system, and is given by

$$\mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = \dot{\boldsymbol{q}}^T\boldsymbol{p} - \mathcal{L}, \tag{2.26}$$

where $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{p})$. The Hamiltonian equations (2.25) provide a particular state-space representation of the system dynamics equivalent to the Euler-Lagrange model (2.12). Besides, Hamiltonian models are more general in the sense that

there are systems that admit Hamiltonian but not Lagrangian representations [60].

Within the IDA-PBC, stabilization is achieved assigning to the closed-loop the target dynamics

$$\begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{2\times 2} & \boldsymbol{M}^{-1}\boldsymbol{M}_d\\ -\boldsymbol{M}_d\boldsymbol{M}^{-1} & \boldsymbol{J}_2\end{bmatrix}\nabla\mathcal{H}_d, \tag{2.27}$$

where $\boldsymbol{J}_2 = -\boldsymbol{J}_2^T$ is the desired interconnection matrix, depending on $\boldsymbol{q}$ and $\boldsymbol{p}$, and $\mathcal{H}_d$ is the desired total energy function

$$\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p} + V_d(\boldsymbol{q}), \tag{2.28}$$

with desired mass matrix $\boldsymbol{M}_d(\boldsymbol{q}) = \boldsymbol{M}_d(\boldsymbol{q})^T > 0$, and desired potential energy function $V_d(\boldsymbol{q})$. The condition under which $(\boldsymbol{q}^*, 0)$ is a stable equilibrium point of the closed-loop (2.27) with Lyapunov function $\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p})$ is

$$\boldsymbol{q}^* = argmin\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p}) = argmin V_d(\boldsymbol{q}), \tag{2.29}$$

that corresponds to the respectively necessary and sufficient conditions on $V_d(\boldsymbol{q}^*)$

- $\nabla V_d(\boldsymbol{q}^*) = 0$,
- $\nabla^2 V_d(\boldsymbol{q}^*) > 0$.

The stabilization of the desired equilibrium is achieved by identifying the class of Hamiltonian systems that can be obtained via feedback. The conditions under which this feedback law exists are called *matching conditions*, i.e. matching the original dynamic system (2.25) and the target dynamic system (2.27)

$$\begin{bmatrix}\mathbf{0}_{2\times 2} & \boldsymbol{I}_{2\times 2}\\ -\boldsymbol{I}_{2\times 2} & \mathbf{0}_{2\times 2}\end{bmatrix}\nabla\mathcal{H}(\boldsymbol{p},\boldsymbol{q}) + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{G}\end{bmatrix}\tau = \begin{bmatrix}\mathbf{0}_{2\times 2} & \boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{M}_d(\boldsymbol{q})\\ -\boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q}) & \boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})\end{bmatrix}\nabla\mathcal{H}_d(\boldsymbol{p},\boldsymbol{q}), \tag{2.30}$$

The matching conditions correspond to the following set of nonlinear partial differential equations (PDEs)

$$\boldsymbol{G}^\perp\left(\nabla_q\mathcal{H}(\boldsymbol{q},\boldsymbol{p}) - \boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_q\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p}) + \boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}\right) = 0, \tag{2.31}$$

with the full rank left annihilator $\boldsymbol{G}^\perp = \begin{bmatrix}0 & 1\end{bmatrix}$ of $\boldsymbol{G}$, for the class of mechanical systems under study. The PDEs (2.31) can be separated in two sets of PDEs

$$\boldsymbol{G}^\perp\left(\nabla_q(\boldsymbol{p}^T\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{p}) - \boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_q(\boldsymbol{p}^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}) + 2\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}\right) = 0, \tag{2.32}$$

$$\boldsymbol{G}^\perp\left(\nabla V(\boldsymbol{q}) - \boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla V_d(\boldsymbol{q})\right) = 0, \tag{2.33}$$

where (2.32) are named *kinetic energy PDEs*, and (2.33) are named *potential energy PDEs*.

If the sets of PDEs (2.32)-(2.33) are solved, and the conditions on $\boldsymbol{M}_d(\boldsymbol{q})$, $V_d(\boldsymbol{q})$, $\boldsymbol{J}_2(\boldsymbol{q})$ are respected, the energy shaping control τ_{es} results

$$\tau_{es} = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T(\nabla_q\mathcal{H}(\boldsymbol{q},\boldsymbol{p}) - \boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_q\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p}) + \boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}). \tag{2.34}$$

By applying (2.34) to the Hamiltonian dynamics (2.25) the closed-loop target dynamics (2.27) is obtained. A damping term, τ_{di}, aimed at achieving asymptotic stability, is then injected through

$$\tau_{di} = -k_v\boldsymbol{G}^T\nabla_p\mathcal{H}_d(\boldsymbol{q},\boldsymbol{p}), \tag{2.35}$$

where k_v is a positive scalar gain. The damping injection (2.35) and the energy shaping control (2.34) are then assembled to generate the IDA-PBC

$$\tau = \tau_{es} + \tau_{di}. \tag{2.36}$$

Therefore, the closed-loop target dynamics is shaped as

$$\begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}_{2\times2} & \boldsymbol{M}^{-1}\boldsymbol{M}_d\\ -\boldsymbol{M}_d\boldsymbol{M}^{-1} & \boldsymbol{J}_2 - \boldsymbol{R}_d\end{bmatrix}\nabla\mathcal{H}_d, \tag{2.37}$$

in which dependencies have been dropped, and $\boldsymbol{R}_d = k_v\boldsymbol{G}\boldsymbol{G}^T > 0$ is the dissipation matrix [82, 83].

2.4 Proposed Approach for Constant Mass Matrix

The class of separable nonprehensile planar rolling manipulation, such as for example the disk on disk, is initially considered. This can be modeled through the Hamiltonian canonical equations, equivalent to the dynamics (2.12), with Hamiltonian function given by (2.26), and with the constant mass matrix

$$\boldsymbol{M} = \begin{bmatrix}b_{11} & b_{12}\\ b_{12} & b_{22}\end{bmatrix} = \boldsymbol{M}^T > 0. \tag{2.38}$$

In this approach, it is assumed to parametrize the constant desired inertia matrix satisfying the conditions of positive definiteness and symmetry, as follows

$$\boldsymbol{M}_d = \begin{bmatrix}a_{11} & a_{12}\\ a_{12} & a_{22}\end{bmatrix} = \boldsymbol{M}_d^T > 0. \tag{2.39}$$

Now, the desired potential energy function $V_d(\boldsymbol{q})$ must be found. The PDEs (2.32) and (2.33) are reduced to

$$\boldsymbol{G}^{\perp}\left(2\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{M}_d^{-1}\boldsymbol{p}\right)=0, \tag{2.40}$$

$$\boldsymbol{G}^{\perp}\left(\nabla V(\boldsymbol{q})-\boldsymbol{M}_d\boldsymbol{M}^{-1}\nabla V_d(\boldsymbol{q})\right)=0. \tag{2.41}$$

The first PDE (2.40) can be solved choosing the trivial zero matrix $\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})=\boldsymbol{0}_{2\times 2}$, while the second PDE (2.41) provides a candidate desired potential energy function. Replacing the desired mass matrix (2.39), the mass matrix (2.38), the $\boldsymbol{G}^{\perp}$ vector, and the gradient of the potential energy $\nabla V_d(\boldsymbol{q})$, the potential energy target equation (2.41) to solve with respect to $V_d(\boldsymbol{q})$ becomes

$$\nabla_{\theta_h}V_d(\boldsymbol{q})(a_{12}b_{22}-a_{22}b_{12})+\nabla_{s_h}V_d(\boldsymbol{q})(a_{22}b_{11}-a_{12}b_{12})+g_2(\boldsymbol{q})(b_{12}^2-b_{11}b_{22})=0, \tag{2.42}$$

that, for the symmetric nonprehensile planar rolling manipulation, corresponds to

$$\begin{aligned}&c_1\nabla_{\theta_h}V_d(\boldsymbol{q})+c_2\nabla_{s_h}V_d(\boldsymbol{q})+m_o g\left(-u_o'(s_h)\sin(k_r s_h+\theta_h)-k_r u_o(s_h)\cos(k_r s_h+\theta_h)\right)\\&-m_o g\left(v_o'(s_h)\cos(k_r s_h+\theta_h)+k_r v_o(s_h)\sin(k_r s_h+\theta_h)+\sin(\theta_h)u_h'(s_h)\right)\\&-m_o g\left(\cos(\theta_h)v_h'(s_h)\right)=0,\end{aligned} \tag{2.43}$$

where

$$c_1=\frac{a_{12}b_{22}-a_{22}b_{12}}{b_{12}^2-b_{11}b_{22}}, \tag{2.44}$$

and

$$c_2=\frac{a_{22}b_{11}-a_{12}b_{12}}{b_{12}^2-b_{11}b_{22}}, \tag{2.45}$$

and the expression of $g_2(\boldsymbol{q})$ is replaced from (2.15b).

The solution $V_d(\boldsymbol{q})$ of the PDE (2.43), obtained thanks to the symbolic computation program Mathematica, is

$$\begin{aligned}&\int_1^{\theta_h}\frac{1}{c_1}\left(gm_o\sin(k_r\psi(\tau,\boldsymbol{q})+\tau)u_o'(\psi(\tau,\boldsymbol{q}))+gk_rm_ou_o(\psi(\tau,\boldsymbol{q}))\cos(k_r\psi(\tau,\boldsymbol{q})+\tau)\right)+\\&\frac{1}{c_1}\left(gm_o\cos(k_r\psi(\tau,\boldsymbol{q})+\tau)v_o'(\psi(\tau,\boldsymbol{q}))-gk_rm_ov_o(\psi(\tau,\boldsymbol{q}))\sin(k_r\psi(\tau,\boldsymbol{q})+\tau)\right)-\\&\frac{1}{c_1}\left(gm_o\sin(\tau)u_h'(\psi(\tau,\boldsymbol{q}))-gm_o\cos(\tau)v_h'(\psi(\tau,\boldsymbol{q}))\right)d\tau+f\left(s_h-\frac{c_2\theta_h}{c_1}\right),\end{aligned} \tag{2.46}$$

where $\psi(\tau,\boldsymbol{q})=s_h+\frac{c_2(\tau-\theta_h)}{c_1}$ and $f(\cdot)$ is a generic function of its argument.

This result can be applied to find the desired potential energy for a symmetric planar rolling manipulation system modeled by a separable dynamics. The

general closed form solution for V_d can be used taking into account the particular shapes of object and hand through a specific arclength parametrization. Therefore, there is not need to solve again the matching equations for each example that verifies the considered assumptions. In case of non-separable dynamics, the general potential energy target equation (2.43), which depends on the shapes of the contacting objects, becomes impossible to solve.

2.4.1 Disk on Disk Application

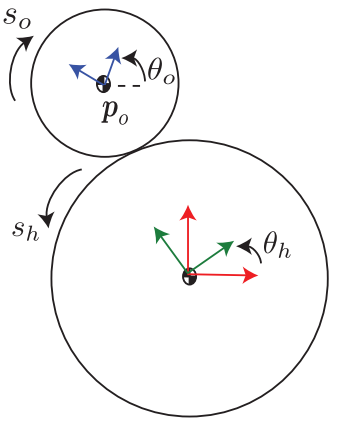


Figure 2.3: A representation of the disk on disk system. In red the world fixed frame Σ_w. In green the hand frame Σ_h, while in blue the object frame Σ_o, placed at the respective centers of mass.

As example, the model presented in [64] for the balancing of a disk free to roll on an actuated disk is considered. Referring to Fig. 2.3, the shape of the hand, i.e. the actuated disk, is parametrized by the chart

$$\boldsymbol{c}_h^h(s_h) = \rho_h \begin{bmatrix} -\sin\frac{s_h}{\rho_h} & \cos\frac{s_h}{\rho_h} \end{bmatrix}^T, \tag{2.47}$$

with $\rho_h \in \mathbb{R}^+$ the radius of the hand. The shape of the object is parametrized by the chart

$$\boldsymbol{c}_o^o(s_h) = -\rho_o \begin{bmatrix} \sin\frac{s_h}{\rho_o} & \cos\frac{s_h}{\rho_o} \end{bmatrix}^T, \tag{2.48}$$

with $\rho_o \in \mathbb{R}^+$ the radius of the ball. Considering (2.2), the relative curvature is given by

$$\kappa_r = \frac{\rho_h + \rho_o}{\rho_h \rho_o}. \tag{2.49}$$

Table 2.1: Elements of the dynamic model for the disk on disk example.

b_{11}	$I_h + I_o + m_o(\rho_h + \rho_o)^2$
b_{12}, b_{21}	$I_o\kappa_r + m_o\frac{(\rho_h+\rho_o)^2}{\rho_h}$
b_{22}	$I_o\kappa_r^2 + m_o\rho_o^2\kappa_r^2$
g_1	$-m_o g(\rho_h + \rho_o)\sin\left(\theta_h + \frac{s_h}{\rho_h}\right)$
g_2	$-m_o g\rho_o\kappa_r \sin\left(\theta_h + \frac{s_h}{\rho_h}\right)$

The upper disk angular velocity is given by $\dot{\theta}_o = \dot{\theta}_h + \kappa_r\dot{s}_h$. The vectors $\boldsymbol{\gamma}(\boldsymbol{q})$ and $\boldsymbol{\eta}(\boldsymbol{q})$ are computed like in (2.7):

$$\boldsymbol{\gamma}(\boldsymbol{q}) = -(\rho_h + \rho_o)\left[\cos\left(\theta_h + \frac{s_h}{\rho_h}\right) \quad \sin\left(\theta_h + \frac{s_h}{\rho_h}\right)\right]^T, \tag{2.50}$$

and

$$\boldsymbol{\eta}(\boldsymbol{q}) = -\rho_o\kappa_r\left[\cos\left(\theta_h + \frac{s_h}{\rho_h}\right) \quad \sin\left(\theta_h + \frac{s_h}{\rho_h}\right)\right]^T. \tag{2.51}$$

The resulting elements of the dynamics (2.14) are detailed in Table 2.1, with $c_{11} = c_{12} = c_{21} = c_{22} = 0$, as derived from (2.13). Notice that the quantity $\theta_h + \frac{s_h}{\rho_h}$ is the angle of the object's CoM with respect to the vertical axis of Σ_w.

For this system, replacing the charts representing the shapes of hand/object (2.47) (2.48) and the relative curvature (2.49), and integrating with Mathematica, the desired potential energy function (2.46) becomes

$$V_d(\boldsymbol{q}) = -\frac{g m_o(\rho_o + \rho_h)\left(\cos\left(\theta_h + \frac{s_h}{\rho_h}\right) - \cos\left(\frac{c_1(s_h+\rho_h)+c_2(-\theta_h)+c_2}{c_1\rho_h}\right)\right)}{c_1\rho_h + c_2} + f\left(s_h - \frac{c_2\theta_h}{c_1}\right), \tag{2.52}$$

where c_1, c_2 derive from (2.44) and (2.45). This function $V_d(\boldsymbol{q})$ obviously satisfies the potential energy matching equation for the disk on disk system. With an appropriate choice of the function $f(\cdot)$, the potential energy (2.52) can be related to the one proposed in [32], where a possible constant positive definite mass matrix $\boldsymbol{M}_d$ is proposed, together with the conditions on the gains such that the $V_d(\boldsymbol{q})$ has a minimum at the equilibrium.

2.4.2 Approximated Ball and Beam Application

As further example, the traditional ball and beam benchmark system is considered. Referring to Fig. 2.4, the beam can rotate around its CoM while the ball can only roll along the beam. The shape of the hand, i.e. the beam, is parametrized

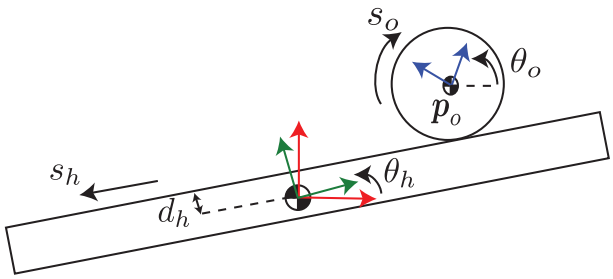


Figure 2.4: A representation of the ball and beam system. In red the world fixed frame Σ_w. In green the hand frame Σ_h, while in blue the object frame Σ_o, placed at the respective centers of mass.

through the chart

$$\boldsymbol{c}_h^h(s_h) = \begin{bmatrix} -s_h & d_h \end{bmatrix}^T, \tag{2.53}$$

with $d_h \in \mathbb{R}^+$ a fixed distance between the beam's CoM and its surface where the ball rolls. The shape of the object, i.e. the ball, is parametrized by the chart

$$\boldsymbol{c}_o^o(s_h) = -\rho_o \begin{bmatrix} \sin\frac{s_h}{\rho_o} & \cos\frac{s_h}{\rho_o} \end{bmatrix}^T, \tag{2.54}$$

with $\rho_o \in \mathbb{R}^+$ the radius of the ball. Considering (2.2), the signed curvatures of the beam and the ball are $\kappa_h = 0$ and $\kappa_o = -1/\rho_o$, respectively. The relative curvature is thus given by

$$\kappa_r = 1/\rho_o. \tag{2.55}$$

The ball's angular velocity is instead given by (2.3b) as $\dot{\theta}_o = \dot{\theta}_h + \dfrac{\dot{s}_h}{\rho_o}$. In order to compute the mass matrix of the system, the vectors $\boldsymbol{\gamma}(\boldsymbol{q})$ and $\boldsymbol{\eta}(\boldsymbol{q})$ in (2.7) are

$$\boldsymbol{\gamma}(\boldsymbol{q}) = \begin{bmatrix} -(\rho_o + d_h)c_{\theta_h} + s_h s_{\theta_h} & -(\rho_o + d_h)s_{\theta_h} - s_h c_{\theta_h} \end{bmatrix}^T, \tag{2.56}$$

and

$$\boldsymbol{\eta}(\boldsymbol{q}) = -\begin{bmatrix} c_{\theta_h} & s_{\theta_h} \end{bmatrix}^T. \tag{2.57}$$

Therefore, the resulting elements of the dynamics (2.14) are reported in Table 2.2, where $c_{b1} = I_h + I_o + m_o d_h^2 + 2m_o d_h \rho_o + m_o \rho_o^2$ and $c_{b2} = m_o$. The Coriolis elements $c_{11}, c_{12}, c_{21}, c_{22}$ can be derived from (2.13).

The mass matrix is clearly not constant for the ball and beam, since $b_{11} = b_{11}(s_h)$. This model is well known in the literature, nevertheless, in many cases it is possible to approximate $b_{11}(s_h)$ neglecting the square of s_h. This is true for small velocities of the beam, small masses of the ball and not so long beam [47].

Table 2.2: Elements of the dynamic model for the ball and beam example.

b_{11}	$c_{b1} + c_{b2} s_h^2$
b_{12}, b_{21}	$\frac{I_o}{\rho_o} + m_o d_h + m_o \rho_o$
b_{22}	$\frac{I_o}{\rho_o^2} + m_o$
g_1	$-m_o g((d_h + \rho_o)\sin\theta_h + s_h \cos\theta_h)$
g_2	$-m_o g \sin\theta_h$

Hence, by putting $b_{11} = c_{b1}$, only for control purposes, it is possible to consider the ball and beam as a separable system. The same approximation is used in the work [64]. For this system, replacing the charts representing the shapes of hand/object (2.53) (2.54) and the relative curvature (2.55), and integrating with Mathematica, the desired potential energy function (2.46) becomes

$$V_d(\boldsymbol{q}) = -\frac{g m_o \cos(\theta_h)}{c_1} + f\left(s_h - \frac{c_2 \theta_h}{c_1}\right), \tag{2.58}$$

where c_1, c_2 derive from (2.44) and (2.45). This function $V_d(\boldsymbol{q})$ obviously satisfies the potential energy matching equation for the approximated ball and beam system. The constant parameters a_{11}, a_{12}, a_{22} of desired mass matrix $\boldsymbol{M}_d$ should then be properly tuned. In the following, the complete model of the ball and beam system is employed in order to apply a more general derivation of the control law.

2.5 Proposed Approach for Non-constant Mass Matrix

In this Section the objective is to extend the study to non-separable nonprehensile planar rolling systems. A systematic approach to solve the matching equations is proposed employing a target potential energy matching equation, which depends on a parametrization of the desired closed-loop mass matrix, to simultaneously simplify the recognition of the desired mass matrix and select the desired potential energy function for the closed-loop system. This method simplifies the solution of the PDEs imposed by the matching conditions. The authors of [40] propose to transform the PDEs derived from the kinetic energy matching condition in nonlinear ODEs through a parametrization of the desired closed-loop mass matrix. With the similar purpose to simplify the solution of the PDEs, a different parametrization of the closed-loop mass matrix is designed to simplify the solution of the potential energy PDEs. On one hand, the proposed expression

of the desired closed-loop mass matrix provides directly a solution of the potential energy PDEs. On the other hand, the kinetic energy matching condition is satisfied exploiting the technique proposed in [93].

This Section is focused on the class of nonprehensile planar rolling systems that can be modeled through the Hamiltonian canonical equations, equivalent to the dynamics (2.12), with the Hamiltonian function (2.26), with mass matrix

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} b_{11}(\boldsymbol{q}) & b_{12}(\boldsymbol{q}) \\ b_{12}(\boldsymbol{q}) & b_{22}(\boldsymbol{q}) \end{bmatrix} = \boldsymbol{M}(\boldsymbol{q})^T > 0, \tag{2.59}$$

and $\boldsymbol{G}^{\perp} = \boldsymbol{e}_2^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$. The first step towards the solution of the matching conditions is related to the potential energy PDE and to the conditions of symmetry and positive definiteness of the desired closed-loop mass matrix. In order to satisfy the potential energy matching equation, the desired inertia matrix is parametrized as follows

$$\boldsymbol{M}_d(\boldsymbol{q}) = \Delta \begin{bmatrix} a_{11}(\boldsymbol{q}) & a_{12}(\boldsymbol{q}) \\ a_{12}(\boldsymbol{q}) & a_{22}(\boldsymbol{q}) \end{bmatrix} = \boldsymbol{M}_d(\boldsymbol{q})^T \tag{2.60}$$

where $\Delta = b_{11}(\boldsymbol{q})b_{22}(\boldsymbol{q}) - b_{12}^2(\boldsymbol{q}) > 0$ is the determinant of $\boldsymbol{M}(\boldsymbol{q})$. Under this assumption, the potential energy matching equation (2.33) becomes

$$\boldsymbol{e}_2^T \left(\begin{bmatrix} g_1(\boldsymbol{q}) \\ g_2(\boldsymbol{q}) \end{bmatrix} - \boldsymbol{\Gamma} \begin{bmatrix} \nabla_{\theta_h} V_d(\boldsymbol{q}) \\ \nabla_{s_h} V_d(\boldsymbol{q}) \end{bmatrix} \right) = 0, \tag{2.61}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} a_{11}(\boldsymbol{q})b_{22}(\boldsymbol{q}) - a_{12}(\boldsymbol{q})b_{12}(\boldsymbol{q}) & a_{12}(\boldsymbol{q})b_{11}(\boldsymbol{q}) - a_{11}(\boldsymbol{q})b_{12}(\boldsymbol{q}) \\ a_{12}(\boldsymbol{q})b_{22}(\boldsymbol{q}) - a_{22}(\boldsymbol{q})b_{12}(\boldsymbol{q}) & a_{22}(\boldsymbol{q})b_{11}(\boldsymbol{q}) - a_{12}(\boldsymbol{q})b_{12}(\boldsymbol{q}) \end{bmatrix}, \tag{2.62}$$

and then

$$g_2(\boldsymbol{q}) - (a_{12}(\boldsymbol{q})b_{22}(\boldsymbol{q}) - a_{22}(\boldsymbol{q})b_{12}(\boldsymbol{q}))\nabla_{\theta_h} V_d(\boldsymbol{q}) - (a_{22}(\boldsymbol{q})b_{11}(\boldsymbol{q}) - a_{12}(\boldsymbol{q})b_{12}(\boldsymbol{q}))\nabla_{s_h} V_d(\boldsymbol{q}) = 0 \tag{2.63}$$

The two functions, $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$, are supposed to group the terms that multiply respectively $\nabla_{\theta_h} V_d(\boldsymbol{q})$ and $\nabla_{s_h} V_d(\boldsymbol{q})$ in (2.63). Hereafter, $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ can be selected such that the following target potential energy matching equation has always a solution $V_d(\boldsymbol{q})$ and a minimum in the equilibrium

$$g_2(\boldsymbol{q}) + \alpha(\boldsymbol{q})\nabla_{\theta_h} V_d(\boldsymbol{q}) + \beta(\boldsymbol{q})\nabla_{s_h} V_d(\boldsymbol{q}) = 0. \tag{2.64}$$

After that the functions $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ are fixed, the functions $a_{12}(\boldsymbol{q})$ and $a_{22}(\boldsymbol{q})$ of the desired mass matrix $\boldsymbol{M}_d(\boldsymbol{q})$ can be retrieved imposing the following algebraic system of equations

$$\begin{aligned} a_{22}(\boldsymbol{q})b_{12}(\boldsymbol{q}) - a_{12}(\boldsymbol{q})b_{22}(\boldsymbol{q}) &= \alpha(\boldsymbol{q}), \\ a_{12}(\boldsymbol{q})b_{12}(\boldsymbol{q}) - a_{22}(\boldsymbol{q})b_{11}(\boldsymbol{q}) &= \beta(\boldsymbol{q}). \end{aligned} \tag{2.65}$$

Consequently, grouping the determinant of the open-loop mass matrix, the solution results

$$\begin{aligned} a_{12}(\boldsymbol{q}) &= -\frac{\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q})}{\Delta}, \\ a_{22}(\boldsymbol{q}) &= -\frac{\alpha(\boldsymbol{q})b_{12}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{22}(\boldsymbol{q})}{\Delta}. \end{aligned} \tag{2.66}$$

Therefore, the proposed desired closed-loop mass matrix is given by

$$\boldsymbol{M}_d(\boldsymbol{q}) = \begin{bmatrix} \Delta a_{11}(\boldsymbol{q}) & -(\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q})) \\ -(\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q})) & -(\alpha(\boldsymbol{q})b_{12}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{22}(\boldsymbol{q})) \end{bmatrix}, \tag{2.67}$$

which is structurally symmetric. Moreover, the positive definiteness of the desired closed-loop mass matrix corresponds to the following conditions in $a_{11}(\boldsymbol{q})$:

- $a_{11}(\boldsymbol{q}) > 0$;
- $a_{11}(\boldsymbol{q})a_{22}(\boldsymbol{q}) - a_{12}^2(\boldsymbol{q}) > 0$.

Selecting the element $a_{11}(\boldsymbol{q})$ of the desired mass matrix as follows

$$a_{11}(\boldsymbol{q}) = \frac{k_a a_{12}^2(\boldsymbol{q})}{a_{22}(\boldsymbol{q})} > 0, \tag{2.68}$$

where $k_a \in \mathbb{R}$ is a constant parameter, both conditions on $a_{11}(\boldsymbol{q})$ are respected if the choice of the functions $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ is such that $a_{22}(\boldsymbol{q}) > 0$, i.e.

$$(\alpha(\boldsymbol{q})b_{12}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{22}(\boldsymbol{q})) < 0. \tag{2.69}$$

In fact, in this case, the determinant of $\boldsymbol{M}_d(\boldsymbol{q})$ is always positive

$$a_{11}(\boldsymbol{q}) > \frac{a_{12}^2(\boldsymbol{q})}{a_{22}(\boldsymbol{q})}, \tag{2.70}$$

assuming that $k_a > 1$. Consequently, the condition about the positive definiteness of $\boldsymbol{M}_d(\boldsymbol{q})$ is satisfied, and the proposed desired mass matrix becomes

$$\boldsymbol{M}_d(\boldsymbol{q}) = \begin{bmatrix} -\dfrac{k_a(\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q}))^2}{(\alpha(\boldsymbol{q})b_{12}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{22}(\boldsymbol{q}))} & -(\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q})) \\ -(\alpha(\boldsymbol{q})b_{11}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{12}(\boldsymbol{q})) & -(\alpha(\boldsymbol{q})b_{12}(\boldsymbol{q}) + \beta(\boldsymbol{q})b_{22}(\boldsymbol{q})) \end{bmatrix}. \tag{2.71}$$

Selecting in this way $\boldsymbol{M}_d(\boldsymbol{q})$ and $V_d(\boldsymbol{q})$ ensures that the first step of the IDA-PBC design is accomplished, meaning that the potential energy matching condition is satisfied.

Subsequently, the degree of freedom given by the matrix $\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p})$ is used to satisfy the kinetic energy matching equation. Whereas the solution of the potential energy matching target equation is completely novel, the approach proposed by [93] is followed to deal with the kinetic energy matching equation without solving any PDE. The kinetic energy PDE (2.32) can be expressed as

$$\boldsymbol{e}_2^T\nabla_q(\boldsymbol{p}^T\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{p}) - \boldsymbol{e}_2^T\boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_q(\boldsymbol{p}^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}) - 2j_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{e}_1^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p} = 0, \tag{2.72}$$

since $G^\perp = \boldsymbol{e}_2^T$ and $\boldsymbol{e}_2^T\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p}) = -j_2(\boldsymbol{q},\boldsymbol{p})\boldsymbol{e}_1^T$, where $\boldsymbol{e}_1^T = \begin{bmatrix}1 & 0\end{bmatrix}$ and

$$\boldsymbol{J}_2(\boldsymbol{q},\boldsymbol{p}) = \begin{bmatrix} 0 & j_2(\boldsymbol{q},\boldsymbol{p}) \\ -j_2(\boldsymbol{q},\boldsymbol{p}) & 0 \end{bmatrix}. \tag{2.73}$$

Since the desired mass matrix is fixed, the function $j_2(\boldsymbol{q},\boldsymbol{p})$ can be easily obtained solving (2.72) as an algebraic equation

$$j_2(\boldsymbol{q},\boldsymbol{p}) = \frac{\boldsymbol{e}_2^T\nabla_q(\boldsymbol{p}^T\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{p}) - \boldsymbol{e}_2^T\boldsymbol{M}_d(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\nabla_q(\boldsymbol{p}^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p})}{2\boldsymbol{e}_1^T\boldsymbol{M}_d^{-1}(\boldsymbol{q})\boldsymbol{p}}. \tag{2.74}$$

Finally, the control law can be computed from the standard equation (2.36).

Notice that, on one hand, the main complexity within the proposed approach is related to the presence of possible singular solutions of (2.74), that can be anyway numerically managed. The method used to satisfy the kinetic energy matching equation, inspired by the work [93], provides a solution that is not always well defined. Close to the equilibrium, the numerator of (2.74), which has a quadratic dependence on $\boldsymbol{p}$, tends towards zero faster than the denominator, which depends linearly on $\boldsymbol{p}$, thus avoiding any singularity issues. Despite this, a study about the denominator of the relation (2.74) reveals that it might be nullified if the equality $(b_{12}(\boldsymbol{q}) + b_{22}(\boldsymbol{q})k)p_1 = (b_{11}(\boldsymbol{q}) + b_{12}(\boldsymbol{q})k)p_2$ holds. This situation is addressed in practice by saturating the denominator of (2.74) when its absolute value is under a small enough threshold. On the other hand, through the method described in this Section, once the two functions $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ (providing a solution of the potential energy PDE, with a minimum in the equilibrium, and satisfying the condition (2.69)) are retrieved, an IDA-PBC can be generally designed for separable and non-separable systems, under the above mentioned conditions. The main advantage of this approach is that the set of nonlinear PDEs associated to the matching conditions is transformed in a set of algebraic equations, that, with a constructive method, allows to take into account the full nonlinear dynamics of the rolling systems for the IDA-PBC design. Also in the work [99] the authors propose a technique to avoid the solution of the matching conditions, and design a passivity-based control to shape the energy of a non-separable system, i.e. the circular ball and beam. Nevertheless, the method presented in [99] does not consider the gyroscopic term for control,

since the energy shaping is applied to a modified dynamics resulting from a geometric feedback transformation. Moreover, in the derivation proposed within this Thesis, the open-loop mass matrix and the desired closed-loop mass matrix can be dependent on both actuated and unactuated variables, differing from the approaches presented in [40, 83], where a necessary condition for the validity of the methods is that the mass matrix of the system depends only on the unactuated variable, in order to simplify the solution of the kinetic energy matching equations.

In the following Sections some applications of this methodology for non-separable systems are detailed, and numerical tests are presented to bolster the performance of the proposed control laws.

2.5.1 Ball and Beam Application

The previously introduced ball and beam case study is considered again, with dynamics (2.14), in which the elements of the non constant mass matrix

$$\boldsymbol{M}(s_h) = \begin{bmatrix} b_{11}(s_h) & b_{12} \\ b_{12} & b_{22} \end{bmatrix} \tag{2.75}$$

are detailed in Table 2.2. The potential energy of this system is

$$V(\boldsymbol{q}) = m_o g((d_h + \rho_o)\cos(\theta_h) - s_h \sin(\theta_h)). \tag{2.76}$$

Control designs for the ball and beam system are derived with other energy based approaches in [40, 83]. In these works the system is represented with a different model, that does not take into account the distance between the beam's CoM and its surface where the ball rolls.

According to the procedure illustrated in the previous Section, the expression of $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ have to be found such that the PDE (2.64) has a solution $V_d(\boldsymbol{q})$, with the properties of the desired potential energy function, and such that the condition (2.69) to obtain a positive definite desired mass matrix is satisfied.

Therefore, it is proposed $\alpha(\theta_h) = k sinc(\theta_h)^2$ and $\beta(\theta_h) = -sinc(\theta_h)$, where $k \in \mathbb{R}$ is a constant parameter to tune. Replacing these functions in (2.64), the target potential energy matching equation results

$$-m_o g \sin(\theta_h) + k sinc(\theta_h)\nabla_{\theta_h} V_d(\boldsymbol{q}) - sinc(\theta_h)\nabla_{s_h} V_d(\boldsymbol{q}) = 0, \tag{2.77}$$

which solution can be found in Mathematica as

$$V_d(\boldsymbol{q}) \to \frac{m_o g \theta_h^2}{2k} + f\left(\frac{\theta_h + k s_h}{k}\right), \tag{2.78}$$

[2]Notice that the $sinc(\cdot)$ function is analytic everywhere since: $sinc(x) = \frac{sin(x)}{x} = 1$, for $x = 0$.

where $f(\cdot)$ is a generic function of its argument. Therefore, the following desired potential function is proposed

$$V_d(\boldsymbol{q}) = \frac{m_o g \theta_h^2}{2k} - \cos\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right), \tag{2.79}$$

where s_h^* defines the desired equilibrium $\boldsymbol{q}^* = (0, s_h^*)$.

Moreover, it is verified that the Jacobian of (2.79),

$$\nabla V_d(\boldsymbol{q}) = \begin{bmatrix} \frac{m_o g}{k}\theta_h + \frac{k_f}{k}\sin\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) \\ k_f \sin\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) \end{bmatrix} \tag{2.80}$$

is zero at the desired equilibrium, and the Hessian of (2.79)

$$\nabla^2 V_d(\boldsymbol{q}) = \begin{bmatrix} \frac{m_o g}{k} + \frac{k_f^2}{k^2}\cos\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) & \frac{k_f^2}{k}\cos\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) \\ \frac{k_f^2}{k}\cos\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) & k_f^2\cos\left(\frac{k_f}{k}(\theta_h + k(s_h - s_h^*))\right) \end{bmatrix} \tag{2.81}$$

is positive definite at the desired equilibrium if $k > 0$ and $k_f \neq 0$, with $k_f \in \mathbb{R}$. The desired potential function has then a minimum in the equilibrium.

The next step consists in the computation of the entries $a_{12}(\boldsymbol{q})$ and $a_{22}(\boldsymbol{q})$ of $\boldsymbol{M}_d(\boldsymbol{q})$. Replacing $\alpha(\theta_h)$ and $\beta(\theta_h)$ in (2.66) yields

$$\begin{aligned} a_{12}(\boldsymbol{q}) &= -\frac{sinc(\theta_h)(kb_{11}(s_h) - b_{12})}{\Delta}, \\ a_{22}(\boldsymbol{q}) &= -\frac{sinc(\theta_h)(kb_{12} - b_{22})}{\Delta}, \end{aligned} \tag{2.82}$$

where $\Delta > 0$ is the determinant of (2.75). In addition, $a_{11}(\boldsymbol{q})$ is selected as in (2.68)

$$a_{11}(\boldsymbol{q}) = -\frac{k_a(kb_{11}(s_h) - b_{12})^2}{\Delta(kb_{12} - b_{22})} > 0, \tag{2.83}$$

imposing $0 < k < \frac{b_{22}}{b_{12}}$, and verifying that

$$a_{11}(\boldsymbol{q}) > -\frac{sinc(\theta_h)(kb_{11}(s_h) - b_{12})^2}{\Delta(kb_{12} - b_{22})}, \tag{2.84}$$

where the condition $k_a > 1$ must hold. Consequently, the condition (2.69) on $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ is satisfied, and then the desired mass matrix is positive definite.

As remark, notice that $0 < sinc(\theta_h) < 1$ if $-\pi < \theta_h < \pi$, then within the domain of interest of θ_h the $sinc(\cdot)$ function is positive. Additionally, for this application the inequality $\frac{b_{22}}{b_{12}} > 0$ is true, and therefore the condition on k is well-posed.

According to (2.60), the proposed desired mass matrix is

$$\boldsymbol{M}_d(\boldsymbol{q}) = \begin{bmatrix} -\frac{k_a(kb_{11}(s_h)-b_{12})^2}{(kb_{12}-b_{22})} & -sinc(\theta_h)(kb_{11}(s_h)-b_{12}) \\ -sinc(\theta_h)(kb_{11}(s_h)-b_{12}) & -sinc(\theta_h)(kb_{12}-b_{22}) \end{bmatrix}. \quad (2.85)$$

This choice of $\boldsymbol{M}_d(\boldsymbol{q})$ and $V_d(\boldsymbol{q})$ allows to satisfy the potential energy matching equation. The kinetic energy matching equation (2.32) is fulfilled employing (2.74). Finally, the IDA-PBC control law is computed from the standard equation (2.36).

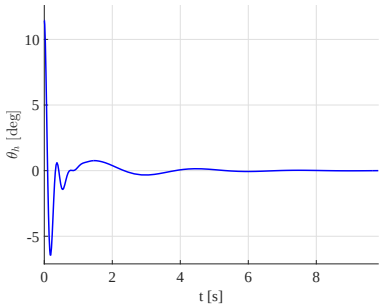
Ball and Beam Simulations

Some simulation tests are implemented in the Matlab/Simulink environment to validate the IDA-PBC approach with non constant mass matrix for the ball and beam case study. The parameters employed for simulations are $m_o = 0.05\ kg$, $\rho_o = 0.1\ m$, $I_o = m_o\rho_o^2$ for the ball, $d_h = 0.01\ m$, $I_h = 0.02\ m^2kg$ for the beam, and $g = 9.81\ m/s^2$.

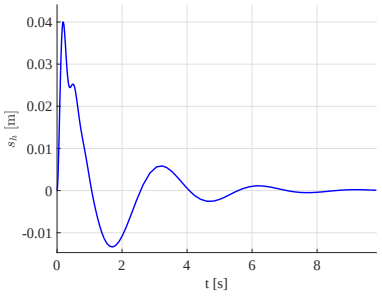
Fig. 2.5 shows the results obtained in the first case study, where the initial conditions are $\theta_h(0) = 0.2\ rad$, $\dot{\theta}_h(0) = 0.01\ rad/s$, $s_h(0) = 0\ m$, $\dot{s}_h(0) = 0\ m/s$. The desired value for the contact position is $s_h^* = 0\ m$, therefore $\boldsymbol{q}^* = (0, 0)$. The gains $k = 4$, $k_a = 10$, $k_v = 10$ and $k_f = 1$ are chosen for the control law. In particular, in Fig. 2.5(a), 2.5(c), 2.5(b) and 2.5(d), the time-series $\theta_h(t)$, $\dot{\theta}_h(t)$, $s_h(t)$, and $\dot{s}_h(t)$ are respectively shown; while Fig. 2.5(e) and Fig. 2.5(f) depict the evolution of the total energy given by the Hamiltonian function $\mathcal{H}(t)$ and the control torque $\tau(t)$ with respect to time. The pictures confirm that the control action is able to drive the state to the desired configuration, with a smooth control torque.

Fig. 2.6 shows the evolution of the controlled ball and beam system in the phase plane for different initial conditions, assuming that the desired configuration and the gains are not changed. In particular, $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0))$ are assigned as $(0.1, 0, 0.1, 0)$ in the black line, $(-0.1, 0, 0.1, 0)$ in the blue line, $(0.1, 0, -0.1, 0)$ in the red line, and $(-0.1, 0, -0.1, 0)$ in the green line. Moreover, Fig. 2.7 depicts the surface of the desired potential function V_d, confirming that it has a minimum in the equilibrium. The red line represents the 3D evolution of $V_d(t)$ in (2.79) with respect to $\theta_h(t)$ and $s_h(t)$, for the initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (-0.1, 0, -0.1, 0)$.

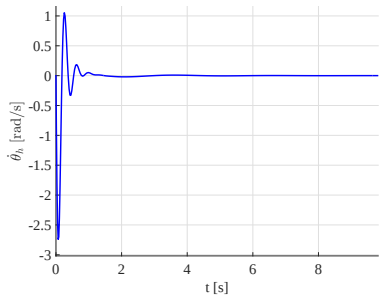
Another test is performed considering the desired configuration $\boldsymbol{q}^* = (0, 0.08)$, and the initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0, 0.1, 0, 0.05)$, see Fig. 2.8. In particular, in Fig. 2.8(a), 2.8(c), 2.8(b) and 2.8(d), the evolution of $\theta_h(t)$, $\dot{\theta}_h(t)$, $s_h(t)$, and $\dot{s}_h(t)$ are respectively shown; while Fig. 2.8(e) and Fig. 2.8(f) depict the evolution of the total energy of the system $H(t)$ and the control action $\tau(t)$ with respect to time. A comparison of the performance of the controller with



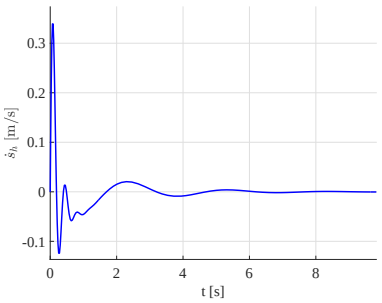
(a) Evolution of $\theta_h(t)$.



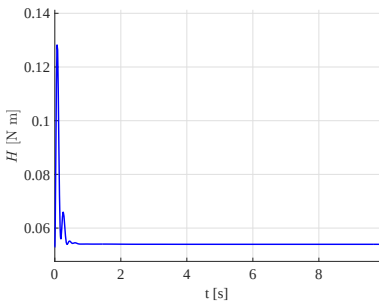
(b) Evolution of $s_h(t)$.



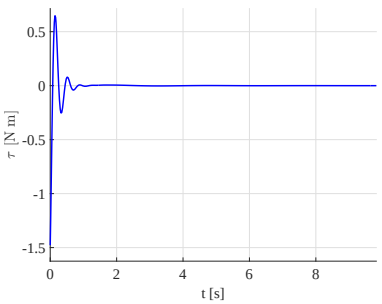
(c) Evolution of $\dot{\theta}_h(t)$.



(d) Evolution of $\dot{s}_h(t)$.



(e) Evolution of the total energy $H(t)$.



(f) Evolution of the control action $\tau(t)$.

Figure 2.5: Simulation test for the IDA-PBC of the ball and beam system in case of non constant mass matrix. The initial conditions are $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0.2, 0.01, 0, 0)$, and the desired configuration is $\boldsymbol{q}^* = (0, 0)$.

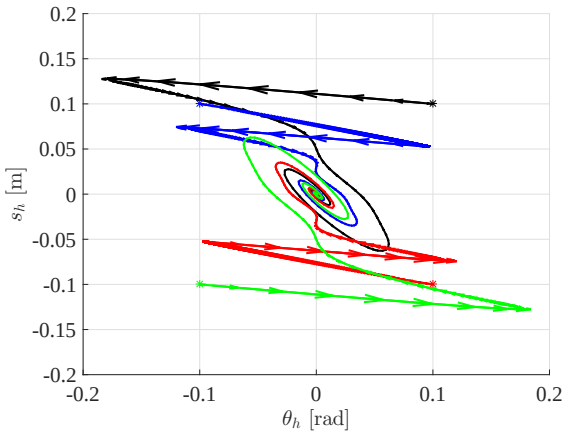


Figure 2.6: Evolution of the controlled ball and beam system in the phase plane for different initial conditions (highlighted by stars) ($\theta_h(0)$, $\dot{\theta}_h(0)$, $s_h(0)$, $\dot{s}_h(0)$): $(0.1, 0, 0.1, 0)$ black line, $(-0.1, 0, 0.1, 0)$ blue line, $(0.1, 0, -0.1, 0)$ red line, $(-0.1, 0, -0.1, 0)$ green line. The reference is $\boldsymbol{q}^* = (0, 0)$.

lower gains $k_a = k_v = 10$, and higher gains $k_a = k_v = 15$, reveals that lower values of the gains k_a and k_v provide a better result.

2.5.2 Circular Ball and Beam Application

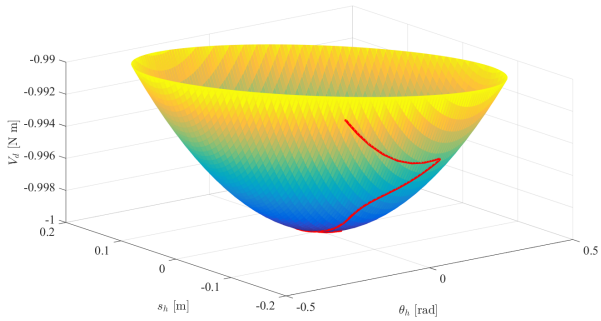
The circular ball and beam is selected as additional case study, see Fig. 2.9. The complexity in the stabilization of such a system is due to the presence of two unstable equilibrium points and to the relevant effect of the gyroscopic forces in its dynamics. These factors makes the nonlinear control design more complicated for the ball on circular beam system than the conventional ball on a straight beam system [99].

In order to deal with the modeling of this specific application, the asymmetric nonprehensile planar rolling manipulation dynamics (2.23) is employed. In this system the assumption that the hand can only rotate around its CoM has been relaxed, i.e. the hand rotates around a point that does not correspond to its CoM.

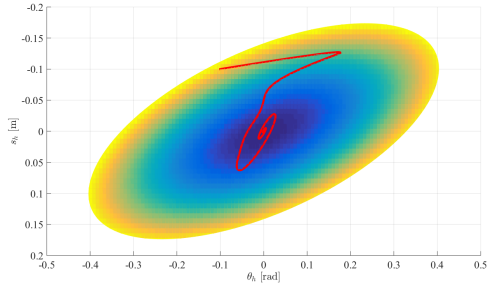
The shape of the hand (the circular beam) is parametrized through the chart

$$\boldsymbol{c}_h^h(s_h) = \rho_h \left[-\sin \tfrac{s_h}{\rho_h} \quad \cos \tfrac{s_h}{\rho_h} \right]^T, \tag{2.86}$$

with $\rho_h \in \mathbb{R}^+$ the radius of the hand. The shape of the ball is parametrized by

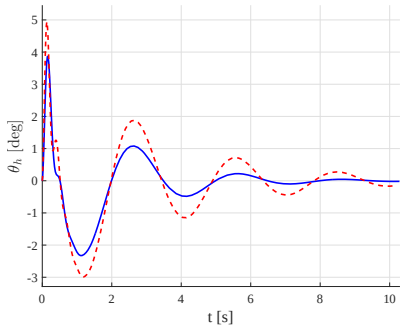


(a) 3D view of V_d.

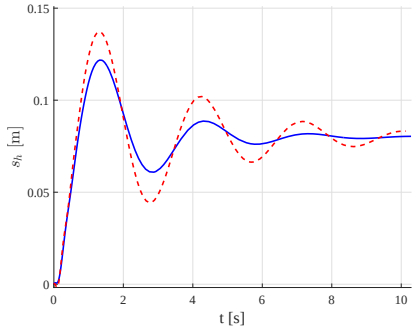


(b) 2D view of V_d in the plane (θ_h, s_h).

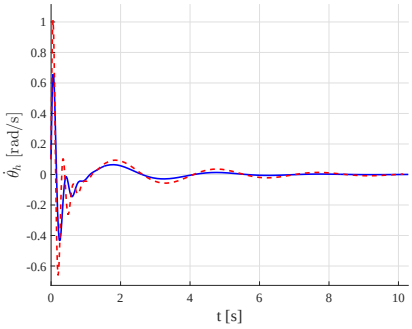
Figure 2.7: Surface of the desired potential function V_d with a minimum in the desired equilibrium configuration. The red line represents the 3D evolution of $V_d(t)$ with respect to $\theta_h(t)$ and $s_h(t)$, considering as initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (-0.1, 0, -0.1, 0)$, and the desired configuration $\boldsymbol{q}^* = (0, 0)$.



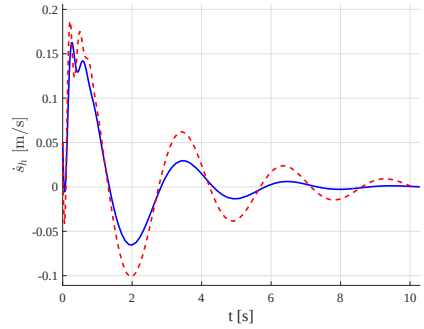
(a) Evolution of $\theta_h(t)$.



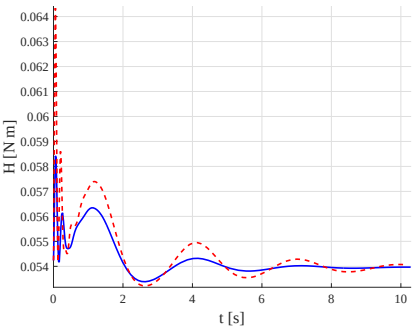
(b) Evolution of $s_h(t)$.



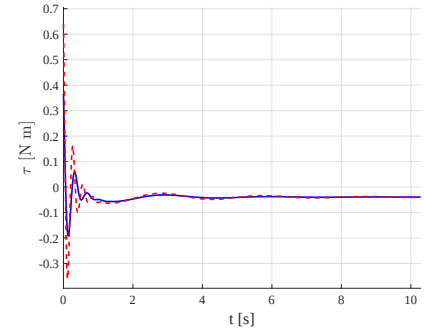
(c) Evolution of $\dot{\theta}_h(t)$.



(d) Evolution of $\dot{s}_h(t)$.



(e) Evolution of the total energy $H(t)$.



(f) Evolution of the control action $\tau(t)$.

Figure 2.8: Simulation test for the IDA-PBC of the ball and beam system in case of non constant mass matrix. The initial conditions are $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0, 0.1, 0, 0.05)$, and the desired configuration is $\boldsymbol{q}^* = (0, 0.08)$. Two tests with lower gains $(k, k_a, k_v, k_f) = (4, 10, 10, 1)$ (solid blue lines), and higher gains $(k, k_a, k_v, k_f) = (4, 15, 15, 1)$ (dashed red lines) are presented.

Table 2.3: Elements of the dynamic model for the circular ball and beam example.

b_{11}	$c_{b1} + c_{b2}\cos(\frac{s_h}{\rho_h})$
b_{12}, b_{21}	$c_{b3} + c_{b4}\cos(\frac{s_h}{\rho_h})$
b_{22}	$I_o\kappa_r^2 + m_o\rho_o^2\kappa_r^2$
g_1	$-m_o g(\rho_h+\rho_o)\sin\left(\theta_h + \frac{s_h}{\rho_h}\right) - (m_h+m_o)g\lambda s_{\theta_h}$
g_2	$-m_o g\kappa_r\rho_o \sin\left(\theta_h + \frac{s_h}{\rho_h}\right)$

the chart

$$\boldsymbol{c}_o^o(s_h) = -\rho_o \begin{bmatrix}\sin\frac{s_h}{\rho_o} & \cos\frac{s_h}{\rho_o}\end{bmatrix}^T, \tag{2.87}$$

with $\rho_o \in \mathbb{R}^+$ the radius of the ball. The position of the CoM of the hand in Σ_w is given by $\boldsymbol{p}_h(\theta_h) = \lambda \begin{bmatrix}-s_{\theta_h} & c_{\theta_h}\end{bmatrix}^T$, with $\lambda \in \mathbb{R}^+$. Considering (2.2), the relative curvature is given by

$$\kappa_r = \frac{\rho_h + \rho_o}{\rho_h\rho_o}. \tag{2.88}$$

The upper disk angular velocity is given by $\dot{\theta}_o = \dot{\theta}_h + \kappa_r\dot{s}_h$. The vectors $\boldsymbol{\gamma}(\boldsymbol{q})$ and $\boldsymbol{\eta}(\boldsymbol{q})$ are computed like in (2.19):

$$\boldsymbol{\gamma}(\boldsymbol{q}) = -(\rho_h+\rho_o)\left[\cos\left(\theta_h+\tfrac{s_h}{\rho_h}\right) \quad \sin\left(\theta_h+\tfrac{s_h}{\rho_h}\right)\right]^T - \lambda\begin{bmatrix}c_{\theta_h} & s_{\theta_h}\end{bmatrix}^T, \tag{2.89}$$

and

$$\boldsymbol{\eta}(\boldsymbol{q}) = -\rho_o\kappa_r\left[\cos\left(\theta_h+\tfrac{s_h}{\rho_h}\right) \quad \sin\left(\theta_h+\tfrac{s_h}{\rho_h}\right)\right]^T. \tag{2.90}$$

Therefore, the elements of the dynamics (2.23) are detailed in Table 2.3 for the circular ball and beam, where $c_{b1} = I_h + I_o + \lambda^2(m_h+m_o) + m_o(\rho_h+\rho_o)^2$, $c_{b2} = 2\lambda m_o(\rho_h+\rho_o)$, $c_{b3} = I_o\kappa_r + m_o\frac{(\rho_h+\rho_o)^2}{\rho_h}$, and $c_{b4} = m_o\lambda\rho_o\kappa_r$. The mass matrix has now the following structure

$$\boldsymbol{M}(s_h) = \begin{bmatrix} b_{11}(s_h) & b_{12}(s_h) \\ b_{12}(s_h) & b_{22}\end{bmatrix}. \tag{2.91}$$

Notice that $b_{22}, c_{b1}, c_{b2}, c_{b3}, c_{b4}$ are constant values, while the Coriolis elements $c_{11}, c_{12}, c_{21}, c_{22}$, which are not constant, are derived through (2.13). The potential energy of the circular ball and beam is given by

$$V(\boldsymbol{q}) = g(m_o(\rho_h+\rho_o)\cos\left(\theta_h + \frac{s_h}{\rho_h}\right) + (m_o+m_h)\lambda\cos(\theta_h)). \tag{2.92}$$

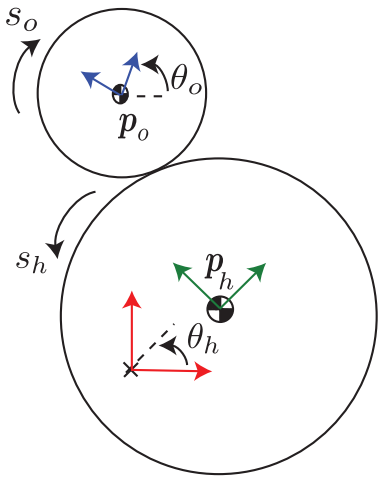


Figure 2.9: A representation of the circular ball and beam system, where the center of rotation of the hand (indicated by the $\times$ symbol) does not correspond to its geometric center. In red the world fixed frame Σ_w. In green the hand frame Σ_h, while in blue the object frame Σ_o, placed at the respective centers of mass.

Following the proposed methodology, the expression of $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ have to be found again such that the PDE (2.64) is solvable, $V_d(\boldsymbol{q})$ has a minimum in the equilibrium, and the additional constraint (2.69) is verified. Therefore, it is proposed $\alpha(\boldsymbol{q}) = sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)$ and $\beta(\boldsymbol{q}) = ksinc\left(\theta_h + \frac{s_h}{\rho_h}\right)$, where $k \in \mathbb{R}$ is a constant gain to tune. Employing these functions in (2.64), the target potential energy matching equation becomes

$$-c_v \sin\left(\theta_h + \frac{s_h}{\rho_h}\right) + sinc\left(\theta_h + \frac{s_h}{\rho_h}\right) \nabla_{\theta_h} V_d(\boldsymbol{q}) + ksinc\left(\theta_h + \frac{s_h}{\rho_h}\right) \nabla_{s_h} V_d(\boldsymbol{q}) = 0, \tag{2.93}$$

where $c_v = m_o g \frac{\rho_h + \rho_o}{\rho_h}$ is a positive parameter. The solution of the PDE (2.93) provided by Mathematica is

$$V_d(\boldsymbol{q}) \to \frac{c_v \theta_h^2 (\rho_h - k) + 2 c_v \theta_h s_h}{2\rho_h} + f(s_h - k\theta_h), \tag{2.94}$$

where $f(\cdot)$ is a generic function of its argument. Therefore, the following desired potential function is proposed

$$V_d(\boldsymbol{q}) = \frac{c_v \theta_h^2 (\rho_h - k) + 2 c_v \theta_h s_h}{2\rho_h} + k_f (s_h - k\theta_h)^2, \tag{2.95}$$

where $k_f \in \mathbb{R}$ is a constant parameter. Then, the Jacobian of (2.95)

$$\nabla V_d(\boldsymbol{q}) = \begin{bmatrix} \frac{c_v(-k\theta_h+\theta_h\rho_h+s_h)}{\rho_h} + 2kk_f(k\theta_h - s_h) \\ \frac{c_v\theta_h}{\rho_h} - 2kk_f\theta_h + 2k_f s_h \end{bmatrix} \tag{2.96}$$

is zero at the desired equilibrium, $\boldsymbol{q}^* = (0, 0)$, and the Hessian of (2.95)

$$\nabla^2 V_d(\boldsymbol{q}) = \begin{bmatrix} c_v + 2k^2k_f - \frac{c_v k}{\rho_h} & -2kk_f + \frac{c_v}{\rho_h} \\ -2kk_f + \frac{c_v}{\rho_h} & 2k_f \end{bmatrix} \tag{2.97}$$

is positive definite at the desired equilibrium, if $k > -\rho_h$ and $k_f > \frac{c_v}{2\rho_h(k+\rho_h)}$. In this way, the desired potential energy has a minimum in the equilibrium.

In order to compute the entries of the desired mass matrix $a_{12}(\boldsymbol{q})$ and $a_{22}(\boldsymbol{q})$ the proposed functions $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ are substituted in (2.66)

$$\begin{aligned} a_{12}(\boldsymbol{q}) &= -\frac{sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{11}(s_h) + kb_{12}(s_h))}{\Delta}, \\ a_{22}(\boldsymbol{q}) &= -\frac{sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{12}(s_h) + kb_{22})}{\Delta}, \end{aligned} \tag{2.98}$$

The resulting desired mass matrix is

$$\boldsymbol{M}_d(\boldsymbol{q}) = \begin{bmatrix} \Delta a_{11}(\boldsymbol{q}) & -sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{11}(s_h) + kb_{12}(s_h)) \\ -sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{11}(s_h) + kb_{12}(s_h)) & -sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{12}(s_h) + kb_{22}) \end{bmatrix}, \tag{2.99}$$

which is structurally symmetric, and $a_{11}(\boldsymbol{q})$ is selected to obtain positive definiteness. As a matter of fact, according to (2.68),

$$a_{11}(\boldsymbol{q}) = -\frac{k_a sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{11}(\boldsymbol{q}) + kb_{12}(\boldsymbol{q}))^2}{\Delta(b_{12}(\boldsymbol{q}) + kb_{22})} > 0, \tag{2.100}$$

assuming that $-\rho_h < k < -\frac{(c_{b3}+c_{b4})}{b_{22}}$. Moreover,

$$a_{11}(\boldsymbol{q}) > -\frac{sinc\left(\theta_h + \frac{s_h}{\rho_h}\right)(b_{11}(\boldsymbol{q}) + kb_{12}(\boldsymbol{q}))^2}{\Delta(b_{12}(\boldsymbol{q}) + kb_{22})}, \tag{2.101}$$

if the following condition holds: $k_a > 1$. Consequently, the condition (2.69) on $\alpha(\boldsymbol{q})$ and $\beta(\boldsymbol{q})$ is satisfied, and the desired mass matrix is always positive definite.

It is remarkable that $0 < sinc\left(\theta_h + \frac{s_h}{\rho_h}\right) < 1$ if $-\pi < \left(\theta_h + \frac{s_h}{\rho_h}\right) < \pi$, therefore, within the domain of interest of θ_h and s_h, the $sinc(\cdot)$ function is positive.

Notice that for this application $\rho_h > \frac{(c_{b3}+c_{b4})}{b_{22}}$ and therefore the condition on k is well-posed.

This choice of $\boldsymbol{M}_d(\boldsymbol{q})$ and $V_d(\boldsymbol{q})$ allows to satisfy the potential energy matching equation. Even in this case study, the kinetic energy matching equation (2.32) is fulfilled employing (2.74). The IDA-PBC control law is computed from the standard equation (2.36).

Circular Ball and Beam Simulations

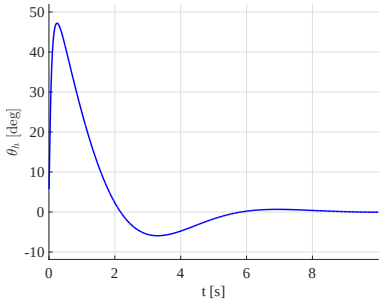
Some simulation tests are implemented in the Matlab/Simulink environment to validate the IDA-PBC approach with non constant mass matrix for the ball and beam case study. The parameters employed for simulations are $m_o = 0.0216\ g$, $\rho_o = 0.075\ m$, $I_o = 0.0001215\ m^2 g$ for the upper disk, $\rho_h = 0.15\ m$, $m_h = 0.2356\ g$, $I_h = 0.0053\ m^2 g$ for the lower disk, and $g = 9.81\ m/s^2$, $\lambda = 0.05\ m$.

Fig. 2.10 shows the results obtained in the first case study, where the initial conditions are $\theta_h(0) = 0.1\ rad$, $\dot{\theta}_h(0) = 0\ rad/s$, $s_h(0) = -0.05\ m$, $\dot{s}_h(0) = 0\ m/s$. The equilibrium state is $\boldsymbol{q}^* = (0, 0)$. The gains $k = -0.13$, $k_a = 10$, $k_v = 10$ and $k_f = 60$ are chosen for the control law. In particular, in Fig. 2.10(a), 2.10(c), 2.10(b) and 2.10(d), the timeseries $\theta_h(t)$, $\dot{\theta}_h(t)$, $s_h(t)$, and $\dot{s}_h(t)$ are respectively shown; while Fig. 2.10(e) and Fig. 2.10(f) depict the evolution of the total energy given by the Hamiltonian function $\mathcal{H}(t)$ and the control torque $\tau(t)$ with respect to time. The pictures confirm that the control action is able to drive the state to the desired configuration, with a smooth control torque.

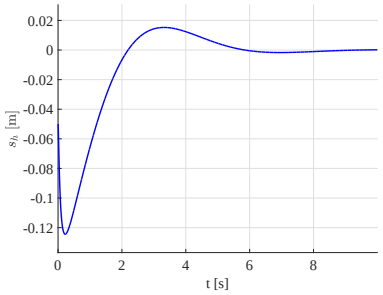
Fig. 2.11 depicts the surface of the desired potential function V_d, confirming that it has a minimum in the equilibrium. Assuming as initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0.1, 0, -0.05, 0)$, and the desired configuration $\boldsymbol{q}^* = (0, 0)$, the red line represents the 3D evolution of $V_d(t)$ with respect to $\theta_h(t)$ and $s_h(t)$. Fig. 2.11(c) corresponds to the numerical values in Fig. 2.11(b) where the surface V_d and the corresponding time evolution are rotated 0.144 rad around the vertical direction, $\boldsymbol{e}_3 = \begin{bmatrix} 0, & 0, & 1 \end{bmatrix}^T$, and enlarged for better visualization.

Another test is performed considering the desired configuration $\boldsymbol{q}^* = (0, 0)$, and the initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (-0.4, 0.1, 0.02, 0.1)$, see Fig. 2.12. In particular, in Fig. 2.12(a), 2.12(c), 2.12(b) and 2.12(d), the evolution of $\theta_h(t)$, $\dot{\theta}_h(t)$, $s_h(t)$, and $\dot{s}_h(t)$ are respectively shown; while Fig. 2.12(e) and Fig. 2.12(f) depict the evolution of the total energy of the system $H(t)$ and the control action $\tau(t)$ with respect to time. A comparison of the performance of the controller with lower gains $k_a = 10$, $k_v = 10$, $k_f = 60$, and higher gains $k_a = 15$, $k_v = 30$, $k_f = 70$, reveals that higher values of k_a, k_v, k_f provide a better result.

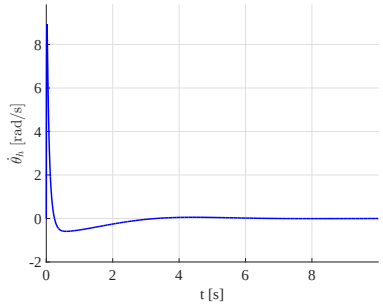
Unfortunately, experiments are not yet available. The hardware set-up and the code for testing, in particular the control approach designed for the circular ball and beam application, is currently under development.



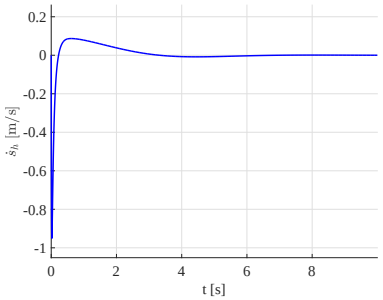
(a) Evolution of $\theta_h(t)$.



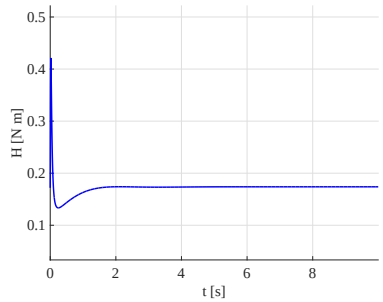
(b) Evolution of $s_h(t)$.



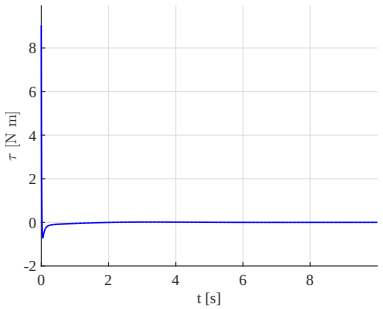
(c) Evolution of $\dot{\theta}_h(t)$.



(d) Evolution of $\dot{s}_h(t)$.

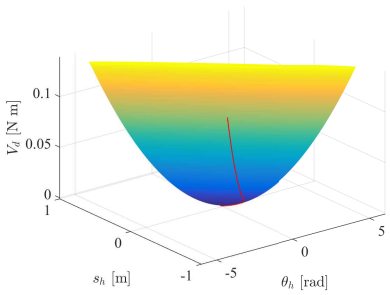


(e) Evolution of the total energy $H(t)$.

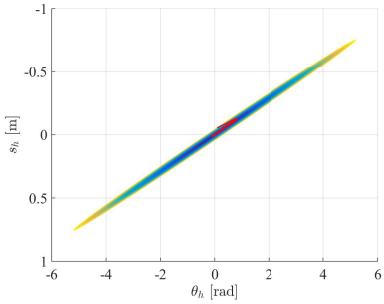


(f) Evolution of the control action $\tau(t)$.

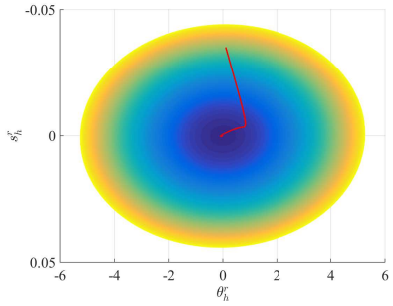
Figure 2.10: Simulation test for the IDA-PBC of the circular ball and beam system. The initial conditions are $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0.1,\, 0,\, -0.05,\, 0)$, and the desired configuration is $\boldsymbol{q}^* = (0, 0)$.



(a) 3D view of V_d.

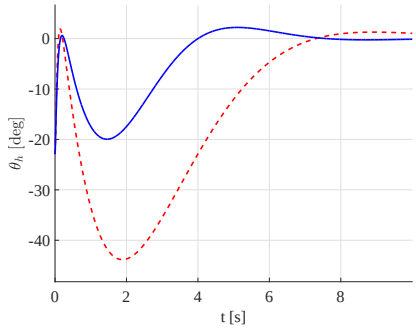


(b) 2D view of V_d in the plane (θ_h, s_h).

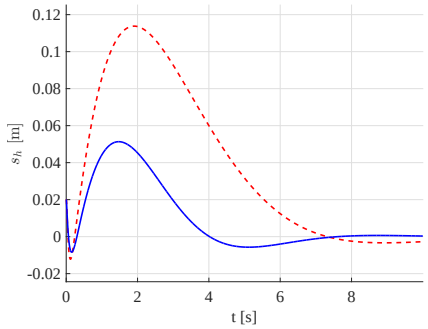


(c) 2D view of V_d with respect to the rotated variables (θ_h^r, s_h^r) for better visualization.

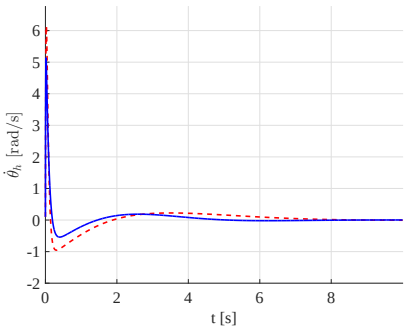
Figure 2.11: Surface of the desired potential function V_d with a minimum in the desired equilibrium configuration. The red line represents the 3D evolution of $V_d(t)$ with respect to $\theta_h(t)$ and $s_h(t)$, considering as initial conditions $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (0.1, 0, -0.05, 0)$, and the desired configuration $\boldsymbol{q}^* = (0, 0)$.



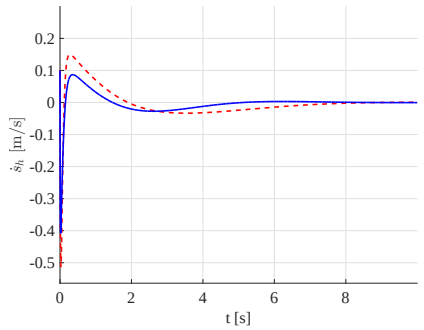
(a) Evolution of $\theta_h(t)$.



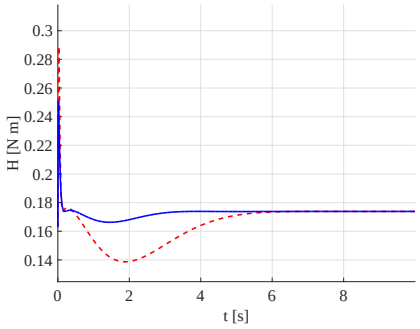
(b) Evolution of $s_h(t)$.



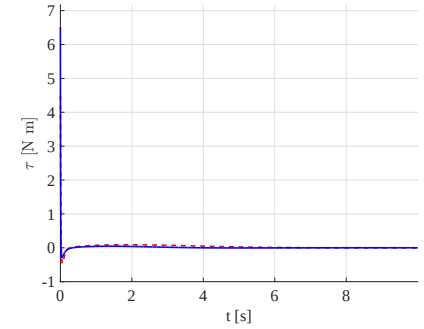
(c) Evolution of $\dot{\theta}_h(t)$.



(d) Evolution of $\dot{s}_h(t)$.



(e) Evolution of the total energy $H(t)$.



(f) Evolution of the control action $\tau(t)$.

Figure 2.12: Simulation test for the IDA-PBC of the circular ball and beam system. The initial conditions are $(\theta_h(0), \dot{\theta}_h(0), s_h(0), \dot{s}_h(0)) = (-0.4,\ 0.1,\ 0.02,\ 0.1)$, and the desired configuration is $\boldsymbol{q}^* = (0, 0)$. Two tests with lower gains $(k, k_a, k_v, k_f) = (-0.13, 10, 10, 60)$ (dashed red lines), and higher gains $(k, k_a, k_v, k_f) = (-0.13, 15, 30, 70)$ (solid blue lines) are presented.

2.6 Conclusion

In this Chapter, a general model for nonprehensile dynamic rolling manipulation tasks in 2D is presented. This class of mechanical systems is specified for symmetric or asymmetric planar rolling systems, i.e., assuming respectively that the center of rotation is equivalent or not to the geometric center. Successively, through the Legendre transformation, the Lagrangian dynamics for nonprehensile rolling manipulation is transformed into the Hamiltonian canonical equations of motion. Thereafter, the Port-Hamiltonian framework for passivity-based control of mechanical systems is introduced. An IDA-PBC approach is proposed for the class of separable - with constant mass matrix - planar rolling systems. A general analytic expression of the desired potential energy is derived for such class of systems. This function is a solution of the partial differential equations resulting from the matching equation. The balancing of the disk on disk system is considered as application for the proposed control approach. Subsequently, removing the assumption of constant mass matrix, a method to solve the matching equations for non-separable planar rolling systems is proposed with the aim to reduce the design complexity, while preserving the effectiveness of the IDA-PBC method. In particular the straight ball and beam, and the circular ball and beam, are considered as examples of non-separable systems. The proposed approach employs a target potential energy matching equation, depending on a parametrization of the desired closed-loop mass matrix, to simultaneously simplify the recognition of the desired mass matrix and select the desired energy function for the closed-loop system. This is a systematic approach that simplifies the IDA-PBC design, transforming the set of PDEs resulting from the matching conditions in a set of algebraic equations.

Drawing a conclusion, the results within this Chapter extend previous results in the state of the art in both modeling and control. The dynamic model for nonprehensile planar rolling manipulation described in [64] for symmetric planar rolling has been extended, removing the assumption that the center of rotation of the hand corresponds to its geometric center (asymmetric planar rolling manipulation). In [64], a partial feedback linearization is applied to the disk on disk and to the ball and beam systems, considering an approximated model (with constant mass matrix) for control design. The IDA-PBC methodology here presented allows to include in the control law the gravitational and the Coriolis effects, avoiding in this way the use of approximate models for control design. This approach differs from the feedback linearization where a linear dynamics is imposed at the expense of canceling the nonlinear dynamics of the system. Moreover, extending the result proposed in [32], a passivity-based control is designed for a whole class of nonprehensile planar rolling systems. Numerical tests confirm the validity of the control approach, in different case studies, for each of

the above mentioned planar rolling benchmark systems.

Chapter 3

Optimal Motion Planning of Impact Manipulation

3.1 Introduction

Besides rolling manipulation, other kinds of dynamic nonprehensile manipulation, involving impacts between manipulator and manipulated object, are inspected in this Thesis. Such tasks are very challenging to control because of their fast and scarcely predictable dynamics. In the batting task, for example, an object (a ball) is intercepted by the end-effector (a paddle), without grasping it, and it is thrown towards a precise goal. This motion primitive is typically used by athletes, such as baseball batters or table tennis players. Also jugglers use this primitive when their hands control the continuous motion of one or more objects through intermittent contacts. Such dynamic motions require high velocity and precision. The design of planning and control methods to deal with them would strongly enhance capabilities of robot manipulators, extending the workspace size and dimensionality, and the repertoire of available actions [70].

The robotic table tennis is one of the applications considered in this Chapter. One of the first real-time table tennis robot prototype is proposed by Andersson in [5]. It is built on a commercial PUMA 260 robot arm, which is a 5 degrees of freedom (DoF) industrial robot. In [6], the same author employs fifth-order polynomials to generate a trajectory for the paddle intercepting the ball. The trajectory of the arm of the robot is adjusted while the ball is in free flight through a sensor-driven approach. A low-cost ping-pong player prototype is proposed in [2]. The authors propose to detect the location of the ball combining the information about the ball and its shadow on the table. An expert module defines the desired return point for the ball. A high-speed trajectory planner is presented in [103],

where the authors propose to split the motion in two phases: a high speed phase and a reactive one, named swing and hitting motion, respectively. The hitting point is estimated before the impact, while the batting task is accomplished by modifying this point through a visual feedback. An approach to keep stability of a biped humanoid robot while playing table tennis is presented in [112]. In that work, an optimal momentum compensation method using lower body joints to cancel the momentum generated by arms is discussed. Additionally, the authors of [65] propose an algorithm for returning a table tennis ball to a desired position with a desired spin. An approximated hybrid aerodynamics of the ball is exploited to compute the configuration of the paddle at impact time to accomplish the batting task.

In the field of artificial intelligence, learning techniques exploiting data-driven perspectives instead of inverse kinematics and physical models are often applied to the robotic table tennis game, through offline training of the system. The work presented in [121] shows instead some experiments on two humanoid robots playing ping-pong. The approach employs an adjustment of the trajectory prediction from an offline training of the model parameters based on a neural network. The work in [51] presents an approach for robotic table tennis game consisting in two stages: a first regression phase, in which the joint trajectories are generated to strike the incoming ball, and a second reinforcement learning phase, where the joint trajectories are updated to properly return the ball. Moreover, in [41] a probabilistic approach to intercept a table tennis ball in space is presented. A probabilistic representation is employed to find the initial time and the duration of the movement primitive maximizing the likelihood of hitting the ball. Modeling and learning of complex motor tasks is applied to the robotic table tennis also in [76].

Furthermore, planning and control of rhythmic tasks has been a very active research area over the last years. In [21, 22] the authors investigated the stabilization of juggling tasks. In their work they considered the trajectory control of a puck; as control input they used a bar which is actuated around a revolute joint. They proposed and experimentally verified the well known mirror algorithm to control the system. Several works have investigated robotic juggling with emphasis on the aspect of learning control [1, 100, 101, 102]. Following the work of Buehler, Brogliato and Lynch proposed stabilizing feedback laws for the planar juggler. In [20] the juggling robot is considered as a complementary-slackness hybrid mechanical system, where the force input mainly consists of a family of dead-beat feedback control laws, introduced via a recursive procedure. The work [67] shows how to control a one joint revolute arm to bat juggle a planar disk in a gravity field to a desired juggling limit cycle. It is required an open connected subset of the state space such that, for every state in the subset, there exists a feedback control which keeps the system in a closed orbit (forced recurrence).

The controller is based on real-time nonlinear optimization using a model of the discrete dynamics and the recurrent control as an initial seed. In the two works [91, 92] feedback control strategies are investigated to continuously bounce a ball in the air, and discuss the influence of the impact acceleration on the robustness of the system to parameter uncertainties.

Trajectory tracking control for a one DoF juggling system is deeply studied by [98], also with multiple balls. The mechanical systems with impacts is modeled as hybrid dynamical systems, given by a set of differential equation/inclusion and difference equation/inclusion, on specific subsets of the state space. Juggling experiments are presented in [116] to validate a hybrid control algorithm, that is capable of tracking a periodic reference trajectory. In [88, 89] the authors propose a bouncing ball robot named blind juggler. They took the impact time measurements as feedback and proved that the closed-loop performance is only marginally improved as compared to open-loop control. The paper [9] inspects the flower-stick juggling task with an analytical technique. The stability of the approach is analyzed exploiting the concept of virtual connecting manipulation, using the Poincarè maps.

Interestingly, [114] tackles the concept of tossing and catching manipulation to pass an object between a couple of one DoF manipulators. The authors provide a kinematic model for the task and an iterative learning control approach. The paper [4] proposes a planner to reconfigure planar polygonal objects by juggling between the palm of two hand-like manipulators. The authors analyze the problem of dynamic grasp when the palm holds the object and minimize the relative velocity between object and hand.

In this Chapter, a minimum acceleration planner in SE(3), based on the theory of differential geometry, is used to generate an optimal path for the manipulator, according to what is proposed in [120]. By choosing the acceleration norm as measure of smoothness, the trajectories can be made to satisfy boundary conditions on the velocities. In order to ensure that the computed trajectories are independent of the parameterization of positions and orientations, the notions of Riemannian metric and covariant derivative are borrowed from differential geometry and the problem is formulated as a variational problem on the Lie group of spatial rigid body displacements. This optimal planner is employed in the algorithms designed for two impact manipulation tasks, i.e. the batting [108], and the juggling [106] tasks. The hybrid dynamics of these tasks are taken into account in the respective planning algorithms, together with nonlinear estimation methods that allow to properly intercept the ball in time. The hybrid dynamic models here considered are based on the works [65, 79, 80], where the original application is the table tennis game. The nonlinear estimation techniques for these nonprehensile tasks are instead inspired by [63], which is motivated by a prehensile manipulation (the ball catching task with robot endowed with an eye in-hand

monocular visual system integrated into a gripper). The algorithms presented in this Chapter are implemented with numerical simulations in the Matlab/Simulink environment, along with some comparisons with other state-of-the-art planning algorithms.

3.2 Generation of the Optimal Path

The first problem that is addressed in this Chapter is the generation of an optimal trajectory in SE(3) for the robot end-effector to hit the ball at the impact time. By knowing the initial configuration of the end-effector, and after having computed its desired configuration at the impact time (as detailed in the following Sections), many different paths and trajectories can be followed to fulfill the requirements. A wiser method would be to find this path such that a certain objective function is optimized. The acceleration functional of the end-effector along the path in SE(3) is chosen as objective function. In this way, the resulting trajectories require smooth velocities and accelerations at the end-effectors of the robot. Notice that the planned trajectories in SE(3) are independent from the chosen representation for the orientation of the robot end-effector. Changing the representation for the orientation do not imply a change in the planner, but only a different mapping from SE(3).

In the following, a brief background about differential geometry is provided. Afterwards, the theory about the minimum acceleration planner for the end-effector is reported.

3.2.1 Brief Background about Differential Geometry

Within this context, trajectories for which it is possible to specify both the initial and final position and velocity are of interest. The motion can be specified in either the joint space, which is a torus, or the task space, which is SE(3). At this stage, to be general, it is assumed that the path is generated on an arbitrary Riemannian manifold $\mathcal{M}$ (see [31] for more details).

Let $\gamma : (a, b) \to \mathcal{M}$ be the path, and $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ be the metric on $\mathcal{M}$. Let $f : (-\epsilon, \epsilon) \times (a, b) \to \mathcal{M}$ be a proper variation of γ satisfying

$$\begin{aligned} f(0, t) &= \gamma(t), \quad \forall t \in (a, b) \\ f(s, a) &= \gamma(a), \quad \text{and} \quad f(s, b) = \gamma(b). \end{aligned}$$

Two vector fields are relevant along the path γ. The former is the variation field which is defined by

$$S_{\gamma(s)} := \frac{\partial f(s, t)}{\partial s} = \frac{\mathrm{d} f_t(s)}{\mathrm{d} s}.$$

The latter is the velocity vector field of γ, given by

$$V_{\gamma(s)} := \frac{\mathrm{d}\gamma(t)}{\mathrm{d}t} = \frac{\partial f(s,t)}{\partial t} = \frac{\mathrm{d}f_s(t)}{\mathrm{d}t}.$$

Hence, the Levi-Civita connection ∇ is introduced to perform calculus on the curves of $\mathcal{M}$. Therefore, given a curve $\gamma(t)$ and a connection, there exists a covariant derivative denoted by $\frac{\mathrm{D}}{\mathrm{d}t}$.

The Levi-Civita connection satisfies the following compatibility and symmetry conditions:

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle\!\langle U, W\rangle\!\rangle = \langle\!\langle \frac{\mathrm{D}U}{\mathrm{d}t}, W\rangle\!\rangle + \langle\!\langle U, \frac{\mathrm{D}W}{\mathrm{d}t}\rangle\!\rangle \tag{3.1a}$$

$$\nabla_X Y - \nabla_Y X = [X, Y] \tag{3.1b}$$

for any vector fields U and W, along the differentiable curve γ, and any vector field $X, Y \in \mathfrak{X}(\mathcal{M})$, where $\mathfrak{X}(\mathcal{M})$ is the set of all vector fields on $\mathcal{M}$.

The curvature R of a Riemannian manifold $\mathcal{M}$ associates to every pair $X, Y \in \mathfrak{X}(\mathcal{M})$ a mapping $R(X,Y) : \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$ given by

$$R(X,Y)Z = \nabla_Y \nabla_X Z - \nabla_X \nabla_Y Z + \nabla_{[X,Y]} Z,$$

where $Z \in \mathfrak{X}(\mathcal{M})$. In the next subsection, the following properties related to the curvature are employed

$$\frac{\mathrm{D}}{\partial t}\frac{\mathrm{D}}{\partial s}X - \frac{\mathrm{D}}{\partial s}\frac{\mathrm{D}}{\partial t}X = R\left(\frac{\partial f}{\partial s}, \frac{\partial f}{\partial t}\right)X,$$
$$\langle R(X,Y)Z,\ T\rangle = \langle R(Z,Y)X, Y\rangle.$$

3.2.2 Minimum Acceleration Path in SE(3)

Following the theory developed in [120], the following acceleration function along the path of the robot end-effector in SE(3) is minimized within the proposed framework

$$J = \int_{t_a}^{t_b} \langle \nabla_V \boldsymbol{V}, \nabla_V \boldsymbol{V}\rangle \,\mathrm{d}t, \tag{3.2}$$

where $[t_a, t_b]$ is the time interval over which the trajectory is planned, $\boldsymbol{V} = (\boldsymbol{\omega}_P, \boldsymbol{v}_P) \in \mathfrak{se}(3)$ is the velocity of the end-effector along a particular path, and ∇ denotes the Levi-Civita affine connection derived from a particular choice of metric on SE(3). This latter object allows the differentiation along curves on any smooth manifold. In particular, the inner product of the acceleration of a particular path with itself is expressed in (3.2): this may also be identified by the squared norm of the acceleration of this path at a particular point. Choosing the metric on SE(3) as

$$\boldsymbol{W} = \begin{bmatrix} \alpha \boldsymbol{I}_{3\times 3} & \boldsymbol{0} \\ \boldsymbol{0} & \beta \boldsymbol{I}_{3\times 3} \end{bmatrix},$$

where $\alpha, \beta > 0$ so that for $\boldsymbol{T}_1, \boldsymbol{T}_2 \in \mathfrak{se}(3)$ then $\langle\!\langle \boldsymbol{T}_1, \boldsymbol{T}_2 \rangle\!\rangle = \boldsymbol{t}_1^T \boldsymbol{W} \boldsymbol{t}_2$ with $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ the 6×1 components of $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, the Levi-Civita connection can be expressed by

$$\nabla_X Y = \left\{ \frac{\mathrm{d}}{\mathrm{d}t}\omega_y + \frac{1}{2}\omega_x \times \omega_y, \frac{\mathrm{d}v_y}{\mathrm{d}t} + \omega_x \times v_y \right\},$$

where ω_x and ω_y are the angular components and v_x and v_y are the linear components of the rigid body velocities $X \in \mathfrak{se}(3)$ and $Y \in \mathfrak{se}(3)$, respectively.

In order to have necessary conditions to find a path minimizing the acceleration function, the first variation of such a cost (3.2) has to be equated to zero. This yields a fourth order boundary problem given by

$$\nabla_V \nabla_V \nabla_V \boldsymbol{V} + R\left(\boldsymbol{V}, \nabla_V \boldsymbol{V}\right) \boldsymbol{V} = \boldsymbol{0}, \tag{3.3}$$

where R is the curvature tensor associated with the Levi-Civita affine connection [31]. Notice that it is possible to write down (3.3) in terms of the angular and linear velocity components of the end-effector as

$$\boldsymbol{\omega}_P^{(3)} + \boldsymbol{\omega}_P \times \ddot{\boldsymbol{\omega}}_P = \boldsymbol{0}, \tag{3.4a}$$

$$\boldsymbol{p}_P^{(4)} = \boldsymbol{0}, \tag{3.4b}$$

where $(\cdot)^{(n)}$ denotes the n^{th} derivative of $(\cdot)$. The obtained ordinary differential equations (3.4) turn into a well-defined boundary value problem with the addition of the boundary conditions. Regarding the rotational path (3.4a), such boundary conditions are

$$\boldsymbol{R}_P(t_a) = \boldsymbol{R}_P^0 \quad \boldsymbol{\omega}_P(t_a) = \boldsymbol{\omega}_P^0, \tag{3.5a}$$

$$\boldsymbol{R}_P(t_b) = \boldsymbol{R}_P^i, \quad \boldsymbol{\omega}_P(t_b) = \boldsymbol{\omega}_P^i, \tag{3.5b}$$

where $\boldsymbol{R}_P^0$ and $\boldsymbol{\omega}_P^0$ are the initial orientation and angular velocity of the end-effector, respectively. On the other hand, regarding the translational path (3.4b), the boundary conditions are

$$\boldsymbol{p}_P(t_a) = \boldsymbol{p}_P^0 \quad \boldsymbol{v}_P(t_a) = \boldsymbol{v}_P^0, \tag{3.6a}$$

$$\boldsymbol{p}_P(t_b) = \boldsymbol{p}_P^i \quad \boldsymbol{v}_P(t_b) = \boldsymbol{v}_P^i, \tag{3.6b}$$

Notice that in the practice, the optimal translational motion of the end-effector is found by merely solving a small scale linear system of equations obtained by (3.4b) and (3.6): this may be performed very fast from an elaboration time point of view. Nevertheless, in order to determine the rotary motion of the end-effector, a boundary value problem needs to be solved. In general, a boundary value problem is nonlinear and time invariant, but the forcing function (3.4a) is hardly complicated.

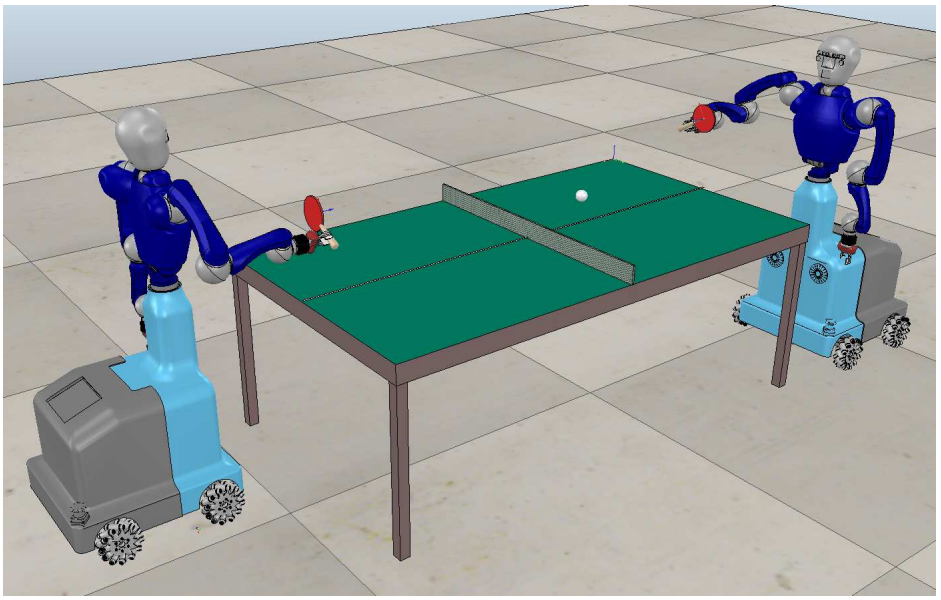


Figure 3.1: Two semi-humanoids playing table tennis in the V-REP simulation platform.

3.3 Optimal Planner for the Batting Task

The goal of the work presented in this Section is to implement the batting primitive on a semi-humanoid. The robot end-effector is equipped with a paddle to intercept a flying ball, as in [107, 108]. A picture of two semi-humanoids exploiting the batting primitive to play table tennis, in the V-REP graphic simulation platform, is shown in Fig. 3.1.

3.3.1 Hybrid Dynamic Modeling

The hybrid dynamics of the ball consists of the free flight aerodynamics and the impact reset map. The former is modeled through Newton's equations of motion, while the latter is a reset of the state, updated according to the impact detection. In order to analytically model the ball dynamics, the work [65] is considered. That work is supported by simulations and experiments in several case studies and follows up a deep study on the hybrid dynamic modeling of the table tennis game, [80]. On the other hand, a first-order dynamics for the paddle is here introduced, assuming that it is possible to directly control its velocity.

In the derivation of the hybrid dynamic model of this system, the following assumptions are made:

- as long as the paddle is made of rubber, the rebound in the direction normal to the paddle's plane does not affect the motion of the ball in the other directions;
- point contact occurs between the ball and the paddle during the impact;
- since the mass of the paddle is usually bigger than the mass of the ball, only the velocity of the ball is considered to be affected by the impact.

According to Fig. 3.2, let Σ_W be the fixed world frame, Σ_P be the frame placed at the center of the paddle, where the z-axis is the outward normal, and Σ_B be the frame placed at the center of the ball. Let $\boldsymbol{p}_B = \begin{bmatrix} p_{Bx} & p_{By} & p_{Bz} \end{bmatrix}^T \in \mathbb{R}^3$ be the position of the ball, $\boldsymbol{v}_B = \begin{bmatrix} v_{Bx} & v_{By} & v_{Bz} \end{bmatrix}^T \in \mathbb{R}^3$ be the velocity of the ball, $\boldsymbol{\omega}_B = \begin{bmatrix} \omega_{Bx} & \omega_{By} & \omega_{Bz} \end{bmatrix}^T \in \mathbb{R}^3$ be the spin of the ball, assumed constant during the free flight, $\boldsymbol{p}_P = \begin{bmatrix} p_{Px} & p_{Py} & p_{Pz} \end{bmatrix}^T \in \mathbb{R}^3$ be the position of the paddle, $\boldsymbol{v}_P = \begin{bmatrix} v_{Px} & v_{Py} & v_{Pz} \end{bmatrix}^T \in \mathbb{R}^3$ be the velocity of the paddle, $\boldsymbol{\omega}_P = \begin{bmatrix} \omega_{Px} & \omega_{Py} & \omega_{Pz} \end{bmatrix}^T \in \mathbb{R}^3$ be the angular velocity of the paddle, all expressed in Σ_W. Finally, let $\boldsymbol{R}_P \in SO(3)$ be the rotation matrix of Σ_P with respect to Σ_W.

The continuous ball and paddle dynamics are given by

$$\dot{\boldsymbol{p}}_B = \boldsymbol{v}_B, \tag{3.7a}$$
$$\dot{\boldsymbol{v}}_B = -\boldsymbol{g} - k_d||\boldsymbol{v}_B||\boldsymbol{v}_B + k_l \boldsymbol{S}(\boldsymbol{\omega}_B)\boldsymbol{v}_B, \tag{3.7b}$$
$$\dot{\boldsymbol{p}}_P = \boldsymbol{v}_P, \tag{3.7c}$$
$$\dot{\boldsymbol{R}}_P = \boldsymbol{R}_P \boldsymbol{S}(\boldsymbol{\omega}_P), \tag{3.7d}$$

where $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ is the gravity acceleration, $||\cdot||$ is the Euclidean norm, $\boldsymbol{S}(\cdot) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix operator. k_d and k_l are drag and lift parameters, respectively, and they are typically modelled as

$$k_d = k_d(\boldsymbol{v}_B, \boldsymbol{\omega}_B) = \frac{\rho\pi r^2(a_d + b_d\nu(\boldsymbol{v}_B, \boldsymbol{\omega}_B))}{2m}, \tag{3.8a}$$
$$k_l = k_l(\boldsymbol{v}_B, \boldsymbol{\omega}_B) = \frac{\rho 4\pi r^3(a_l + b_l\nu(\boldsymbol{v}_B, \boldsymbol{\omega}_B))}{m}, \tag{3.8b}$$

with

$$\nu(\boldsymbol{v}_B, \boldsymbol{\omega}_B) = \frac{1}{\sqrt{1 + \frac{(v_{Bx}^2 + v_{Bx}^2)\omega_{Bz}^2}{(v_{Bx}\omega_{By} - v_{By}\omega_{Bx})^2}}}. \tag{3.9}$$

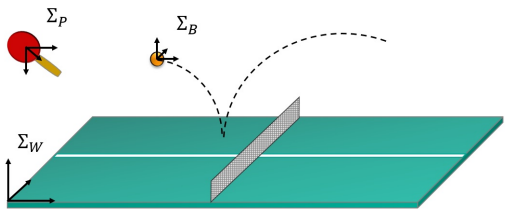


Figure 3.2: Ball and paddle coordinate systems.

The meaning of other parameters, like ρ, r, a_d, a_l, b_d, b_l, and their numerical values employed in simulations are depicted in Section 4.4, Table 3.2, where a standard table tennis ball and a rubber paddle have been considered. Since the paddle can modify the ball velocity only at the impact time, the control action, represented by the paddle linear and angular velocities, enters the ball dynamics through the reset map. Assuming that the superscripts $-$ and $+$ represent the state before and after the impact time, respectively, the rebound equations are given by [65]

$$\boldsymbol{v}_B^+ = \boldsymbol{v}_P + \boldsymbol{R}_P \boldsymbol{A}_{vv} \boldsymbol{R}_P^T (\boldsymbol{v}_B^- - \boldsymbol{v}_P) + \boldsymbol{R}_P \boldsymbol{A}_{v\omega} \boldsymbol{R}_P^T \boldsymbol{\omega}_B^-, \tag{3.10a}$$

$$\boldsymbol{\omega}_B^+ = \boldsymbol{R}_P \boldsymbol{A}_{\omega v} \boldsymbol{R}_P^T (\boldsymbol{v}_B^- - \boldsymbol{v}_P) + \boldsymbol{R}_P \boldsymbol{A}_{\omega\omega} \boldsymbol{R}_P^T \boldsymbol{\omega}_B^-, \tag{3.10b}$$

where the matrices of rebound coefficients are defined as

$$\begin{aligned} &\boldsymbol{A}_{vv} = \mathrm{diag}(1 - e_v, 1 - e_v, e_r), \boldsymbol{A}_{v\omega} = -e_v r \boldsymbol{S}(\boldsymbol{e}_3), \\ &\boldsymbol{A}_{\omega v} = e_\omega r \boldsymbol{S}(\boldsymbol{e}_3), \boldsymbol{A}_{\omega\omega} = \mathrm{diag}(1 - e_\omega r^2, 1 - e_\omega r^2, 1), \end{aligned} \tag{3.11}$$

where $\boldsymbol{e}_i \in \mathbb{R}^3$ is the unit vector along the i^{th}-axis, $i = \{1, 2, 3\}$, while e_v and e_w are described in the Table 3.2.

Notice that (3.7b) takes into account the spin of the ball: the magnitude of the drag and lift forces and their coefficients change according to the spin, influencing the trajectory of the ball. The values of the components of $\boldsymbol{\omega}_B$ determine the kind of the spin, namely:

- backspin, if $\omega_{By} > 0$;
- topspin, if $\omega_{By} < 0$;
- sidespin, if $\omega_{Bz} > 0$.

The effect of the spin of the ball in a table tennis game is not negligible and makes a difference between a serious table tennis player and a novice one. Serious players use spin on both their serves and rallying shots to control the ball and to force errors from their opponents.

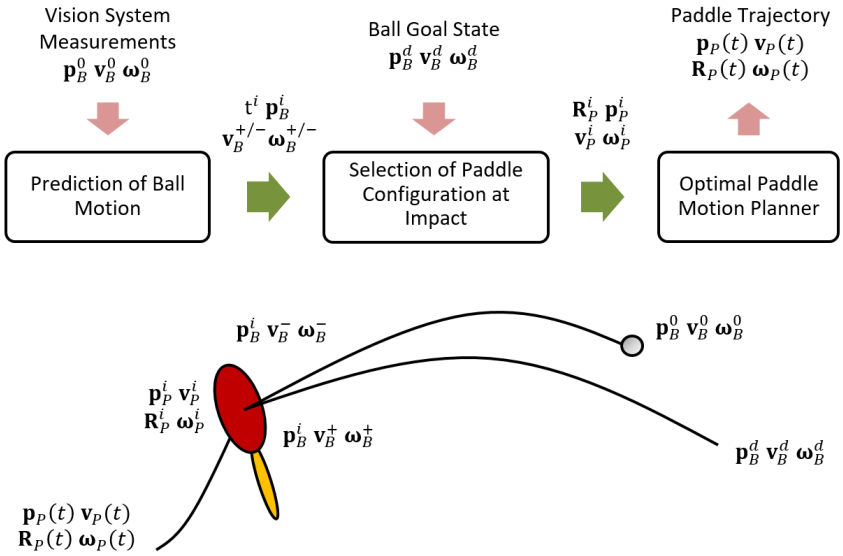


Figure 3.3: Graphic representation of the stages of the batting algorithm.

3.3.2 Time-Optimal Prediction

Workflow of the Algorithm

In order to generate the optimal motion for the paddle to bat a table tennis ball towards a desired position with a desired spin, the paddle has to intercept the ball with a specific orientation and velocity. The algorithm to realize such batting motion primitive can be roughly divided in three main phases: the prediction of the motion of the ball, the selection of the configuration of the paddle at impact time, and the trajectory planning for the paddle. A graphical representation of the phases generating the optimal path for the paddle is showed in Fig. 3.3.

1. In the first stage, the impact time t^i, the position at impact time $\boldsymbol{p}_B^i = \begin{bmatrix} p_{Bx}^i & p_{By}^i & p_{Bz}^i \end{bmatrix}^T \in \mathbb{R}^3$, and the pre-impact velocity $\boldsymbol{v}_B^-$ of the ball are predicted assuming to know the position of the ball at initial time $\boldsymbol{p}_B^0 = \begin{bmatrix} p_{Bx}^0 & p_{By}^0 & p_{Bz}^0 \end{bmatrix}^T \in \mathbb{R}^3$ and its linear and angular velocity, respectively, $\boldsymbol{v}_B^0 = \begin{bmatrix} v_{Bx}^0 & v_{By}^0 & v_{Bz}^0 \end{bmatrix}^T \in \mathbb{R}^3$ and $\boldsymbol{\omega}_B^0 = \begin{bmatrix} \omega_{Bx}^0 & \omega_{By}^0 & \omega_{Bz}^0 \end{bmatrix}^T \in \mathbb{R}^3$. These are produced by the opponent's hit and obtained from the visual

measurement system. This step is accomplished by solving forward the model (3.7a)-(3.7b). Afterwards, choosing the goal configuration of the ball given by $\boldsymbol{p}_B^d = \begin{bmatrix} p_{Bx}^d & p_{By}^d & p_{Bz}^d \end{bmatrix}^T \in \mathbb{R}^3$, $\boldsymbol{\omega}_B^{+d} \in \mathbb{R}^3$, and the desired final time t^d, the post-impact velocity of the ball $\boldsymbol{v}_B^+ \in \mathbb{R}^3$ is obtained through the backward solution of (3.7a)-(3.7b).

2. In the second stage, once the spin and velocity of the ball before and after the impact are computed from the previous stage, the algorithm selects the orientation $\boldsymbol{R}_P^i$ and the velocity $\boldsymbol{v}_P^i$ of the paddle at impact time through the solution of the reset map (3.10).

3. In the third stage, the optimal trajectory planner for the paddle, based on the derivation presented in Section 3.2, receives as input the desired configuration of the paddle at impact time. A boundary value problem is solved to compute the linear and angular trajectories of the paddle without specifying any representation of the angular coordinates. At this point, the linear and angular path of the paddle, namely $\boldsymbol{p}_P(t), \boldsymbol{v}_P(t), \boldsymbol{R}_P(t), \boldsymbol{\omega}_P(t)$, can be tracked by the end-effector of the semi-humanoid robot with a classical second order closed loop kinematic inversion [110].

Notice that the approaches presented in [65, 79], which have inspired the first and the second stages of this algorithm, assume instead that the impact position and pre-impact velocity of the ball are a priori known. Such assumptions are quite restrictive: hence, here the determination of such ball impact configuration is addressed at run-time. In addition, it is remarkable that one of the differences between the approaches presented by the works [107] and [108] is related to the first stage. In fact, the impact time is not a priori defined in [107], as instead in [108], but it is online predicted. In the following, the accurate description of the first two stages will be carried out. As stated above, the third stage is detailed in Section 3.2.

Stage 1: Prediction of the Impacting Time, Position and Velocities of the Ball

In order to predict the impact time, the position of the ball, and its linear pre- and post-impact velocities such that it reaches the desired location after the batting action, the equations (3.7a) and (3.7b) of the aerodynamic model are employed. However, this model is nonlinear and coupled, thus an analytic solution does not exist, and this complicates the estimation of the state at a certain time. Other approaches, such as in [2] and in [77], employ linearized or simplified models with the aim to cut down the elaboration time. For example, the following simplified

model of (3.7a)-(3.7b) is employed in [65]

$$\dot{v}_{Bx} = -\zeta^d |v_{Bx}| v_{Bx}, \qquad \dot{p}_{Bx} = v_{Bx}, \tag{3.12a}$$

$$\dot{v}_{By} = -\zeta^d |v_{By}| v_{By}, \qquad \dot{p}_{By} = v_{By}, \tag{3.12b}$$

$$\dot{v}_{Bz} = -g, \qquad \dot{p}_{Bz} = v_{Bz}, \tag{3.12c}$$

where $\zeta^d = \frac{\rho}{2m}\pi r^2 k^d$, with k^d considered as a suitable constant coefficient. Even though it is nonlinear, its analytic solution can be easily computed. The approach here proposed goes, instead, in a different direction for online state estimation: it exploits a proper numerical solver suitable for real-time processing.

Three different optimization problems are considered to predict the impact time and position of the ball, as well as its velocity before the rebound. The general methodology consists in the solution of nonlinear curve fitting problems. Afterwards, the second step of this stage consists in another optimization problem designed to compute the post-impact velocity of the ball.

Prediction with a Predefined Impact Time A first approach to predict the impact position and the pre-impact velocity of the ball is showed in [108]. By assigning the impact time, the impact configuration of the ball is predicted according to its initial position and velocity produced by the opponent's hit. The following minimization problem is then considered

$$\min_{\boldsymbol{p}_B^i, \boldsymbol{v}_B^-} \left|\left| \begin{bmatrix} \widetilde{\boldsymbol{p}}_B^0 \\ \widetilde{\boldsymbol{v}}_B^0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_B^0 \\ \boldsymbol{v}_B^0 \end{bmatrix} \right|\right|^2, \tag{3.13}$$

where $\boldsymbol{p}_B^i$ and $\boldsymbol{v}_B^-$ are the optimizing variables, $\widetilde{\boldsymbol{p}}_B^0 = \widetilde{\boldsymbol{p}}_B^0(\boldsymbol{p}_B^i, \boldsymbol{v}_B^-)$ and $\widetilde{\boldsymbol{v}}_B^0 = \widetilde{\boldsymbol{v}}_B^0(\boldsymbol{p}_B^i, \boldsymbol{v}_B^-)$ are the position and the velocity of the ball at the initial time, respectively, numerically obtained by backward integrating (3.7a) and (3.7b) starting from the optimization variables $\boldsymbol{p}_B^i, \boldsymbol{v}_B^-$ at the predefined impact time t^i.

Prediction of a Variable Impact Time In a second possible optimization problem, the impact time is predicted as well as the impact position and the pre-impact velocity of the ball. Starting from the initial state of the ball, the following minimization problem is equivalent to (3.13) but it includes the impact time as a decision variable

$$\min_{t^i, \boldsymbol{p}_B^i, \boldsymbol{v}_B^-} \left|\left| \begin{bmatrix} \widetilde{\boldsymbol{p}}_B^0 \\ \widetilde{\boldsymbol{v}}_B^0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_B^0 \\ \boldsymbol{v}_B^0 \end{bmatrix} \right|\right|^2. \tag{3.14}$$

In this case, the resulting impact time is optimized in the sense that it best fits the numerical curve.

Prediction of a Variable Impact Time Minimizing the Paddle Motion A third approach, consists in the computation of the impact time, as well as the impact position and the pre-impact velocity of the ball, by minimizing the distance between the initial position of the paddle and its impact position. The following minimization problem is thus considered

$$\min_{t^i, \boldsymbol{p}_B^i, \boldsymbol{v}_B^-} ||\widetilde{\boldsymbol{p}}_B^i - \boldsymbol{p}_P^0||^2, \tag{3.15}$$

where $\widetilde{\boldsymbol{p}}_B^i = \widetilde{\boldsymbol{p}}_B^i(\boldsymbol{p}_B^0, \boldsymbol{v}_B^0)$ is the position of the ball at impact time, numerically obtained by forward integrating (3.7a) and (3.7b) starting from the optimization variables $\boldsymbol{p}_B^0, \boldsymbol{v}_B^0$ at time t^0. In this case, the resulting impact time is optimized in the sense that the paddle travels the shortest distance to intercept the ball. Therefore, the introduction of this metric in the prediction stage allows not only to predict the position of the ball and its pre-impact velocity but even, more interestingly, to optimize the impact time with respect to the length of the path that the paddle should cover.

Prediction of the Post-impact Velocity of the Ball Moreover, the second step of this stage consists in another optimization problem to solve to compute the post-impact velocity of the ball $\boldsymbol{v}_B^+$ such that it reaches the goal $\boldsymbol{p}_B^d$ at the desired time t^d. The following minimization is solved

$$\min_{\boldsymbol{v}_B^+} \|\widetilde{\boldsymbol{p}}_B^d(\boldsymbol{v}_B^+) - \boldsymbol{p}_B^d\|^2, \tag{3.16}$$

where $\boldsymbol{v}_B^+$ is the decision variable, and $\widetilde{\boldsymbol{p}}_B^d(\boldsymbol{v}_B^+)$ is the position of the ball at time t^d, numerically obtained by forward integrating (3.7a) and (3.7b) starting from the ball position $\boldsymbol{p}_B^i$ at impact time t^i, computed through one of the metrics given above - corresponding to the minimization problems (3.13), (3.14), or (3.15). This solution ensures that the post-impact motion of the ball is such that it reaches the desired location at the time t^d as close as possible. An initial guess for the solution of the minimization problem (3.16) is analytically calculated from (3.12) as

$$v_{Bx}^+ = \frac{(p_{Bx}^d - p_{Bx}^i)(e^{\zeta^d |p_{Bx}^d - p_{Bx}^i|} - 1)}{\zeta^d |p_{Bx}^d - p_{Bx}^i| (t^d - t^i)}, \tag{3.17a}$$

$$v_{By}^+ = \frac{(p_{By}^d - p_{By}^i)(e^{\zeta^d |p_{By}^d - p_{By}^i|} - 1)}{\zeta^d |p_{By}^d - p_{By}^i| (t^d - t^i)}, \tag{3.17b}$$

$$v_{Bz}^+ = -\frac{g(t^d - t^i)}{2} + \frac{p_{Bz}^d - p_{Bz}^i}{t^d - t^i}. \tag{3.17c}$$

Table 3.1: List of symbols for the hybrid model of impact manipulation.

r	Radius of the ball
r_p	Radius of the paddle
m	Mass of the ball
ρ	Density of the air at $(25^\circ C)$
g	Gravity constant
ε_v	Velocity rebound coefficient
ε_ω	Spin rebound coefficient
ε_r	Linear rebound coefficient
κ_1^d	Drag coefficient
κ_2^d	Drag coefficient
κ_1^l	Lift coefficient
κ_2^l	Lift coefficient
ζ^d	Simplified drag coefficient

Stage 2: Desired Configuration of the Paddle at Impact

Once the impact time t^i, the impact position $\boldsymbol{p}_B^i$ of the ball and its pre- and post-impact velocities, $\boldsymbol{v}_B^-$ and $\boldsymbol{v}_B^+$ respectively, are computed as in Section 3.3.2, the paddle configuration is derived solving the rebound model of the ball.

Consider the YX-Euler angles (θ, ϕ) as a parametric representation of the orientation of the paddle, with $\phi \in [-\pi/2, \pi/2]$ and $\theta \in [0, \pi]$, and define $\tilde{\boldsymbol{v}} = \begin{bmatrix}\tilde{v}_x & \tilde{v}_y & \tilde{v}_z\end{bmatrix}^T = \boldsymbol{v}_B^+ - \boldsymbol{v}_B^-$ and $\tilde{\boldsymbol{\omega}} = \begin{bmatrix}\tilde{\omega}_x & \tilde{\omega}_y & \tilde{\omega}_z\end{bmatrix}^T = \boldsymbol{\omega}_B^{+d} - \boldsymbol{\omega}_B^-$. The velocity and orientation of the paddle at impact time are respectively computed through

$$\boldsymbol{v}_P^i = \boldsymbol{v}_B^- + \boldsymbol{R}_P^i(\boldsymbol{I}_{3\times 3} - \boldsymbol{A}_{vv})^{-1}(\boldsymbol{R}_P^{iT}\tilde{\boldsymbol{v}} - \boldsymbol{A}_{v\omega}\boldsymbol{\omega}_B^-), \tag{3.18a}$$

$$\boldsymbol{R}_P^i = \boldsymbol{R}_Y(\theta)\boldsymbol{R}_X(\phi), \tag{3.18b}$$

where $\boldsymbol{I}_{3\times 3} \in \mathbb{R}^{3\times 3}$ is the identity matrix, $\boldsymbol{R}_i(\cdot) \in SO(3)$ is the elementary rotation matrix with $i = \{X, Y\}$, representing the rotation of an angle around the i-axis, and θ, ϕ are such that

$$\tilde{v}_z \cos\phi \sin\theta - \tilde{v}_x \cos\phi \cos\theta = \tilde{\omega}_y, \tag{3.19a}$$

$$e_c^2||\tilde{\boldsymbol{v}}||^2 \sin^2\phi - 2e_c\boldsymbol{e}_2 S(\tilde{\boldsymbol{v}})\tilde{\boldsymbol{\omega}}\sin\phi + (\boldsymbol{e}_1 + \boldsymbol{e}_3)||\tilde{\boldsymbol{\omega}}||^2 - e_c\boldsymbol{e}_2||\tilde{\boldsymbol{v}}||^2 = 0, \tag{3.19b}$$

where $e_c = \varepsilon_\omega r/\varepsilon_v$, and the other symbols are listed in Table 3.1. In order to obtain a well-posed solution of the reset map, the ball motions must comply with the Proposition 1 in Section III-A of [65].

Table 3.2: Numerical values of the considered parameters of the hybrid dynamic model.

r	2e-2 m
r_p	1.5e-1 m
m	2.7e-3 kg
ρ	1.184 kg/m^3
g	9.81 m/s^2

κ_1^d	5.05e-1
κ_2^d	6.5e-2
κ_1^l	9.4e-2
κ_2^l	-2.6e-2

ε_v	6.15e-1
ε_ω	2.57e3
ε_r	7.3e-1
ζ^d	5.4e-1

3.3.3 Batting Task Simulations

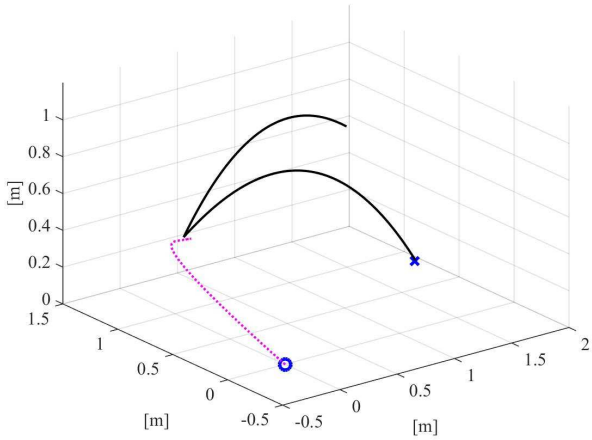
This Section presents a discussion about the numerical evaluation of the batting algorithm. The three metrics introduced in the previous Section in (3.13), (3.14), and (3.15) are taken into account. In detail, an exemplar simulation of the proposed technique with a predefined impact time corresponding to (3.13) is firstly showed. Subsequently, the approach is evaluated with the prediction of a variable impact time using the last two metrics (3.14)-(3.15). Finally, some tests focused on the paddle motion are presented to underline the properties of the planned trajectories.

The values of the parameters of the dynamic model considered to simulate the physical system (3.7) and (3.10) are listed in Table 3.2. The Matlab environment is used for numerical tests. For the hybrid dynamics, the *ode45* solver, with the *events* option, is employed. The *lsqcurvefit* function, which is based on the Levembert-Marquardt's algorithm, is adopted to solve the nonlinear curve fittings. The boundary value problem, for the minimum acceleration planner, is solved through the *bvp4c* function. In this work, the Levembert-Marquardt's algorithm is employed to solve the non-linear least squares problems (3.13), (3.14), and (3.15) since it guarantees small elaboration time, as explained in [63] and [26]. In order to speed up the convergence of the optimization problems, the initial guess for the solution is obtained solving analytically the approximated model (3.17).

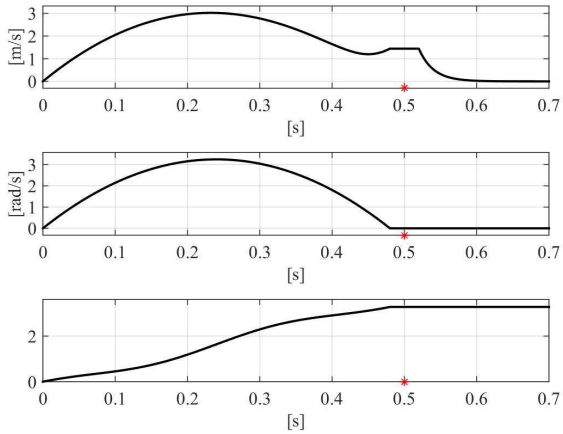
Evaluation of Problem (3.13): Predefined Impact Time

In this case study the proposed algorithm, depicted in Fig. 3.3, is simulated considering a constant predefined impact time. Then, supposing to have at disposition the estimated trajectory of the ball from the visual system and the desired final configuration of the ball, the optimal paddle trajectory is derived through the two minimization problems (3.13) and (3.16), and the solution of (3.18) and (3.4).

In particular, the initial state of the ball is assumed to be equal to $\boldsymbol{p}_B^0 =$



(a) 3D trajectories of the ball and the paddle (respectively solid and dashed line) obtained from the predefined impact time method. The blue circle represents the initial position of the paddle, while the blue cross identifies the goal position of the ball.



(b) Euclidean norm of the linear (top) and angular (center) velocity paths planned for the paddle, and evaluation of the acceleration functional J in (3.2) between the motion plan devised using the Euler angles and the optimal proposed one (bottom). The red star represents the impact time t^i.

Figure 3.4: Simulation of the batting task with predefined impact time.

$\begin{bmatrix}1.2, & 0.7, & 0.9\end{bmatrix}$ m, $\boldsymbol{v}_B^0 = \begin{bmatrix}-3, & 0.2, & 1.5\end{bmatrix}$ m/s and $\boldsymbol{\omega}_B^- = \begin{bmatrix}0, 150, 0\end{bmatrix}$. The impact time is fixed to $t^i = 0.5$ s and the goal position of the ball after the rebound is assigned as $\boldsymbol{p}_B^d = \begin{bmatrix}1.9, & 0.8, & 0.02\end{bmatrix}$ m. The desired final time corresponds to $t^d = 0.6$ s, and the post-impact spin of the ball to $\begin{bmatrix}\boldsymbol{\omega}_{By}^+, \boldsymbol{\omega}_{Bz}^+\end{bmatrix} = \begin{bmatrix}-100, 0\end{bmatrix} rad/s$. The term $\boldsymbol{\omega}_B^{+d}$ derives from the expression $\tilde{\boldsymbol{v}}_B^T \tilde{\boldsymbol{\omega}}_B = 0$. Notice that the actual final time is evaluated when the third component of the position vector of the ball corresponds to the radius of the ball.

Solving (3.13) yields the values $\boldsymbol{p}_B^i = \begin{bmatrix}-0.1394, & 0.7892, & 0.4820\end{bmatrix}$ m and $\boldsymbol{v}_B^- = \begin{bmatrix}-2.4156, & 0.1570, & -2.9788\end{bmatrix}$ m/s, while the solution of the minimization (3.16) yields $\boldsymbol{v}_B^+ = \begin{bmatrix}4.0516, & 0.0214, & 2.0984\end{bmatrix}$ m/s. Furthermore, the solution of the rebound model (3.18) yields

$$\boldsymbol{R}_p^i = \begin{bmatrix} 0.8614 & 0.0054 & 0.5080 \\ 0 & 0.9999 & -0.0106 \\ -0.5080 & 0.0092 & 0.8613 \end{bmatrix},$$

and $\boldsymbol{v}_P^i = \begin{bmatrix}1.4388, & 0.0220, & -0.1131\end{bmatrix}$ m/s. Afterwards, the minimum acceleration trajectory for the paddle is planned employing (3.4).

Let Δt_i be the error between the predefined impact time and the one obtained during the simulation through the *ode45* solver together with the events option. Let $\Delta \boldsymbol{p}_B^i$ be the Euclidean norm of the error between the planned and actual impact position of the ball, Δt_d be the error between the predefined and actual final time t^d, $\Delta \boldsymbol{p}_B^d$ be the Euclidean norm of the error between the goal position of the ball and the actual one. The first row of Table 3.3 shows these errors in case of a predefined impact time. Whereas, the resulting plots are depicted in Fig. 3.4. In particular, the 3D trajectories of both the ball and the paddle are represented in Fig. 3.4(a). The solid line represents the motion of the ball, while the trajectory for the paddle is depicted by a dashed line. The blue cross represents the final desired position of the ball $\boldsymbol{p}_B^d$, while the blue circle is the initial position of the paddle $\boldsymbol{p}_P^0$. A more detailed evaluation of the proposed batting algorithm with fixed impact time in case of sidespin, backspin, and topspin can be found in [108].

Moreover, a video[1] shows this simulation performed in Matlab in connection with the V-Rep virtual simulation environment. A 21 DoF semi-humanoid robot is used within this simulation. The 7 DoF right arm is equipped with a parallel jaw gripper firmly grasping the paddle (see Fig. 3.1). The rebound model of the ball with the table is given by [80], and a second order closed loop kinematic inversion is implemented to map the planned Cartesian variables to the joints space of the robot [110].

Table 3.3: Comparison of the fixed and optimized impact time prediction methods.

	Δt_i	$\Delta \boldsymbol{p}_B^i$	Δt_d	$\Delta \boldsymbol{p}_B^d$
Fixed Impact Time Prediction	4.5e-3 s	1.73e-2 m	1.9e-3 s	9.4e-3 m
Optimal Impact Time Prediction	4e-3 s	1.69e-2 m	1.1e-3 s	1.33e-2 m

Evaluation of Problems (3.14)**-**(3.15)**: Optimal Impact Time Prediction**

In this case study, the same initial configuration of the ball used in the previous Section is considered, and the proposed algorithm is evaluated including the online prediction of the impact time. The results obtained with the two prediction metrics (3.14)-(3.15) are here reported.

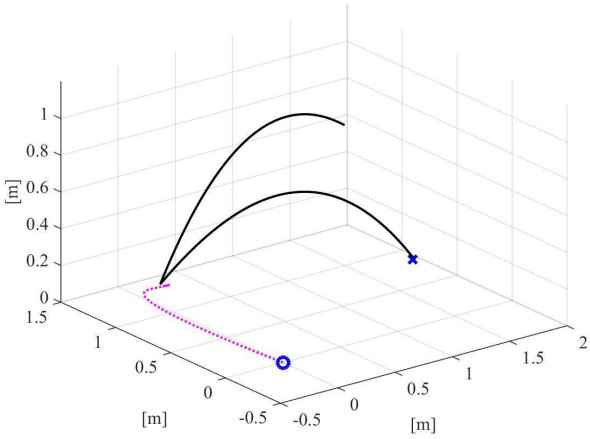
A first test is done considering the minimization problem (3.14). In this case, the prediction phase provides: t^i=0.5051 s, $\boldsymbol{p}_B^i = \begin{bmatrix} -0.1516, & 0.79, & 0.4668 \end{bmatrix}$ m, and $\boldsymbol{v}_B^- = \begin{bmatrix} -2.4105, & 0.1565, & -3.0174 \end{bmatrix}$ m/s. The obtained result is not substantially different from the one got from the simulation considering a pre-defined impact time. For this reason, figures for this case are not included.

Whereas, another test is done solving (3.15), (3.16), (3.18), and (3.4). In this case, the solution of (3.15) is $t^i = 0.5735$ s, $\boldsymbol{p}_B^i = \begin{bmatrix} -0.3141, & 0.8, & 0.243 \end{bmatrix}$ m, and $\boldsymbol{v}_B^- = \begin{bmatrix} -2.34, & 0.1498, & -3.521 \end{bmatrix}$ m/s. The predicted impact time guarantees that the paddle follows the minimum length path to intercept the ball. The plots resulting from this test are displayed in Fig. 3.5. The 3D paths of the ball and the paddle are reported in Fig. 3.5(a). The solid and dashed lines represent the motion of ball and paddle, respectively. The blue cross identifies the goal position of the ball, whereas the blue circle is the initial position of the paddle. The second row of Table 3.3 shows the values of Δt_i, $\Delta \boldsymbol{p}_B^i$, Δt_d, and $\Delta \boldsymbol{p}_B^d$ for this numerical test. Now, Δt_i represents the error between the impact time resulting from the minimization problem and the one obtained during the simulation. The results point out that, even if the position error is slightly increased with respect to the constant impact time prediction case, the impact time and position are planned online in [107], and the ball hits the table at a time closer to the desired one.

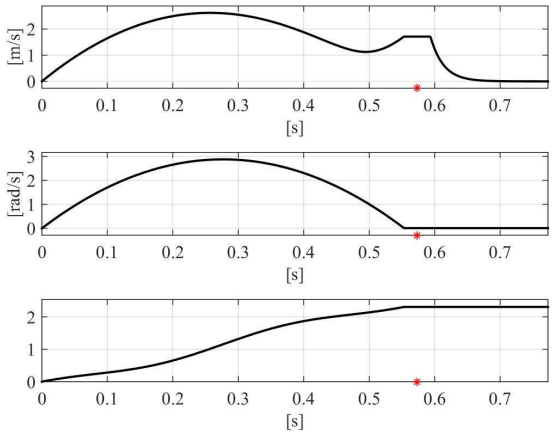
Comparative Case Studies for the Batting Task

The purpose of this subsection is to compare the planning method with a predefined impact time, here proposed, with the state-of-the-art approach introduced by [65], which presents several case studies, with ample implementation details,

[1] `https://youtu.be/GXtBvbUHu5s`

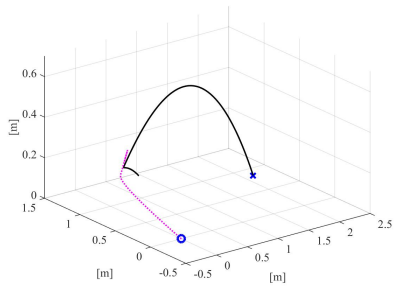


(a) 3D trajectories of the ball and the paddle (respectively solid and dashed line) obtained considering the optimal impact time method. The initial position of the paddle is represented by the blue circle, while the goal position of the ball is the blue cross.

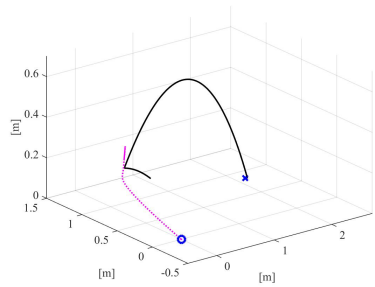


(b) Euclidean norm of the linear (top) and angular (center) velocity planned for the paddle, and evaluation of the acceleration functional J in (3.2) between the motion plan devised using the Euler angles and the optimal one (bottom). The red star represents the impact time t^i.

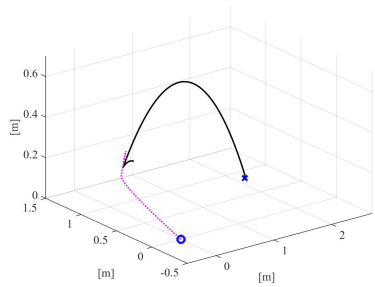
Figure 3.5: Simulation of the batting task with optimal impact time.



(a) Backsping case study.

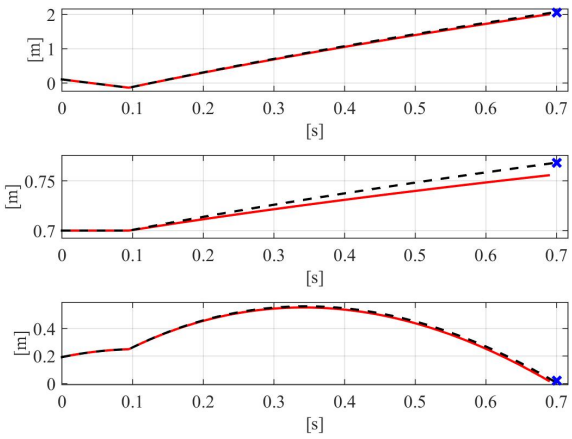


(b) Topspin case study.

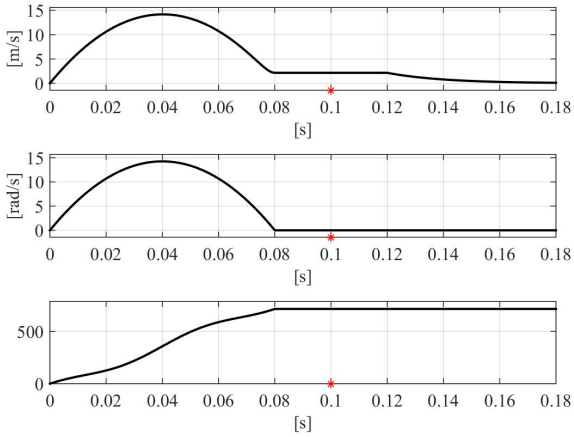


(c) Sidespin case study.

Figure 3.6: 3D trajectories of the ball, solid line, and the paddle, dashed line, obtained with the proposed method. The blue circle represents the initial position of the paddle, while the blue cross is the desired final position of the ball.

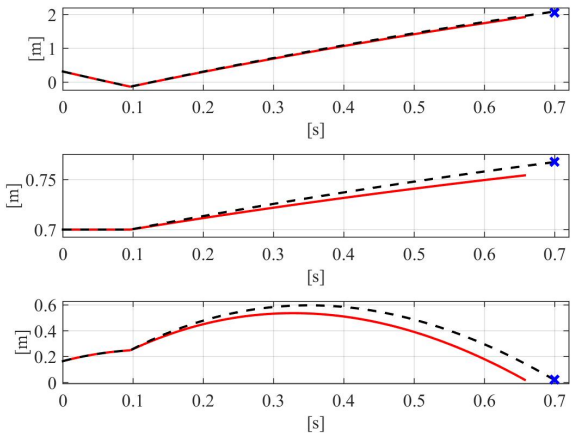


(a) Comparison between ball trajectories: the solution given by analytically solving (3.12) is depicted through a solid line, the proposed one is instead represented with a dashed line. The blue cross represents the desired final position of the ball.

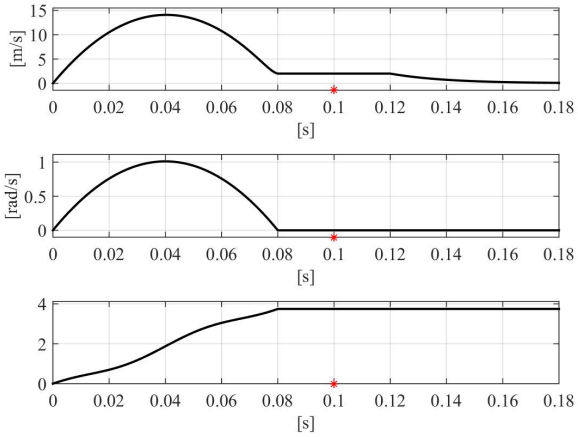


(b) From the top to the bottom: magnitude of the planned linear and angular velocity of the paddle, evaluation of the acceleration functional J in (3.2) between the motion plan devised using the Euler angles and the optimal proposed one. The red star represents the impact time t_i.

Figure 3.7: Comparative backspin first case study.

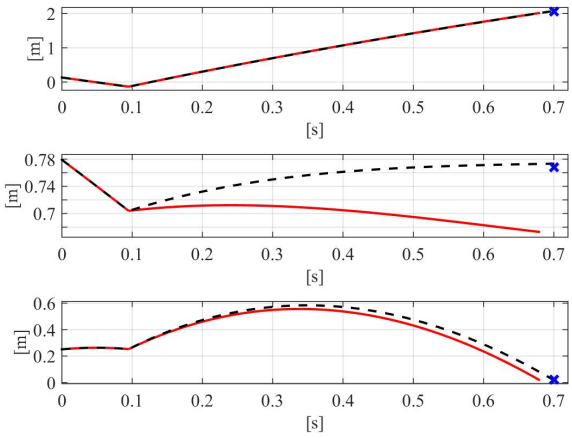


(a) Comparison between ball trajectories: the solution given by analytically solving (3.12) is depicted through a solid line, the proposed one is instead represented with a dashed line. The blue cross represents the desired final position of the ball.

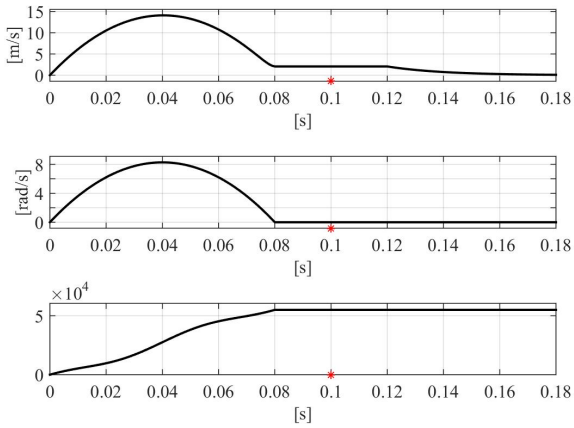


(b) From the top to the bottom: magnitude of the planned linear and angular velocity of the paddle, evaluation of the acceleration functional J in (3.2) between the motion plan devised using the Euler angles and the optimal proposed one. The red star represents the impact time t_i.

Figure 3.8: Comparative topspin second case study.



(a) Comparison between ball trajectories: the solution given by analytically solving (3.12) is depicted through a solid line, the proposed one is instead represented with a dashed line. The blue cross represents the desired final position of the ball.



(b) From the top to the bottom: magnitude of the planned linear and angular velocity of the paddle, evaluation of the acceleration functional J in (3.2) between the motion plan devised using the Euler angles and the optimal proposed one. The red star represents the impact time t_i.

Figure 3.9: Comparative sidespin third case study.

Table 3.4: Ball pre-impact configuration.

	$\boldsymbol{v}_B^-$ $[m/s]$	$\boldsymbol{\omega}_B^-$ $[rad/s]$
1	$[-2.5, 0, 0.1]$	$[0, 150, 0]$
2	$[-4.5, 0, 0.3]$	$[0, -150, 0]$
3	$[-2.75, -0.8, -0.5]$	$[0, -50, 150]$

Table 3.5: Comparative case study 1 - Numerical results.

	Δt_i $[s]$	$\Delta \boldsymbol{p}_B^i$ $[m]$	Δt_d $[s]$	$\Delta \boldsymbol{p}_B^d$ $[m]$	$t_c[ms]$
A	5.2e-3	1.3e-2	1.12e-2	5.47e-2	9e-3
B	5.14e-3	1.29e-2	4.88e-3	6.47e-4	21

Table 3.6: Comparative case study 2 - Numerical results.

	Δt_i $[s]$	$\Delta \boldsymbol{p}_B^i$ $[m]$	Δt_d $[s]$	$\Delta \boldsymbol{p}_B^d$ $[m]$	$t_c[ms]$
A	3.86e-3	1.74e-2	4.26e-2	1.3e-1	8e-3
B	4.01e-3	1.81e-2	7.34e-4	3.29e-2	21

Table 3.7: Comparative case study 3 - Numerical results.

	Δt_i $[s]$	$\Delta \boldsymbol{p}_B^i$ $[m]$	Δt_d $[s]$	$\Delta \boldsymbol{p}_B^d$ $[m]$	$t_c[ms]$
A	4.88e-3	1.42e-2	2.12e-2	1.03e-1	1e-2
B	4.86e-3	1.41e-2	9.02e-4	1.94e-2	25

allowing a fair and critical comparison. Three case studies are considered, respectively: backspin, topspin and sidespin. In order to obtain a fair comparison, to comply with the case studies in [65], the ball pre-impact state and the impact time are assumed to be known a-priori and then the minimization problems (3.13)-(3.14)-(3.15) are not solved in this case. As a matter of fact, in each of these cases, the impact position of both the ball and the paddle is assigned to be $\boldsymbol{p}_P^i = \boldsymbol{p}_B^i = \begin{bmatrix} -0.15 & 0.70 & 0.25 \end{bmatrix}$ m, while the linear and angular velocity of the ball before the impact for the backspin, topspin and sidespin case studies are shown in Table 3.4. On the other hand, the impact time is $t_i = 0.1$ s, the desired goal position for the ball is $\boldsymbol{p}_B^d = \begin{bmatrix} 2.055 & 0.7680 & 0.02 \end{bmatrix}$ m, while the desired final time is $t_d = 0.6$ s. Moreover, the desired spin of the ball after the impact $\begin{bmatrix} \boldsymbol{\omega}_{By}^+, \boldsymbol{\omega}_{Bz}^+ \end{bmatrix}$ is set for the first, second and third case studies to $\begin{bmatrix} -100, 0 \end{bmatrix}$ rad/s, $\begin{bmatrix} 100, 0 \end{bmatrix}$ rad/s and $\begin{bmatrix} 0, -100 \end{bmatrix}$ rad/s respectively, according to the setting of the work described in [65].

The results obtained for each of the three case studies are depicted by their time histories in Fig.s 3.7, 3.8 and 3.9, respectively. While the 3D trajectories of both the ball and the paddle are represented in Fig. 3.6, the solid line represents the ball, while the paddle is depicted by a dashed line. The time histories of each component p_{Bx}, p_{By} and p_{Bz} related to the trajectory of the ball are represented in Fig.s 3.7(a), 3.8(a) and 3.9(a), respectively, for both of the compared methods: the solid line graphs the solution obtained by exploiting the method described by [65], while the dashed curve illustrates the trajectory of the ball obtained using the proposed method. The figures indicate that method proposed in [108] yields an improvement over [65] by directing the ball closer to the goal configuration.

The quantitative results for the backspin, topspin and sidespin case studies are shown in Tables 3.5, 3.6 and 3.7, where A represents the results obtained employing the Liu's planner, whereas B the results obtained employing the planner here proposed. The first two columns of each of the aforementioned tables refer to the precision of the impact. In both the approaches the results are very similar. The gist of the comparison may be captured by analyzing the last three columns of these tables. The final time error is reduced by about an order of magnitude. The Euclidean norm of the error between the actual final position of the ball and the desired one is smaller by about an order of magnitude, too.

Unfortunately, experiments are not yet available for the proposed approach since the practical set-up is under development. However, the switching from Matlab into C++ language is already accomplished. The computational time t_c required to solve at run-time both the simplified aerodynamic model and the complete one is show in the last column of Tables 3.5, 3.6 and 3.7. The code is running on a computer with specifications Intel Core 2 Quad CPU Q6600 @ 2.4 GHz, Ubuntu 12.04 32-bit operating system, including the Levenberg-Marquardt C++ library [66]. The same numerical results obtained with Matlab have been retrieved, but the evaluation of the computation burden is more precise, in the sense that it is the one that will appear during the practical experiments. To elaborate, compared to [65], the presented method increases the accuracy of the final desired position an order of magnitude for topspin and sidespin cases and two orders of magnitude for the backspin case. Furthermore, the C++ implementation of the proposed nonlinear minimization problem, which considers the full aerodynamic model of the ball, takes about 20 ms to give the desired velocity of the ball after the impact. This duration is grater than what is shown in [65], but it is still acceptable for real-time implementation.

Optimal Planner Simulations

For each case study, the paddle trajectory is planned over the time interval $[t_a, t_b] = [t_0, t_i - \epsilon]$, where $\epsilon = 0.02$ s. The paddle trajectory is supposed to start still, from the origin of the world frame, with initial orientation $\boldsymbol{R}_P^0 =$

$\boldsymbol{R}_Y(\pi/2)\boldsymbol{R}_X(0)$, without loss of generality. According to the proposed algorithm, the position, orientation and linear velocity of the paddle at the impact time are given by the ball reset map solution. The Euclidean norm of the linear and angular velocities of the paddle, planned with the minimum total acceleration, are represented by the top and middle plots in Fig.s 3.4(b), 3.5(b), 3.7(b), 3.8(b), and 3.9(b), for each simulation. Once the desired orientation is achieved with zero angular velocity, the angular acceleration is set to zero so that the orientation of the paddle remains the same until the impact occurs. As long as one has the control authority at the torque level, this control strategy, which switches only once, is straightforward to implement. Once the impact has occurred at time t_i, notice that the linear velocity of the paddle is exponentially dissipated by the following term $\exp(-\mu(t-(t_i+\delta)))$, where t is the time, $\mu = 50$ and $\delta = 0.02$, so that the paddle stops. The optimal trajectory discovered by solving the two-point boundary value problem (3.4) indeed minimizes the L_2 norm of the total acceleration of the paddle. So as to illustrate this fact, another typical trajectory for the orientation of the paddle is planned. This alternative plan constructs a third-order polynomial function for the Euler angles, ϕ and θ, such that the initial and final orientation and angular velocity constraints are satisfied. Both the angular acceleration that corresponds to this motion plan and the acceleration functional J in (3.2) are then computed. For each case study, the bottom time histories of Fig.s 3.4(b), 3.5(b), 3.7(b), 3.8(b) and 3.9(b), depict the value of the difference of the acceleration functional between the motion plan devised using the Euler angles and the optimal motion plan. Notice that the value of this cost functional is positive at $t = t_i$, indicating that the optimal motion plan indeed yields a smaller value of the acceleration functional than a typical plan performed using polynomials on the Euler angles, ϕ and θ.

Computational Efficiency

About the computational burden of the proposed planner, the code has been translated in C++ and evaluated on the same PC detailed in the previous Section. The boundary value problem for the optimal paddle trajectory planner takes less than 30 ms. To sum up, after the high-speed vision system gives a stable trajectory estimation of the ball coming towards our court, it is possible to compute the desired trajectory for the paddle in 50 ms (20 ms + 30 ms), hitting the ball with a proper velocity to redirect it to the opposite court at the desired position with the imposed spin. Another possibility is that, once the desired impact position, velocity and orientation is determined, one can immediately start controlling the paddle to achieve these via a PD controller, and revert to a trajectory following controller once the optimal trajectory is available and is periodically updated.

3.4 Optimal Planner for the Ball Juggling Task

The juggling task is here intended as repetitive batting (throwing and catching in a single collision) of a ball between two paddles, or hands, in a nonprehensile way. Since juggling actions, in general, require high velocity and precision, the investigation of the related motion planning is useful to confer dexterity and powerful manipulation skills to the robotic system. Indeed, many works in the literature can be found about robotic ball juggling, as explained in Section 3.1. The aim of this Section is to exploit the single batting primitive to plan an optimal path for the dual-hand ball juggling task, demonstrating how a complex nonprehensile manipulation can be dealt with a bottom-up approach, from the single primitive to the complete task [106].

3.4.1 Modeling of Dual-Hand Ball Juggling

The dynamics of the ball is modeled using the equations (3.7) and (3.10), reported in Section 3.3.1. In the following, the paddle that in turn is going to catch the ball is referred to as *impacting* paddle, whereas the other one is referred to as *free*. They are indicated through the i and f subscripts, respectively.

Let Σ_W be the fixed world frame, and let Σ_{ip} and Σ_{fp} be the frames placed at the center of the impacting and free paddles, respectively. The z-axis is denoted as the outward normal to the surfaces of the paddles. Let $\boldsymbol{p}_{ip}, \boldsymbol{v}_{ip}, \boldsymbol{\omega}_{ip}, \boldsymbol{p}_{fp}, \boldsymbol{v}_{fp}, \boldsymbol{\omega}_{fp} \in \mathbb{R}^3$ be position, linear and angular velocities of the impacting and free paddles, respectively, all expressed in Σ_W. Finally, let $\boldsymbol{R}_{ip}, \boldsymbol{R}_{fp} \in SO(3)$ be the rotation matrices of Σ_{ip} and Σ_{fp} with respect to Σ_W, respectively.

The equations of motion for the free and impacting paddles are given by

$$\dot{\boldsymbol{p}}_{ip} = \boldsymbol{v}_{ip}, \tag{3.20a}$$

$$\dot{\boldsymbol{R}}_{ip} = \boldsymbol{R}_{ip}\boldsymbol{S}(\boldsymbol{\omega}_{ip}), \tag{3.20b}$$

$$\dot{\boldsymbol{p}}_{fp} = \boldsymbol{v}_{fp}, \tag{3.20c}$$

$$\dot{\boldsymbol{R}}_{fp} = \boldsymbol{R}_{fp}\boldsymbol{S}(\boldsymbol{\omega}_{fp}). \tag{3.20d}$$

Notice that the ball is acted only at each impact time by means of the linear and angular velocities of the impacting paddle, which enters the ball dynamics through the reset map.

3.4.2 Algorithm for Dual-Hand Ball Juggling

In order to accomplish the desired task, the paddles must repetitively intercept the ball in turn. It is assumed that the algorithm receives as input the measure of the state of the ball (i.e., through a visual system). This aspect is out of the

scope of this work. Hence, the goal of the proposed algorithm is to compute the orientation and the linear velocity of the impacting paddle so as to re-direct the ball towards the free paddle, similarly to [108]. In order to solve this task, the algorithm has to know a-priori a desired location where the ball has to be re-directed after the collision with the paddle, and the time interval Δt between two consecutive impacts. Such predetermined locations and Δt are input values shaping the trajectory of the ball within the juggling task. They should be thus tuned according to the available robot capabilities (i.e., maximum joint velocities) and the reachable workspace. After each collision, the algorithm swaps the free and impacting paddles.

Fig. 3.10 schematically resumes the solution proposed in [106]. In particular, after that the visual system provides the configuration of the ball (i.e., the initial state following the previous impact), solving two nonlinear minimization problems determines the states of the ball pre- and post- the impact. This allows to solve the reset map and compute the desired configuration for the impacting paddle. Afterwards, the optimal trajectory planner computes the minimum acceleration paths for the free and impacting paddles to respectively reconfigure at the initial pose and intercept the ball. Finally, a closed loop inverse kinematics provides the joint motion for the robot to practically accomplish the task.

Computation of the State of the Impacting Paddle

In order to predict the state of the ball before the impact and to properly control the paddle, a two-stage nonlinear least squares fitting is designed.

The three more relevant positions of the ball in the space are depicted in Fig. 3.11 for a single iteration of the repetitive dual-hand juggling task. $\boldsymbol{p}_{init} \in \mathbb{R}^3$ is the initial position of the ball soon after the previous impact, eventually given by the visual system; $\bar{\boldsymbol{p}}_{des}$ is the desired location of the previous iteration step; $\boldsymbol{p}_{des} \in \mathbb{R}^3$ is the desired location of the current iteration step where the ball has to be re-directed after the impact; Δt is the predetermined time interval between two impacts. Notice that the location in the space where the collision between the paddle and the ball happens could be, in general, different from the predetermined desired location where the ball has to be re-directed after the impact. It goes without saying that, to obtain a repetitive dual-hand ball juggling task, the chosen position $\boldsymbol{p}_{des}$ and the interval time Δt have to shape the pattern trajectory of the ball such that the free paddle can intercept the ball within the robot workspace.

By knowing Δt and the velocity $\boldsymbol{v}^+_{init} \in \mathbb{R}^3$ after the previous collision, given again by the external visual system, it is possible to retrieve the predicted pre-impact states of the ball $(\boldsymbol{p}_{imp}, \boldsymbol{v}^-_{imp}) \in \mathbb{R}^3$ through the following minimization

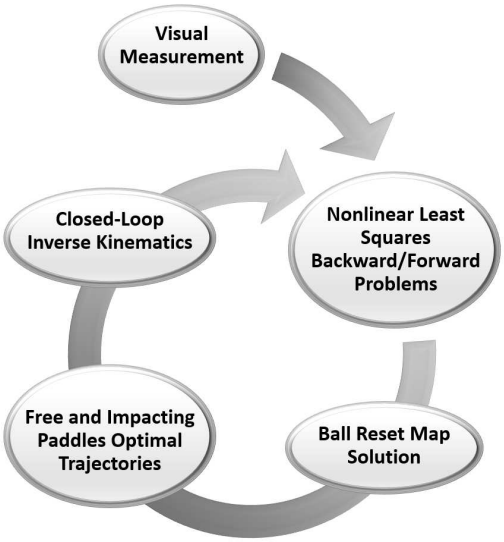


Figure 3.10: Single iteration scheme of the dual-hand ball juggling algorithm.

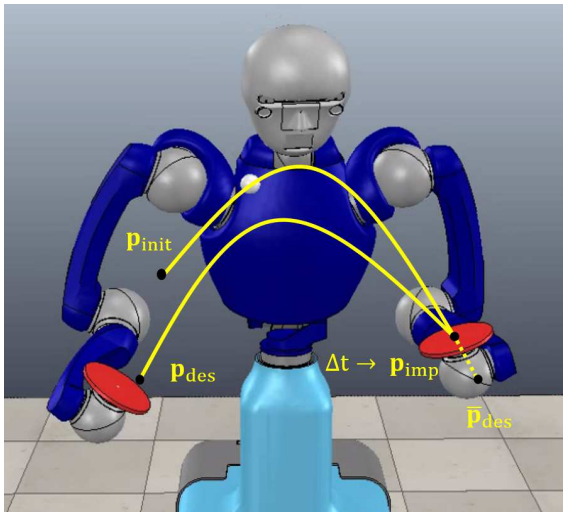


Figure 3.11: Snapshot of one iteration of the dual-hand juggling task.

problem

$$\min_{\boldsymbol{p}_{imp},\boldsymbol{v}_{imp}^{-}} \left\| \begin{bmatrix} \widetilde{\boldsymbol{p}}_{init}(\boldsymbol{p}_{imp},\boldsymbol{v}_{imp}^{-}) - \boldsymbol{p}_{init} \\ \widetilde{\boldsymbol{v}}_{init}^{+}(\boldsymbol{p}_{imp},\boldsymbol{v}_{imp}^{-}) - \boldsymbol{v}_{init}^{+} \end{bmatrix} \right\|^2, \tag{3.21}$$

where $\widetilde{\boldsymbol{p}}_{init}$ and $\widetilde{\boldsymbol{v}}_{init}^{+}$ are obtained by numerically backward integrating the nonlinear equations (3.7a) and (3.7b) starting from the current optimization variables. In rough words, the predicted pre-impact states of the ball are chosen such that, back-integrating the aerodynamic model of the ball by Δt, the value of the obtained initial state is close as much as possible to the measured one.

In order to obtain the velocity $\boldsymbol{v}_{imp}^{+} \in \mathbb{R}^3$ of the ball after the considered impact, which is necessary to direct the ball towards the predefined desired point $\boldsymbol{p}_{des}$, the following minimization problem

$$\min_{\boldsymbol{v}_{imp}^{+}} \left\| \widetilde{\boldsymbol{p}}_{des}(\boldsymbol{v}_{imp}^{+}) - \boldsymbol{p}_{des} \right\|^2, \tag{3.22}$$

is solved. Notice that $\widetilde{\boldsymbol{p}}_{des}(\boldsymbol{v}_{imp}^{+})$ is obtained by numerically forward integrating the nonlinear equations (3.7a) and (3.7b) starting from $\boldsymbol{p}_{imp}$ and the current value of the optimization variable. The predicted velocity of the ball after the impact is chosen such that, forward integrating the aerodynamic model of the ball by Δt, the value of the obtained final position is close as much as possible to the desired one.

Hence, it is now possible to determine the configuration of the impacting paddle. As an assumption, at each iteration, the predicted impact position $\boldsymbol{p}_{imp}$ for the ball corresponds to the goal position for the impacting paddle. On the other hand, having solved (3.21) and (3.22), the pose and the velocity of the impacting paddle are computed through the reset map solution reported in (3.18) in Section 3.3. In the first iteration only the impacting paddle is actuated; after that, at each impact, the free paddle is imposed to stop in a rest position $\boldsymbol{p}_0 \in \mathbb{R}^3$ and orientation $\boldsymbol{R}_0 \in SO(3)$, respectively, yielding

$$\boldsymbol{p}_{fp} = \boldsymbol{p}_0, \qquad \boldsymbol{v}_{fp} = \boldsymbol{0}, \qquad \boldsymbol{R}_{fp} = \boldsymbol{R}_0, \qquad \omega_{fp} = \boldsymbol{0}. \tag{3.23}$$

As soon as the first impact occurs, the initial configurations of ball and paddles for the next juggle cycle is available. Impacting and free paddles are swapped in order to restart another iteration of the algorithm.

The configurations of the two paddles at each impact time are the inputs of the optimal trajectory planner which online computes the optimal path for each paddle from the current state. The optimal trajectories are generated in SE(3), minimizing the acceleration functional, as detailed in Section 3.2. Finally, the motion of the robot joints is computed from the planned minimum acceleration Cartesian trajectories for the two paddles through a kinematic inversion algorithm.

Table 3.8: Input parameters for ball juggling simulations in the first case study.

Initial position of the ball	$\begin{bmatrix} 0.25 & 0 & 0.4 \end{bmatrix}$ m
Initial velocity of the ball	$\begin{bmatrix} 1.65 & -0.2 & 0.05 \end{bmatrix}$ m/s
First desired point (near right paddle)	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ m
Rest position right paddle	$\begin{bmatrix} 0 & 0 & -0.1 \end{bmatrix}$ m
Rest position left paddle	$\begin{bmatrix} 0.7 & 0 & -0.1 \end{bmatrix}$ m
Rest orientation for both paddles	$\boldsymbol{I}_{3\times 3}$

3.4.3 Ball Juggling Task Simulations

Numerical results validating the approach are presented through the three case studies described in the following. For all of them, the considered robot is the same semi-humanoid considered within Section 3.3. The robot is now equipped with two paddles firmly attached at its end-effectors (see Fig. 3.11). The dynamic parameters considered for these simulations are detailed in Table 3.2 of Section 3.3. Simulations are implemented in the Matlab environment, in connection with the V-REP virtual platform by [90]. Within the YouTube video[2], the former and latter case studies are included. The video demonstrates the smoothness of the planned motion for the joints of the semi-humanoid robot, and the synchronized motion of the two hands/paddles.

First Ball Juggling Case Study

In this case study, with reference to Fig. 3.11, the main assumption is given from the following equality $\boldsymbol{p}_{des} = \boldsymbol{p}_{init}$, holding for each iteration. This means that the desired location $\boldsymbol{p}_{des}$ for the ball after the impact with the paddle has been put equal to the previous impact location $\boldsymbol{p}_{init}$. The only exception is given by the first iteration since the ball has not performed yet any previous impact. The very first $\boldsymbol{p}_{des}$ point should be then assigned. Its value together with the initial position and velocity of the ball are detailed in Table 3.8. The chosen time interval between two consecutive impacts is set to the fixed value of $\Delta t = 0.5$ s. Only for the very first impact, this is set to 0.3 s. The rest positions $\boldsymbol{p}_0$ and orientations $\boldsymbol{R}_0$ for the paddles are indicated in Table 3.8.

The 3D trajectories planned for the left and right hands of the semi-humanoid robot and the path of the ball are depicted in Fig. 3.12, where it is possible to appreciate that the paths planned for the paddles oscillate between the respective rest positions $\boldsymbol{p}_0$ and the impact points obtained by solving the minimization

[2] `https://youtu.be/VtRe1zE_lhM`

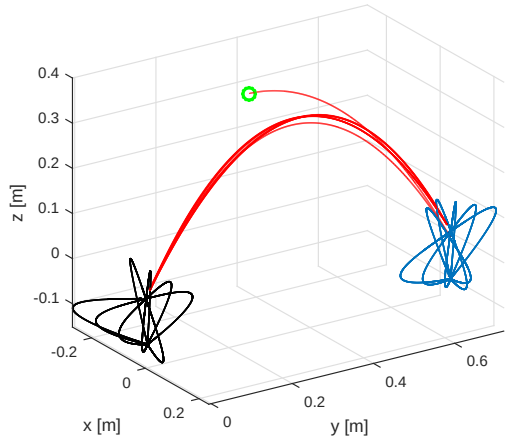


Figure 3.12: 3D trajectories of the ball (red line), left (blue line) and right (black line) paddles, for the first case study. The green marker defines the initial position of the ball.

problems previously introduced. The picture shows that the ball follows the juggling pattern.

Fig. 3.15(a) shows that the average Euclidean norm of the error between the desired points and the actual ones is about 2 cm. Only for the very first iteration, the figure shows the actual impact point compared to the predicted one. The observed error is mainly justified by the numerical optimization procedure employed in the presented formulation and explained in the previous sections.

Fig. 3.13 presents the time sequence of the semi-humanoid motion, resulting from the kinematic inversion, during the first two juggling iterations.

Second Ball Juggling Case Study

In this case study, the assumption made in the previous one is relaxed, and thus $\boldsymbol{p}_{des} \neq \boldsymbol{p}_{init}$. Therefore, seven different desired points are considered and listed in Table 3.9. In the same table, the initial position and velocity of the ball are detailed, which are different from the previous case study. The chosen time interval between two consecutive impacts is set to the fixed value of $\Delta t = 0.5$ s. Only for the very first impact, this is set to 0.3 s. The rest positions $\boldsymbol{p}_0$ and orientations $\boldsymbol{R}_0$ for the paddles are the same as the first case study (see Table 3.8).

The shape of the minimum acceleration paths planned for the paddles of the

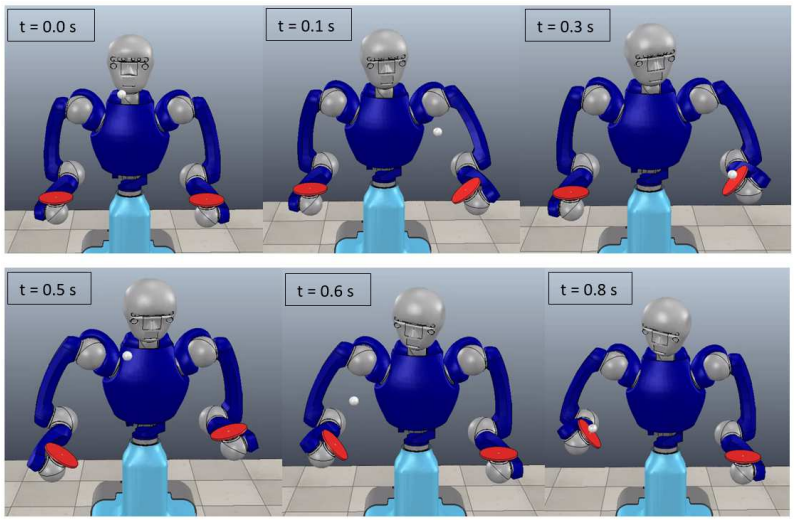


Figure 3.13: Time sequence in the V-REP environment of the motion planned for the semi-humanoid robot to accomplish the first two ball juggling iterations. The first two impacts occur at 0.3 s and 0.8 s. See the accompanying video.

Table 3.9: Input parameters for ball juggling simulations in the second case study.

Initial position of the ball	$[0.5 \quad -0.2 \quad 0.5]$ m
Initial velocity of the ball	$[1 \quad -0.2 \quad 0.05]$ m/s
1st desired point	$[0 \quad 0 \quad 0]$ m
2nd desired point	$[0 \quad -0.01 \quad 0.06]$ m
3rd desired point	$[0.7 \quad -0.085 \quad 0.1026]$ m
4th desired point	$[0 \quad -0.02 \quad 0.12]$ m
5th desired point	$[0.7 \quad -0.095 \quad 0.1626]$ m
6th desired point	$[0 \quad -0.03 \quad 0.18]$ m
7th desired point	$[0.7 \quad -0.105 \quad 0.2226]$ m

Table 3.10: Input parameters for ball juggling simulations in the third case study.

1st impact Δt	0.6 s
2nd impact Δt	0.55 s
3rd impact Δt	0.5 s
4th impact Δt	0.45 s
5th impact Δt	0.4 s
6th impact Δt	0.35 s
7th impact Δt	0.3 s

semi-humanoid and the ball path for this second case study are depicted in Fig. 3.14. Again, the average Euclidean norm of the error between the desired points and the actual ones is about 2 cm, as plotted in Fig. 3.15(b).

Third Ball Juggling Case Study

In this case study, the assumption $\boldsymbol{p}_{des} = \boldsymbol{p}_{init}$ is re-introduced, while the value of Δt now changes during each iteration. Therefore, seven different intervals of time between two impacts are considered in Table 3.10. The initial position and velocity of the ball are equal to the ones of the first case study, as well as the very first desired point. In addition, the rest positions $\boldsymbol{p}_0$ and orientations $\boldsymbol{R}_0$ for the paddles are also equal to the first case study (see Table 3.8).

The shape of the minimum acceleration paths planned for the paddles of the semi-humanoid and the ball path for this third case study are depicted in Fig. 3.16. It is possible to observe that, keeping fixed the desired points as highlighted in the assumption, the time Δt shapes the juggling patter. In particular, by reducing the time interval, the maximum height of the ball reduces as well.

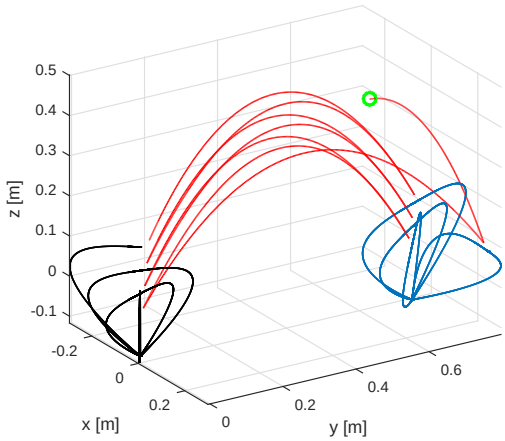
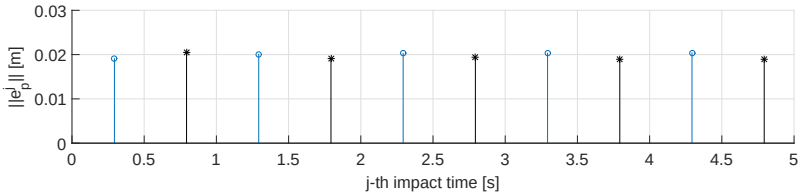
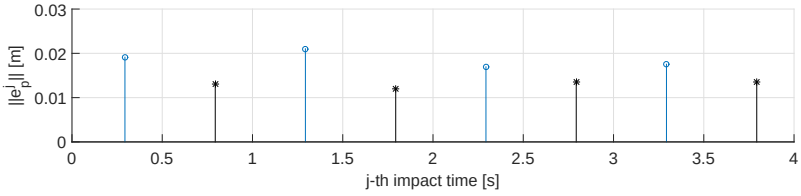


Figure 3.14: 3D trajectories of the ball (red line), left (blue line) and right (black line) paddles, for the second case study. The green marker defines the initial position of the ball.



(a) First ball juggling case study.



(b) Second ball juggling case study.

Figure 3.15: Norm of the error between the desired positions and the actual ones, at each impact time. Blue circles represent the left impact errors, while the black stars depict the right ones.

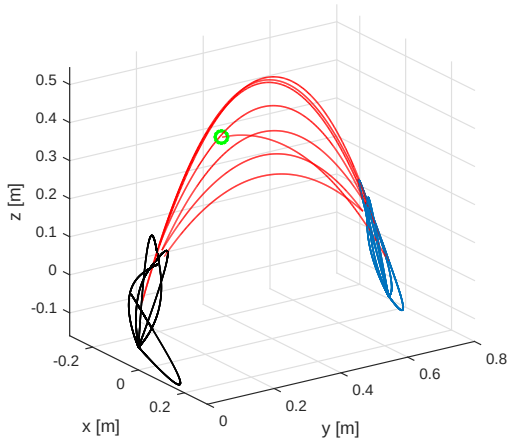


Figure 3.16: 3D trajectories of the ball (red line), left (blue line) and right (black line) paddles, for the third case study. The green marker defines the initial position of the ball.

Moreover, it is possible to notice from the same figure that, with a shorter Δt, the paddle impacts the ball before this last reaches the desired position $\boldsymbol{p}_{des}$, as in general depicted in Fig. 3.11.

3.5 Conclusion

This Chapter shows the theory to design an optimal trajectory planner in SE(3) by solving a boundary value problem with proper boundary conditions, and the hybrid dynamics of the ball given by its aerodynamics and its reset map to be exploited in the proposed model-based motion planner. The optimal planner is then applied to generate the motion of the robot end-effector to hit the ball at the impact time ad direct it towards a predefined location - batting motion primitive. In the conceptual scheme of the optimal motion planner, after that the visual system provides the configuration of the ball, the solution of a sequence of nonlinear minimization problems determines the states of the ball pre- and post-the impact. This allows to solve the reset map and compute the desired configuration for the impacting paddle. Afterwards, the optimal trajectory planner computes the minimum acceleration paths for the paddle to intercept the ball. Finally, a closed loop inverse kinematics provides the joint motion for the robot

to practically accomplish the task with the semi-humanoid robot. Furthermore, the batting primitive is exploited to engender a dual-hand ball juggling task, demonstrating that the framework presented in this Chapter produces a versatile autonomous motion planner for a number of impact manipulation tasks.

The main contribution of this Chapter consists in the design of a motion planner that is able to provide in real-time the optimal path for the robot end-effector to autonomously perform the batting task, assuming to have as input only the visual measurement of the initial configuration of the ball and the desired impact locations. The algorithm has been divided in the following three main phases: the prediction of the motion of the ball, the selection of the configuration of the ball at impact time, and the trajectory planning for the paddle. About the first and second stages, the proposed nonlinear estimation techniques improve the control accuracy with respect to other related works [65, 79], while dealing with the real-time constraints. In those works, the impact position and pre-impact velocity of the ball are a priori known. Such assumptions are quite restrictive: hence, in this Chapter the determination of such ball impact configuration is addressed at run-time. Additionally, the impact time is not a priori defined in [107], as instead in [108], but it is online predicted. Regarding the third stage, a minimum acceleration planner in SE(3), based on the theory of differential geometry, is used to generate an optimal path for the manipulator, following [120]. By choosing the acceleration norm as measure of smoothness, the trajectories can be made to satisfy boundary conditions on the velocities. In order to ensure that the computed trajectories are independent of the parametrization of positions and orientations, the notions of Riemannian metric and covariant derivative are borrowed from differential geometry and the problem is formulated as a variational problem on the Lie group of spatial rigid body displacements. The batting and the dual-hand ball juggling motion planners are implemented in the Matlab environment in connection with the V-Rep platform to numerically evaluate the algorithm and to display the smoothness of the motion planned for the joints of a semi-humanoid robot.

Chapter 4

Nonlinear Predictive Control of Multi-contact Walking

4.1 Introduction

Currently, humanoid robots are still not able to robustly use their arms to gain stability and safety, while executing locomotion tasks - the so-called *loco-manipulation* [17, 36] - or to robustly deal with multiple non-coplanar contacts with the environment - the so-called *multi-contact locomotion* tasks [27, 86, 104]. Manipulation of an object is essentially functionally identical to locomotion with only relativistic differences, see Fig. 4.1 [1]. In object manipulation, the manipulator is typically larger than the manipulated object, and so the object is moved and controlled relative to the manipulator origin. But in locomotion, the object being manipulated is the ground, which is much larger than the manipulator, so the origin of the manipulator moves relative to the manipulated object. In both these cases there is a relative movement between the object and manipulator origin, and the goal is to control that movement [111]. For instance, two slow walking legs can be seen as two fingers grasping a much larger object, i.e., the earth; a bouncing ball system can be assimilated to an hopping robot; and there is a clear analogy between the juggling and the running task. Under certain conditions, it is possible to assume that locomotion is a kind of manipulation. Some works in the literature have already highlighted that balancing, slow walking gaits and grasping tasks share several similarities. Some works have found

[1] From the German DLR Robotics and Mechatronics website - `http://www.dlr.de/`.

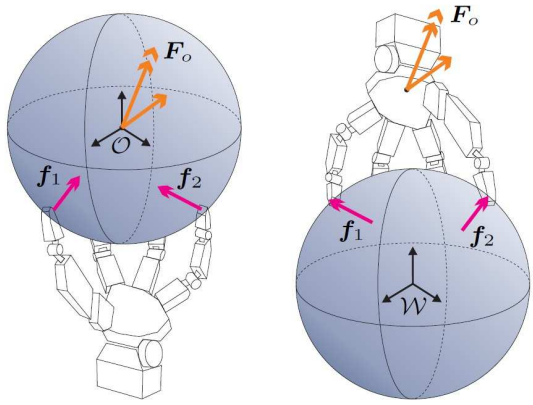


Figure 4.1: Analogy between balancing and grasping.

the duality between a multi-fingered grasp and a multi-legged stance to establish the correctness of the controller by a quasi-static analysis borrowed from the robot manipulation literature [52, 85]. These approaches are based on the observation that the problems of grasping an object and balancing a robot are fundamentally similar, in the sense that both try to achieve a desired wrench (on the object in the grasping case, on the robot in the balancing case) based on the application of suitable forces at the contact points (at the fingertips or at the feet). Moreover, in the nonprehensile manipulation tasks, similarly to the bipedal walking task, the effect of gravity must be controlled to hold the object, or to balance the robot, as explained in [10]. Interestingly, both nonprehensile dynamic manipulation and multi-contact dynamic walking share some common features, since they intrinsically involve:

- fast and hybrid dynamics;
- multiple non-coplanar contacts;
- interaction between complex shapes.

In legged locomotion one of the main difficulty is given by the dynamic constraint on the linear and angular momentum of the robot, due to its dependence on external contact forces to control them [119]. This is particularly evident in the Newton–Euler equations of motion of the robot, which can typically be expressed in the following form:

$$\begin{bmatrix} m(\ddot{\boldsymbol{c}} + \boldsymbol{g}) \\ m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} + \boldsymbol{g}) + \dot{\boldsymbol{L}} \end{bmatrix} \in \mathcal{C}, \tag{4.1}$$

where $m \in \mathbb{R}$ is the mass of the robot, $\boldsymbol{c} \in \mathbb{R}^3$ is the position of its CoM in a world frame, $\boldsymbol{L} \in \mathbb{R}^3$ is its angular momentum with respect to the CoM, and $\mathcal{C} \subset \mathbb{R}^6$ is the cone of available contact wrenches (forces and torques), which depends on the existing contacts between the robot and its environment.

This model allows to tackle scenarios such as bipedal legged locomotion, or even dynamic object manipulation, to generate a reference motion that, in general, should be:

- kinematically feasible, i.e. ensuring that the constraints on the physics of the system are always respected;
- dynamically feasible, i.e. guaranteeing that the center of pressure (CoP) is inside the support polygon, or, at a torque level, that the dynamic wrench is inside the contact wrench cone;
- robust, i.e. maximizing some kind of robustness metric in the dynamic feasibility condition;
- smooth, i.e. minimizing the acceleration, or the jerk, of the CoM.

Additonally, in particular for dynamic walking tasks, natural motion results from the minimization of the centroidal angular momentum. Even if a high angular momentum does not always result in a fall, low angular momentum indicates a good balance and a low fall risk during locomotion. For dynamic object manipulation, the centroidal angular momentum should be regulated in a task-specific way. For example, if the manipulation is supposed to end with a stationary object, the final centroidal angular momentum is supposed to be zero, otherwise other constraints can be conceived. However, in this Chapter the focus is on multi-contact dynamic walking applications.

The standard approach to compute and control walking motions, satisfying the balance contraints, is through MPC [74, 119]. When walking on a flat, or moderately uneven ground, as long as hand contacts are not necessary, the constraint (4.1) can take a linear form [18], and very efficient numerical methods can be employed, to compute such an MPC scheme exactly in a fraction of milli-second. However, humans use hand contacts to interact with the environment, or to simply realize otherwise unfeasible motions. Hand contacts clearly distribute weight, and then improve stability and energetic efficiency [3, 11, 29, 49, 58, 62, 72]. The work [45] indeed exploits the relation between the force exerted by the robot to grasp a handrail and the size of the region of the CoP for keeping balance. It shows that the stronger the hand of a robot grasps a handrail, the larger the region of balance becomes. Whereas, another approach to walking upstairs and downstairs with external support [59] shows that the handrail support can reduce the motor power consumption up to 25%.

Multi-contact locomotion control might have decisive impact in some future robotic applications. It enables a robot to stand up after a fall, or evolve in unstructured environments. For example, in an industrial process, where a walking robot might be required to walk upstairs and downstairs for several hours, the reduction of energy consumption with the use of a handrail could be crucial for efficiency. Nevertheless, with the introduction of additional contacts, the constraint (4.1) becomes generally nonlinear, and it is necessary to consider a Newton method to obtain a solution, a potentially time-consuming and uncertain procedure involving iterative linearizations.

In an MPC scheme, a sequence of optimization problems have to be solved, one at each sampling time, which are closely related. Using tools from parametric optimization and continuation methods, it is possible to obtain very good approximate solutions to these nonlinear problems, with very few, or even only one iteration at each sampling time [30]. The problem, however, is that these approximate solutions may not always be feasible and satisfy exactly the nonlinear constraints.

This is the problem that is addressed in this Chapter, by proposing a Newton method based on carefully crafted linear approximate models, which ensure that the nonlinear constraint (4.1) is always satisfied exactly. The approach is based on a linear approximation which is made robust to polytopic uncertainties, following the same procedure as in [18], and constraining each iteration of the Newton method to fall within the limits of the specified polytopic uncertainty. A specific form of the constraint (4.1) is used, where a CoP is introduced for foot contacts on the ground, while other external forces are considered as a generic force $\boldsymbol{f}_e \in \mathbb{R}^3$ and torque $\boldsymbol{n}_e \in \mathbb{R}^3$ acting on the CoM, as proposed in [3]. Simulation results display the generated walking motions up and down stairs with an additional hand support, which satisfies exactly the balance constraint (4.1) [105].

4.2 Minimization of the Angular Momentum

For a manipulation or a legged locomotion task, the momentum dynamics can be characterized formulating the dynamic balance of momenta of either the object, in case of manipulation, or the robot, in case of walking task. Then, rewriting the Newton–Euler equations of motion (4.1)

$$\begin{aligned} \dot{\boldsymbol{l}} &= m\boldsymbol{g} + \Sigma_i \boldsymbol{f}_i, \\ \dot{\boldsymbol{k}} &= (\boldsymbol{c} - \boldsymbol{s}_f) \times m\boldsymbol{g} + \Sigma_i (\boldsymbol{s}_i - \boldsymbol{s}_f) \times \boldsymbol{f}_i + \boldsymbol{\tau}_i, \\ \dot{\boldsymbol{c}} &= \frac{\boldsymbol{l}}{m}, \end{aligned} \tag{4.2}$$

where $\boldsymbol{c} \in \mathbb{R}^3$ is the position of the CoM of the object/robot, in the world frame, $\boldsymbol{l} \in \mathbb{R}^3$ is the linear momentum of the object/robot in the world frame, $\boldsymbol{k} \in \mathbb{R}^3$ is its angular momentum with respect to $\boldsymbol{s}_f \in \mathbb{R}^3$, which is one of the support points, $m \in \mathbb{R}$ is the mass, $\boldsymbol{s}_i \in \mathbb{R}^3$ is the i-th support point, $\boldsymbol{f}_i \in \mathbb{R}^3$ and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ are respectively the i-th support force and torque, and $\boldsymbol{g} \in \mathbb{R}^3$ is the gravity acceleration vector. In the following superscripts x, y and z indicate the corresponding components of a vector.

When generating standard walking motions, the variation of the centroidal angular momentum is assumed to be zero, and some simplifications, like constant height of the CoM and co-planar contact points, allow to keep the model linear and well suited for preview based control [119]. In [55], the authors present the resolved momentum control, that is one of the first attempt to generate whole body motion of a humanoid robot such that the resultant total linear and angular momenta become specified values. They found a linear relation between the total momentum and the joint speeds; and, as case study, the kick action is evaluated considering zero reference angular momentum.

Nevertheless, while dealing with more complex tasks, like walking motion in multi-contact configurations for example, several simplifying assumptions are no more valid. It is difficult to preserve a linear structure of the optimization problem, and model predictive control methods become computationally hard to apply at once. The general approach to motion generation with such complex nonlinear models is to resort to nonlinear solvers such as sequential quadratic programming [24], [49], or interior point methods [87], that might require long time to find a solution, or, in the worst case, they could not find it at all. For this reason, several methods are investigated in the literature to deal with the non-convexity of the model in such a way that it can be formulated as quadratic program (QP) [28], [86], or quadratically constrained QP [23]. Here, a comparison between different approaches to the minimization of the centroidal angular momentum is showed.

In the expression of the centroidal angular momentum

$$\boldsymbol{k}_c = \boldsymbol{k} - m(\boldsymbol{c} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}, \tag{4.3}$$

the cross product clearly induces a nonlinearity, considering the motion of the CoM as decision variable. Similarly, in (4.2), when both contact points and contact forces are considered as decision variables, the same kind of nonlinearity appears in the cross product $(\boldsymbol{s}_i - \boldsymbol{s}_f) \times \boldsymbol{f}_i$. Therefore, this non-convexity appears not only in the generation of a natural motion for the robot, but also in the problem of planning contact forces and contact locations online.

In [28], in order to plan the motion through convex optimization, a convex upper bound of the L_1 norm of the centroidal angular momentum is minimized with respect to the decision variables $\boldsymbol{k}$, $\boldsymbol{c}$, $\dot{\boldsymbol{c}}$, considering pre-planned contact

positions. An upper bound on $\boldsymbol{k}_c$, suitable for the inclusion in a preview based control, can be conceived in different ways. Several recent works in the literature are focused on the reformulation of this non-convexity in the centroidal angular momentum.

The authors of [28] exploit a bounding on the 3D position of the CoM, stating that with pre-specified footsteps the admissible region of the CoM is also bounded. In particular, for a polytopic admissible CoM region, P_v, with n vertices, $\underline{\boldsymbol{c}}_{xyz} \in \mathbb{R}^3$, this polytope can be represented by linear constraints. The convex upper bound of $|\boldsymbol{k}_c|_1{}^2$ is obtained replacing the CoM position with the vertex of the polytope that maximizes the L_1 norm of the centroidal angular momentum

$$\max_{\underline{\boldsymbol{c}}_{xyz} \in P_v} \quad |\boldsymbol{k} - m(\underline{\boldsymbol{c}} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}|_1 \ . \tag{4.4}$$

Another upper bound can be proposed using the result showed in [18], which bounds the variation of vertical motion, with the aim to generate a 3D motion of the CoM as high as possible above the ground. This approach arises from the observation that in human walking the vertical direction of motion is the less variable one. Considering a polytopic admissible region for the vertical position and velocity of the CoM, P_z, with 4 vertices, $[\underline{\boldsymbol{c}}^z, \underline{\dot{\boldsymbol{c}}}^z] \in \mathbb{R}^2$ (which provides a convex set of constraints), and the corresponding position and velocity of the CoM, $\underline{\boldsymbol{c}}_z = [c^x, c^y, \underline{c}^z] \in \mathbb{R}^3$ and $\underline{\dot{\boldsymbol{c}}}_z = [\dot{c}^x, \dot{c}^y, \underline{\dot{c}}^z] \in \mathbb{R}^3$, the proposed upper bound of $|\boldsymbol{k}_c|_1$ is given by

$$\max_{\underline{c}^z, \underline{\dot{c}}^z \in P_z} \quad |\boldsymbol{k} - m(\underline{\boldsymbol{c}}_z - \boldsymbol{s}_f) \times \underline{\dot{\boldsymbol{c}}}_z|_1 \ . \tag{4.5}$$

Though it provides a bound only on the horizontal components of the centroidal angular momentum, it results in a less constrained motion with respect to the approach in [28] (4 constraints in place of $n \geq 6$), and it can be applied in case of adaptive footstep placement. It is, however, reasonable to neglect the vertical component of $\boldsymbol{k}_c$, since it does not affect significantly the motion of the CoM. In both the above mentioned approaches, an online tuning of the polytopic admissible region would definitely provide the ability to deal with less structured walking motions, as detailed in the following Sections.

A third method, similarly to [86], can be thought exploiting a decomposition of the cross product in scalar products, and then in differences of quadratic functions

$$(\boldsymbol{c} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}} = \frac{1}{\alpha} \begin{bmatrix} a^T \tilde{b} \\ g^T \tilde{d} \\ e^T \tilde{f} \end{bmatrix} = \frac{1}{4\alpha} \begin{bmatrix} ||a + \tilde{b}||_2^2 - ||a - \tilde{b}||_2^2 \\ ||g + \tilde{d}||_2^2 - ||g - \tilde{d}||_2^2 \\ ||e + \tilde{f}||_2^2 - ||e - \tilde{f}||_2^2 \end{bmatrix}, \tag{4.6}$$

[2]The operator $|\cdot|_1$ indicates here the L_1 norm.

where $a = [s_f^z - c^z, c^y - s_f^y]^T$, $b = [\dot{c}^y, \dot{c}^z]^T$, $g = [c^z - s_f^z, s_f^x - c^x]^T$, $d = [\dot{c}^x, \dot{c}^z]^T$, $e = [s_f^y - c^y, c^x - s_f^x]^T$, $f = [\dot{c}^x, \dot{c}^y]^T$, and $\tilde{b} = \alpha b$, $\tilde{d} = \alpha d$, $\tilde{f} = \alpha f$. The normalization factor $\alpha > 0$ is necessary to match the units of measurement of each term of the decomposition. Consequently, an upper bound on the L_1 norm of the centroidal angular momentum is given by the triangular inequality[3]

$$
\begin{aligned}
&|\boldsymbol{k} - m(\boldsymbol{c} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}|_1 \leq |\boldsymbol{k}|_1 + \frac{m}{\alpha}\left(|a^T\tilde{b}|_1 + |g^T\tilde{d}|_1 + |e^T\tilde{f}|_1\right) = \\
&|\boldsymbol{k}|_1 + \frac{m}{4\alpha}\left(\left|||a+\tilde{b}||_2^2 - ||a-\tilde{b}||_2^2\right|_1 + \left|||g+\tilde{d}||_2^2 - ||g-\tilde{d}||_2^2\right|_1 + \left|||e+\tilde{f}||_2^2 - ||e-\tilde{f}||_2^2\right|_1\right) \leq \\
&|\boldsymbol{k}|_1 + \frac{m}{4\alpha}\left(||a+\tilde{b}||_2^2 + ||a-\tilde{b}||_2^2 + ||g+\tilde{d}||_2^2 + ||g-\tilde{d}||_2^2 + ||e+\tilde{f}||_2^2 + ||e-\tilde{f}||_2^2\right) = \\
&|\boldsymbol{k}|_1 + \frac{m}{2\alpha}\left(||a||_2^2 + ||\tilde{b}||_2^2 + ||g||_2^2 + ||\tilde{d}||_2^2 + ||e||_2^2 + ||\tilde{f}||_2^2\right).
\end{aligned} \tag{4.7}
$$

Since, for any x and y vectors in $\mathbb{R}^n$, it holds

$$
||\boldsymbol{x}||_2^2 + ||\boldsymbol{y}||_2^2 - 2|\boldsymbol{x}^T\boldsymbol{y}|_1 = ||\boldsymbol{x} - \boldsymbol{y}||_2^2, \tag{4.8}
$$

in order to obtain the tightest upper bound, the normalization factor α should be chosen in such a way to minimize the difference between x and y, which in this case corresponds to

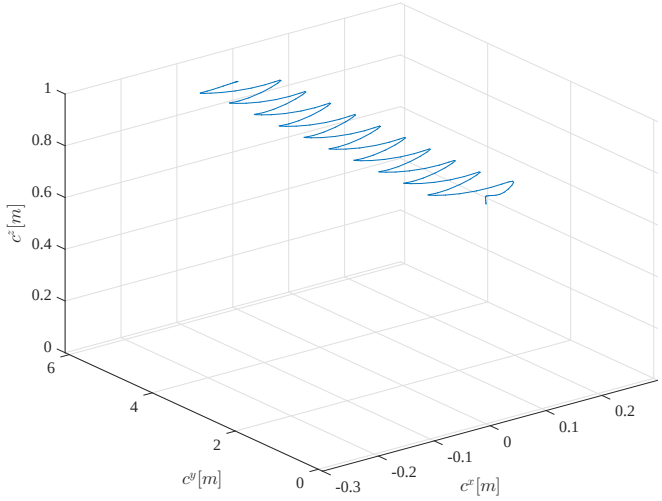
$$
h(\alpha) = \frac{m}{2\alpha}\left(||a - \tilde{b}||_2^2 + ||g - \tilde{d}||_2^2 + ||e - \tilde{f}||_2^2\right). \tag{4.9}
$$

This approach provides a bound on the three components of the centroidal angular momentum, without constraining the motion of the CoM.

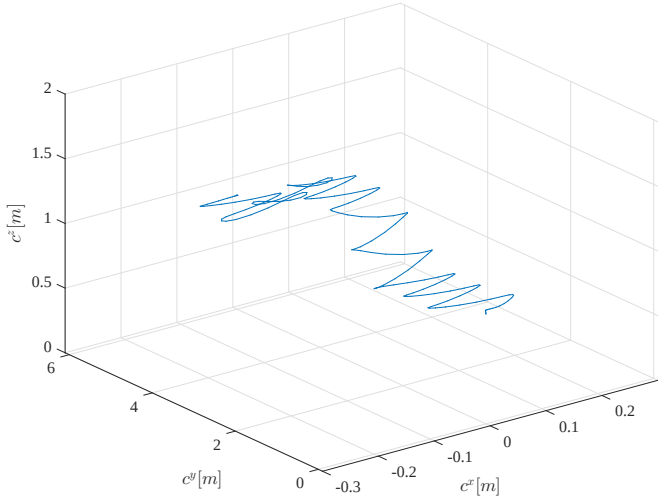
4.2.1 Comparative Results for Walking Applications

In order to numerically compare the different approaches, two sample walking motions are considered, namely on flat floor and on stairs, see Fig. 4.2, resulting from the implementation of a standard walking pattern generator. In this simulation the height of stairs is 0.1 m. Here the focus is on the cross product that appears in the centroidal angular momentum, $(\boldsymbol{c} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}$, since the term $\boldsymbol{k}$ is not a source of non-convexity if it is chosen as parameter in the minimization, which is a reasonable design choice. In this context, $\boldsymbol{s}_f$ is the pre-planned position of the support foot. The cross product that appears in the derivative of the centroidal angular momentum, $\dot{\boldsymbol{k}}_c = \dot{\boldsymbol{k}} - (\boldsymbol{c} - \boldsymbol{s}_f) \times \ddot{\boldsymbol{c}}$, is evaluated as well, since it could be another cost term to minimize. In the following, the upper bound resulting from the methods inspired by [28], [18], [86] are, respectively, named A, B, C. It is necessary to point out that a strong claim on the best method cannot be stated, because it is always difficult to obtain a fair comparison, and this analysis should be validated generating online the linear and angular momenta; however, it can be used as a preliminary result.

[3]The operator $||\cdot||_2$ indicates here the L_2 norm.

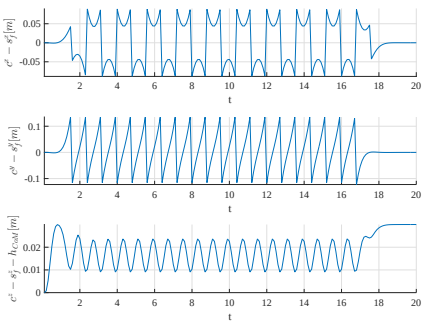


(a) CoM walking sample motion.

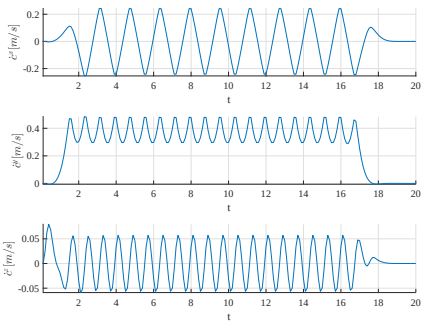


(b) CoM climbing upstairs and downstairs sample motion.

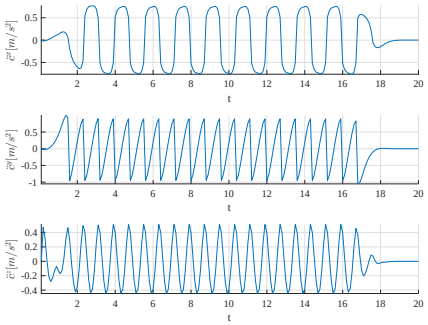
Figure 4.2: Sample motion of the CoM.



(a) CoM position w.r.t. a reference placed at the height of the CoM (h_{CoM}) above the support foot.



(b) CoM velocity w.r.t. the world frame.



(c) CoM acceleration w.r.t. the world frame.

Figure 4.3: CoM for walking on flat floor. The top, middle and bottom figures depict the x, y, z components respectively.

Walking on Flat Floor

Considering standard walking on flat floor, in order to evaluate the upper bound A, 8 vertices for the admissible polytopic region of position of the CoM are chosen according to this specific kind of motion. The region is centered at 0.8 m and is a box of size 0.2 m $\times$ 0.3 m $\times$ 0.08 m. This box is obtained exploiting the data depicted in Fig. 4.3(a). Whereas, to compute the bound B on the centroidal angular momentm, the admissible region is such that the vertical position of the CoM and its vertical velocity are inside a box of size 0.08 m $\times$ 0.16 m/s, which is extracted from the data showed in the bottom picture in Fig. 4.3(b). While, for the derivative of the centroidal angular momentum, the admissible region

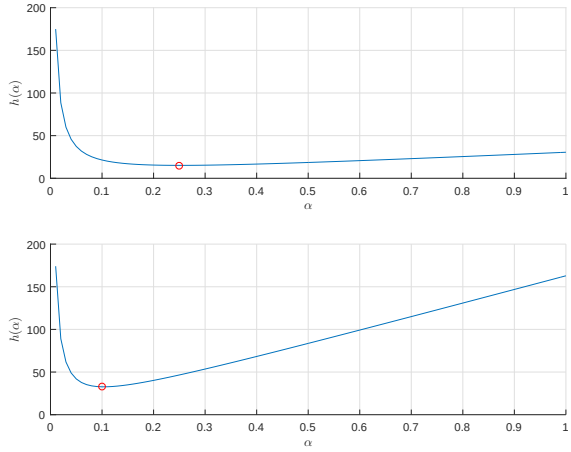


Figure 4.4: Choice of the normalization factor α for the angular momentum (top) and for its derivative (bottom), in case of walking on flat floor.

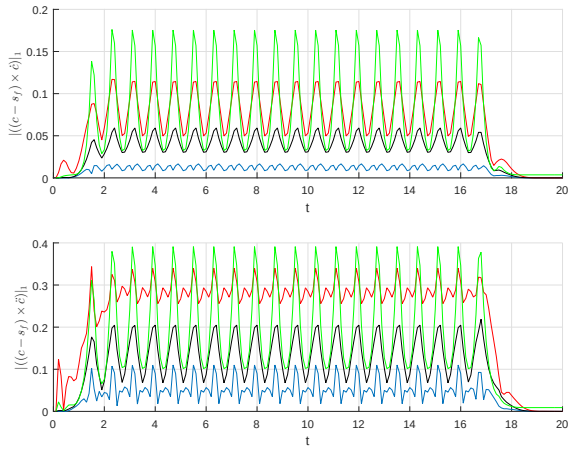


Figure 4.5: Comparison between the upper bounds on the L_1 norm of the cross product in the angular momentum (top) and in its derivative (bottom), in case of walking on flat floor. The blue line is the actual value, the red line is the bound inspired by [28], the black line is the bound inspired by [18], and the green line is the bound inspired by [86].

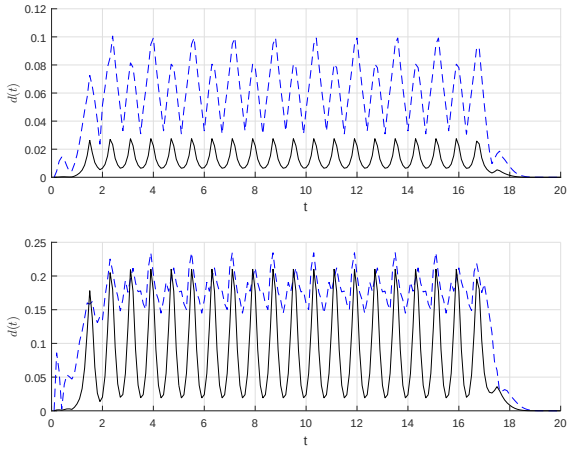
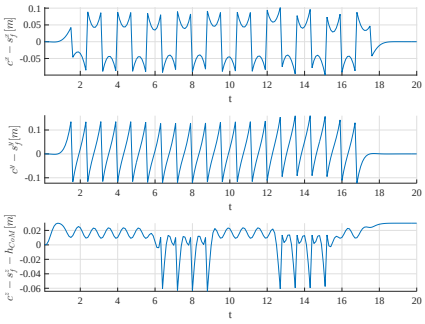


Figure 4.6: Distance between the actual value of the cross product and its upper bound, in the angular momentum (top) and in its derivative (bottom), in case of walking on flat floor. The blue dashed line is from the approach inspired by [28] and the black line is from the approach inspired by [18].

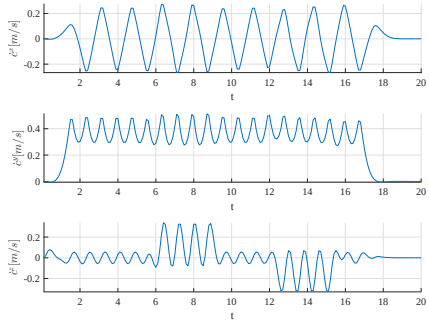
is such that the vertical position of the CoM and its vertical acceleration are inside a box of size 0.08 m $\times$ 1.2 m/s^2, according to the bottom picture in Fig. 4.3(c). Besides, for the bound C, the normalization factor, α, results from the minimization of $h(\alpha)$, which is reported in Fig. 4.4. In Fig. 4.5, a comparison between the actual value of $|(c - s_f) \times \dot{c}|_1$ and the upper bounds obtained with the different approaches reveals that, for walking on flat floor, the tightest bound is the one inspired by [18]. In order to provide a measurement of the distance between the actual value of the cross product and its upper bound for methods A and B, the vertices of the admissible regions for the CoM which maximizes the L_1 norm of the centroidal angular momentum are defined as $\underline{\boldsymbol{c}}^*_{xyz}$ and $[\underline{\boldsymbol{c}}^*_z, \underline{\dot{\boldsymbol{c}}}^*_z]$, for methods A and B respectively. Consequently, the cross product $\boldsymbol{v} = (\boldsymbol{c} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}$ for methods A and B corresponds respectively to: $\boldsymbol{v}_A = (\underline{\boldsymbol{c}}^*_{xyz} - \boldsymbol{s}_f) \times \dot{\boldsymbol{c}}$, and $\boldsymbol{v}_B = (\underline{\boldsymbol{c}}^*_z - \boldsymbol{s}_f) \times \underline{\dot{\boldsymbol{c}}}^*_z$. The Euclidean norms of the error on the cross products are given by

$$d(t) = ||\boldsymbol{v} - \boldsymbol{v}_A||_2, \qquad d(t) = ||\boldsymbol{v} - \boldsymbol{v}_B||_2, \tag{4.10}$$

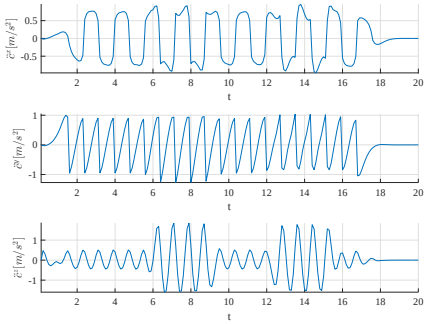
for method A and B respectively. Fig. 4.6 shows the evolution of $d(t)$ for both methods, confirming that the tightest bound is the one inspired by [18].



(a) CoM position w.r.t. a reference placed at the height of the CoM (h_{CoM}) above the support foot.



(b) CoM velocity w.r.t. the world frame.



(c) CoM acceleration w.r.t. the world frame.

Figure 4.7: CoM for walking on stairs. The top, middle and bottom figures depict the x, y, z components respectively.

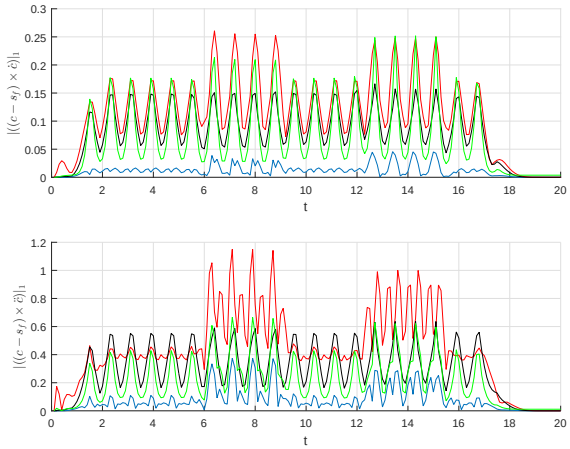


Figure 4.8: Comparison between the upper bounds on the L_1 norm of the cross product in the angular momentum (top) and in its derivative (bottom), in case of walking on stairs. The blue line is the actual value, the red line is the bound inspired by [28], the black line is the bound inspired by [18], and the green line is the bound inspired by [86].

Climbing Upstairs and Downstairs

For walking on stairs, the admissible polytopic regions are adapted to the motion depicted in Fig. 4.7. For the upper bound A, the CoM is constrained in a box of size 0.3 m $\times$ 0.4 m $\times$ 0.14 m, according to the motion depicted in Fig. 4.7(a), where walking upstairs occurs in the time interval $[6s, 10s]$ and walking downstairs occurs in the time interval $[12s, 16s]$. In order to compute the bound B on the centroidal angular momentm, the admissible region is a box of size 0.14 m $\times$ 0.8 m/s, see the bottom picture in Fig. 4.7(b). While, for the derivative of the centroidal angular momentum, the admissible region is a box of size 0.14 m $\times$ 4 m/s^2, see the bottom picture in Fig. 4.7(c). For the bound C, the normalization factor α again results from the minimization of $h(\alpha)$. In Fig. 4.8, the comparison between the actual value of $|(\boldsymbol{c}-\boldsymbol{s}_f)\times\dot{\boldsymbol{c}}|_1$ and the upper bounds obtained with the different approaches reveals that, even if in the average over the whole interval of time the tightest bound is C, considering exclusively time intervals when walking on stairs is planned, the tightest bound results B. This is due to the choice of the box on vertical motion that is well suited for high vertical displacements. The result of the analysis suggests that the online adjustment of the box related to the upper bound B leads to the best bounding solution.

4.3 Nonlinear Model Predictive Control with Always Feasible Iterates

Even if humans often use hands contacts to achieve more stable poses and movements, the problem of generating whole-body motions with multiple contacts for humanoid robots is very challenging and the traditional approaches to guarantee balance cannot be used. This Section presents a NMPC scheme, where the approach inspired by [18] is online adapted. The idea is built on the insight, described in the previous Section, that variations in the vertical direction of motion are much less relevant than variations in the posterior-anterior directions of motion for a walking task. Therefore, a transformation of variables is designed to properly bound the vertical motion, and iteratively adapt the bounds online. The proposed NMPC approach is able to generate reference walking motions in case of multi-contact walking [105].

4.3.1 Model Predictive Control

Model predictive control allows to impose constraints that will be respected within all the preview horizon. For this reason, it is widely adopted to guarantee balance for walking pattern generation [119]. MPC solves online, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant [74]. The MPC algorithm consists in the three following steps, executed at each sampling time:

- measurement of the actual state;
- computation of the control that optimizes a given state-dependent cost function, on a finite horizon, starting from the current discrete time;
- application of the control input at the first-time index only.

This control framework is deeply inspected for walking pattern generation since many years [119], because it allows to impose constraints that will be respected by all states of the preview horizon. MPC allows to control the system such that future states are also taken into account.

Typically, in the MPC technique, the model is iterated over N discrete steps, obtaining a condensed problem formulation where the previewed future states are a function of the current state and of the current control inputs. In this context, the state is given by the linear position, velocity and acceleration of the CoM, at time kT (where $T \in \mathbb{R}$ is the sampling period), for each $\upsilon = \{x, y, z\}$ Cartesian coordinate, $\hat{\boldsymbol{c}}_k^{\upsilon} = \begin{bmatrix} c_k^{\upsilon} & \dot{c}_k^{\upsilon} & \ddot{c}_k^{\upsilon} \end{bmatrix}^T \in \mathbb{R}^3$. Whereas, the input is given by

the jerk of the CoM and the external forces and torques, at time kT, $\dddot{c}_k^{\upsilon} \in \mathbb{R}$. The output of this walking system is the CoP, at time kT. The trajectories of the CoM are discretized as piecewise cubic polynomials, for each $\upsilon = \{x, y, z\}$ Cartesian coordinate, [118]

$$\hat{\boldsymbol{c}}_{k+1}^{\upsilon} = \boldsymbol{A}\hat{\boldsymbol{c}}_k^{\upsilon} + \boldsymbol{B}\dddot{c}_k^{\upsilon}, \tag{4.11}$$

where,

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}. \tag{4.12}$$

4.3.2 Introducing the CoP

Let's consider the contact forces $\boldsymbol{f}_i \in \mathbb{R}^3$ between the feet and the ground separately from the rest of external forces and torques in the Newton and Euler equations of motion, as proposed in [3]:

$$m(\ddot{\boldsymbol{c}} + \boldsymbol{g}) = \boldsymbol{f}_e + \sum_i \boldsymbol{f}_i, \tag{4.13}$$

$$m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} + \boldsymbol{g}) + \dot{\boldsymbol{L}} = \boldsymbol{n}_e + \boldsymbol{c} \times \boldsymbol{f}_e + \sum_i \boldsymbol{s}_i \times \boldsymbol{f}_i, \tag{4.14}$$

where $\boldsymbol{s}_i \in \mathbb{R}^3$ are the positions of the contact points between the feet and the ground, and everything else is as in (4.1).

Dividing the Euler equation by the z component of the Newton equation results

$$\frac{m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} + \boldsymbol{g}) - \boldsymbol{c} \times \boldsymbol{f}_e - \boldsymbol{n}_e + \dot{\boldsymbol{L}}}{m(\ddot{c}^z + g^z) - f_e^z} = \frac{\sum_i \boldsymbol{s}_i \times \boldsymbol{f}_i}{\sum_i f_i^z}. \tag{4.15}$$

In the proposed approach, dynamic feasibility needs to be checked only every 100 ms. By synchronizing this sampling period with the single support phases, it reasonably assumed that, even when walking on uneven ground, dynamic feasibility needs to be checked only at time instances where all contact points are coplanar. For this reason, a different frame for each ground contact surface is chosen in such a way that all the corresponding contact points have the same height, $s_i^z = s^z$, and gravity is orthogonal to the ground, $g^x = g^y = 0$. Focusing on the x and y components of equation 4.15, the CoP is now introduced

$$\frac{\sum_i \boldsymbol{s}_i^{xy} f_i^z}{\sum_i f_i^z} = \boldsymbol{c}^{xy} - \frac{(c^z - s^z)(m\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{f}_e^{xy})}{m(\ddot{c}^z + g^z) - f_e^z} - \frac{\boldsymbol{\Omega}\,\tilde{\boldsymbol{n}}^{xy}}{m(\ddot{c}^z + g^z) - f_e^z}, \tag{4.16}$$

where $\tilde{\boldsymbol{n}} = \dot{\boldsymbol{L}} - \boldsymbol{n}_e = \sum_i (\boldsymbol{s}_i - \boldsymbol{c}) \times \boldsymbol{f}_i$ is the torque exerted by the contact forces $\boldsymbol{f}_i$ with respect to the CoM, and

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Note that the CoP is a linear function of the x and y components of motion ($\boldsymbol{c}^{xy}$, $\ddot{\boldsymbol{c}}^{xy}$, $\boldsymbol{f}_e^{xy}$, $\tilde{\boldsymbol{n}}^{xy}$), and a nonlinear function of the z components (c^z, $\ddot{c}^z$, f_e^z). This model has been chosen for simplicity and for computational issues. It describes the components of motion which are independent from the actuation power of the robot. Validity of this simplified model has been tested with whole body control simulations, presented in Section 4.4.1.

Dynamic Feasibility Constraint

Notice how the dynamic feasibility constraint in case of multi-contact walking differs from the standard walking, or the 3D walking, case. Since the contact forces with the ground are usually unilateral, $f_i^z \geq 0$ for all i, the CoP must lie within the convex hull of the contact points $\boldsymbol{s}_i^{xy}$, $\mathcal{S}(\boldsymbol{s}_i^{xy})$. The classical expression of the constraint on the CoP in case of standard walking motion is given by [54]

$$\boldsymbol{p} = \boldsymbol{c}^{xy} - \frac{h^z}{g^z} \ddot{\boldsymbol{c}}^{xy} \in \mathcal{S}(\boldsymbol{s}_i^{xy}), \tag{4.17}$$

where the height of the CoM, $h^z \in \mathbb{R}$, is assumed fixed. While in case of 3D motion it can be formulated as [18]

$$\boldsymbol{p} = \boldsymbol{c}^{xy} - \frac{c^z - s^z}{\ddot{c}^z + g^z} \ddot{\boldsymbol{c}}^{xy} \in \mathcal{S}(\boldsymbol{s}_i^{xy}), \tag{4.18}$$

where the height of the CoM with respect to the height of the support foot already introduces a nonlinearity in the constraint. In the multi-contact configuration the external force and torque are mapped in the dynamic feasibility constraint as follows [105]

$$\boldsymbol{p} = \boldsymbol{c}^{xy} - \frac{(c^z - s^z)(m\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{f}_e^{xy})}{m(\ddot{c}^z + g^z) - f_e^z} - \frac{\boldsymbol{\Omega}\, \tilde{\boldsymbol{n}}^{xy}}{m(\ddot{c}^z + g^z) - f_e^z} \in \mathcal{S}(\boldsymbol{s}_i^{xy}). \tag{4.19}$$

In this case the constraint includes both the variable height of the CoM and the interaction of the robot with the environment, modeled as the external wrench.

4.3.3 MPC Constraints and Objectives

Constraints

The constraints included in the MPC formulation are the following. Firstly, the dynamic feasibility constraint in case of multi-contact walking:

$$\boldsymbol{p} \in \mathcal{S}(\boldsymbol{s}_i^{xy}). \tag{4.20}$$

Secondly, the kinematic constraint, where the maximal reachable region for the CoM with respect to the center of the support foot $\boldsymbol{s}_f \in \mathbb{R}^3$, is typically approximated by a convex polytope, is given by:

$$\boldsymbol{A}_k(\boldsymbol{c} - \boldsymbol{s}_f) \leq \boldsymbol{b}_k, \tag{4.21}$$

where the expression of $\boldsymbol{A}_k$ and $\boldsymbol{b}_k$ can be derived from the paper [18]. Finally, external forces and torques are bounded as:

$$\underline{\boldsymbol{f}_e} \leq \boldsymbol{f}_e \leq \overline{\boldsymbol{f}_e}, \quad \underline{\tilde{\boldsymbol{n}}} \leq \tilde{\boldsymbol{n}} \leq \overline{\tilde{\boldsymbol{n}}}. \tag{4.22}$$

Regarding the bounding of the external wrench, the simple approach proposed in [59] is followed; the constraint on $\tilde{\boldsymbol{n}}$ bounds the variation of angular momentum created by contacts between the robot and its environment on the CoM.

Cost Terms

One of the objectives of the proposed approach is to minimize the distance between the CoM and a high enough reference $\bar{c}_s^z \in \mathbb{R}$,

$$o_1 = \left\| (c^z - s^z) - \bar{c}_s^z \right\|^2. \tag{4.23}$$

Then, robustness of walking motion to perturbations is improved by minimizing the deviation of the CoP from the center of the support foot $\boldsymbol{s}_f^{xy}$,

$$o_2 = \left\| \boldsymbol{p} - \boldsymbol{s}_f^{xy} \right\|^2. \tag{4.24}$$

And, additionally, motion smoothness is improved by minimizing additionally the jerk of the CoM and the external wrench,

$$o_3 = \left\| \dddot{\boldsymbol{c}} \right\|^2, \quad o_4 = \left\| \boldsymbol{f}_e \right\|^2, \quad o_5 = \left\| \tilde{\boldsymbol{n}} \right\|^2. \tag{4.25}$$

The proposed cost terms are quite standard for walking motion generation. The minimization of the external wrench is mainly introduced as a form of regularization of the objective function. More sophisticated performance metric could be included in the cost function, for example considering a measurement of energy consumption from joint torques, but this is out of the scope of the work. The simplicity of the proposed cost terms make them suitable for the inclusion in a preview control approach.

4.3.4 Optimal Control Problem

In the end, to find the trajectories of the CoM and the external wrench that satisfy the robot dynamics, while generating a motion similar to the one observed in humans, an OCP is formulated in principle, on a finite time horizon T_h

$$\begin{aligned} \min_{\dddot{\boldsymbol{c}}, \boldsymbol{f}_e, \tilde{\boldsymbol{n}}} \quad & \int_0^{T_h} \sum_{i=1}^{5} w_i o_i dt \\ \text{subject to} \quad & (4.20), (4.21), (4.22), \quad \forall t, \end{aligned} \tag{4.26}$$

where o_i are the cost terms reported in (4.23), (4.24), and (4.25), and w_i are the corresponding weights.

4.3.5 Always Feasible Newton Iterates

In the following the proposition for making sure that every iteration of the Newton scheme is always feasible, satisfying exactly the nonlinear constraint (4.1), is presented.

Robustness to Polytopic Uncertainties

Observing that equation (4.16) is linear with respect to the x and y components of motion, and nonlinear with respect to its z components, walking motions up and down stairs were generated in [18] by adopting a Linear MPC scheme on the x and y components, that was made robust to the z component, considered as a polytopic uncertainty. A similar approach is here applied, including the additional capacity to use hand support.

Equation (4.16) can be reformulated as follows,

$$\boldsymbol{p}(\zeta_1, \zeta_2) = \boldsymbol{c}^{xy} - \zeta_1 \left(\ddot{\boldsymbol{c}}^{xy} - \frac{\boldsymbol{f}_e^{xy}}{m} \right) + \zeta_2 \boldsymbol{\Omega}\, \tilde{\boldsymbol{n}}^{xy}, \tag{4.27}$$

by introducing

$$\zeta_1 = \frac{m(c^z - s^z)}{m(\ddot{c}^z + g^z) - f_e^z}, \quad \zeta_2 = \frac{1}{m(\ddot{c}^z + g^z) - f_e^z}. \tag{4.28}$$

Note that this equation is linear with respect to ζ_1 and ζ_2. Let's consider now that these variables stay between some bounds:

$$0 \leq \underline{\zeta}_1 \leq \zeta_1 \leq \overline{\zeta}_1, \quad 0 \leq \underline{\zeta}_2 \leq \zeta_2 \leq \overline{\zeta}_2. \tag{4.29}$$

Ensuring that the constraint (4.20) is satisfied for all extreme values of ζ_1 and ζ_2, guarantees that it is satisfied for all values in between, by a simple convexity

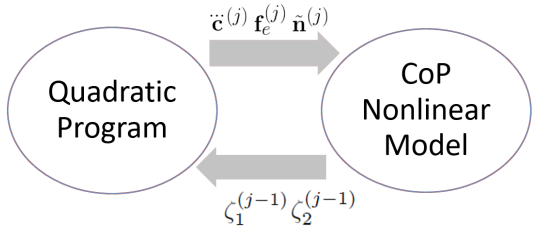


Figure 4.9: Scheme of the Nonlinear MPC approach.

argument:

$$\begin{gathered}\{\boldsymbol{p}(\underline{\zeta}_1, \underline{\zeta}_2),\, \boldsymbol{p}(\underline{\zeta}_1, \overline{\zeta}_2),\, \boldsymbol{p}(\overline{\zeta}_1, \underline{\zeta}_2),\, \boldsymbol{p}(\overline{\zeta}_1, \overline{\zeta}_2)\} \subset \mathcal{S}(\boldsymbol{s}_i^{xy}) \\ \Downarrow \\ \boldsymbol{p}(\zeta_1, \zeta_2) \in \mathcal{S}(\boldsymbol{s}_i^{xy}).\end{gathered} \tag{4.30}$$

Since the constraint (4.30) is defined with fixed, extreme values of ζ_1 and ζ_2, equation (4.27) does not involve anymore the z components of motion, and is simply linear with respect to its x and y components.

On the other hand, the combination of (4.28) and (4.29) imposes linear constraints on the z components of motion:

$$\underline{\zeta}_1(m(\ddot{c}^z + g^z) - f_e^z) \leq m(c^z - s^z) \leq \overline{\zeta}_1(m(\ddot{c}^z + g^z) - f_e^z), \tag{4.31}$$

$$\overline{\zeta}_2^{-1} \leq m(\ddot{c}^z + g^z) - f_e^z \leq \underline{\zeta}_2^{-1}. \tag{4.32}$$

This way, since the x and y components of motion satisfy the linear constraint (4.30) while the z components satisfy the linear constraints (4.31) and (4.32), the 3D motion satisfies the nonlinear constraint (4.20).

Constraining Newton Iterates

Each iteration of a Newton scheme involves computing the solution to a linear approximation of the nonlinear problem. The key idea of the approach is to constrain each of these iterations, to satisfy the linear constraints (4.30), (4.31) and (4.32), in order to make sure that the nonlinear constraint (4.20) is always satisfied exactly.

The value of the variables ζ_1 and ζ_2 over the whole MPC horizon is computed from the previous iterate $(j-1)$, and bounds for the next iterate (j) are defined

Table 4.1: Walking simulation parameters.

Sampling period	0.1 s
Length of the MPC horizon	1.6 s
Height of the stairs	0.1 m
Target height for the CoM	$\overline{c}_s^z = 0.88$ m
Mass of the robot	$m = 50$ kg
Size of the feet	$6{\times}12$ cm^2
Bounds on $\boldsymbol{f}_e$	$\underline{f}_e^{\upsilon} = -20$ N, $\overline{f}_e^{\upsilon} = 20$ N, $\forall \upsilon = \{x, y, z\}$
Bounds on $\tilde{\boldsymbol{n}}$	$\underline{\tilde{n}}^{\upsilon} = -20$ Nm, $\overline{\tilde{n}}^{\upsilon} = 20$ Nm, $\forall \upsilon = \{x, y\}$
Polytopic uncertainty	$\mu_1 = 0.02$ s^2, $\mu_2 = 0.001$ N^{-1}

accordingly in the following way:

$$\underline{\zeta}_1 = \zeta_1^{(j-1)} - k\mu_1 \leq \zeta_1^{(j)} \leq \zeta_1^{(j-1)} + k\mu_1 = \overline{\zeta}_1, \tag{4.33}$$

$$\underline{\zeta}_2 = \zeta_2^{(j-1)} - k\mu_2 \leq \zeta_2^{(j)} \leq \zeta_2^{(j-1)} + k\mu_2 = \overline{\zeta}_2, \tag{4.34}$$

with positive constants k, μ_1 and μ_2.

Fig. 4.9 shows a scheme of the Nonlinear MPC approach. At each iteration j of the Newton algorithm, the nonlinear expression of the CoP (4.16) is used to update the bounds on ζ_1 and ζ_2; thereafter, the quadratic problem, resulting from the linear approximation of the optimal control problem (4.26), is solved. The proposed Newton method is less constraining with respect to [18], since the admissible region for the CoP is no more a priori fixed.

4.4 Multi-Contact Walking Simulations

The proposed approach is evaluated on a walking motion up and down stairs as shown on Fig. 4.10, generated with parameters in Table 4.1, with the external force at the CoM, generated by a hand support at two predefined moments.

Generated Motion

Fig.s 4.11(a) and 4.11(b) show how the presence of an additional hand support during the steps up and down stairs significantly affects the motion of the CoM in the frontal plane (x, z) and in its (x, y) acceleration. In absence of external contacts, the weighted sum of o_1, o_2 and o_3 is considered as cost function. This cost naturally decreases when the external wrench is employed, since it represents an additional degree of freedom for the OCP. Fig. 4.12 shows the corresponding

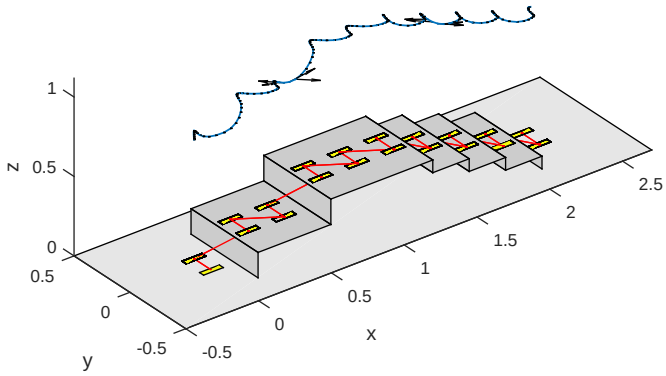
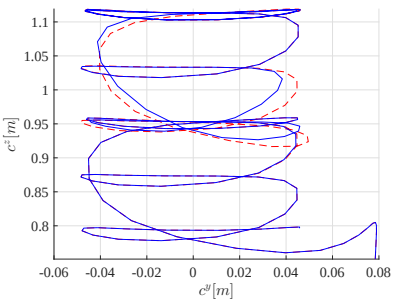
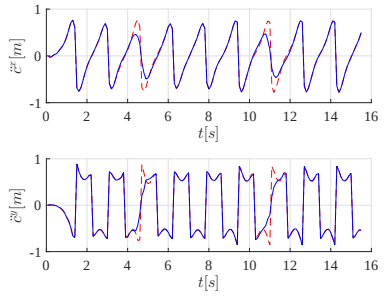


Figure 4.10: Snapshots of a walking motion climbing up and down stairs, generated online with the proposed Nonlinear MPC scheme. The yellow rectangles show the footsteps, the blue curve is the CoM of the robot, the red curve indicates the sequence of steps, and the black arrows represent the external force generated during the motion.



(a) Comparison between CoM trajectories in the frontal plane (x, z).



(b) Comparison between CoM acceleration trajectories in the transverse plane (x, y) w.r.t. time.

Figure 4.11: Comparison between CoM motion with (blue line) and without (dashed red line) additional hand support.

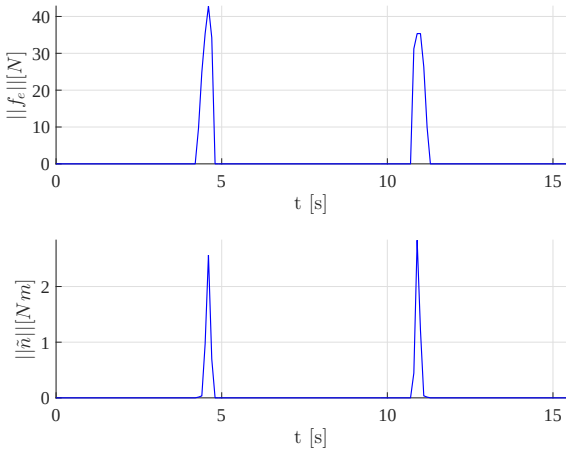
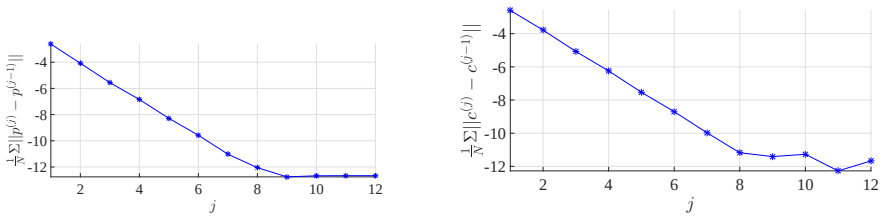


Figure 4.12: Evolution of the norm of the external force $\boldsymbol{f}_e$ (top) and torque $\tilde{\boldsymbol{n}}$ (bottom).



(a) Convergence of the CoP trajectory.

(b) Convergence of the CoM trajectory.

Figure 4.13: Semi-log graph showing the convergence of the CoP and CoM trajectories over the Newton iterations.

Table 4.2: Weights for the different objectives.

Distance between c^z and its reference	$w_1 = 1$
Distance between p and its reference	$w_2 = 1$
CoM jerk norm	$w_3 = 10^{-5}$
External force norm	$w_4 = 10^{-7}$
External torque norm	$w_5 = 10^{-4}$

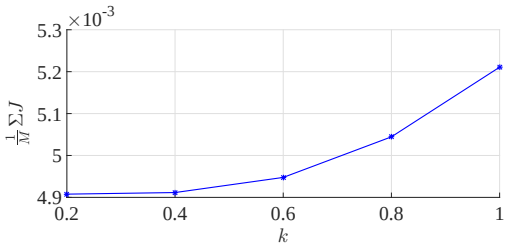


Figure 4.14: Mean value of the objective function over the whole motion w.r.t. the coefficient k. The objective function is impacted by the choice of the coefficient k, by as much as 6%.

evolution of the norm of the external force f_e and torque $\tilde{n}$ during the four predefined moments. Since a minimization of the norm of the wrench is performed, higher weights, w_4 and w_5, on f_e and $\tilde{n}$ generate lower values of the resultant wrench. High gains are chosen for the external torque, to reduce the rotational motion around the CoM. Simulation tests with variable height of the stairs reveal that for higher stairs (20 cm or 30 cm) a lower value of w_1 provides better performances. The proposed weights w_i are given in Table 4.2.

Fig. 4.13(a) and 4.13(b) depict the CoP and the CoM convergence tests respectively, over one preview horizon. This picture shows that the Newton scheme converges in few iterations. More interestingly, it is remarkable that the first iteration already provides a very good approximate solution, only a few millimeters away from the optimum. But in order to use safely such an approximate solution, the nonlinear constraint (4.20) has to be satisfied.

Nonlinear Constraint and Polytopic Uncertainty

Fig. 4.15 shows how the CoP is always kept inside the quadrilateral defined by the four extreme points $\boldsymbol{p}(\underline{\zeta}_1, \underline{\zeta}_2)$, $\boldsymbol{p}(\underline{\zeta}_1, \overline{\zeta}_2)$, $\boldsymbol{p}(\overline{\zeta}_1, \underline{\zeta}_2)$ and $\boldsymbol{p}(\overline{\zeta}_1, \overline{\zeta}_2)$, which are all kept inside the support polygon $\mathcal{S}(s_i^{xy})$. Four different computations are shown, with

a range of polytopic uncertainty $\{\mu_1, \mu_2\}$ multiplied by four different coefficients, $k \in \{1, 0.6, 0.4, 0.2\}$. Notice that the choice of a larger range leads to much more conservative constraints. It is remarkable that $\boldsymbol{p}(\underline{\zeta}_1, \underline{\zeta}_2)$, $\boldsymbol{p}(\underline{\zeta}_1, \overline{\zeta}_2)$, $\boldsymbol{p}(\overline{\zeta}_1, \underline{\zeta}_2)$ and $\boldsymbol{p}(\overline{\zeta}_1, \overline{\zeta}_2)$, vary within the preview horizon, since the boundary values of ζ_1 and ζ_2 are iteratively adapted. Therefore, the smaller quadrilaterals, showed in the bottom picture of Fig. 4.15, represent the less constraining case, since their four extreme points have more room to be placed inside the footprint by the optimization algorithm.

Fig. 4.14 shows how this more conservative constraint affects the value that can be reached for the objective function. It appears clearly that a smaller range leads to better results.

Note that the range of values for the polytopic uncertainties was chosen empirically, as inappropriate values can quickly lead to infeasible problems, what would require a relaxation of constraints.

Computational Efficiency

Solutions to the linear approximate models are computed with *qpOASES* [37], which is an open-source C++ implementation of the online active set strategy. Without using hotstart, on a relatively slow Intel Core i5-4200U @ 1.60GHz, the solutions are obtained in approximately 8 ms (average time over the whole trajectory generation). This result can certainly be improved significantly with hotstart. As a comparison, the computation time reported for a similar problem in [24] is 5.5 s for 85 iterations, which gives approximately 65 ms for each iteration, on a much faster CPU. Not only is this Newton scheme much safer, by providing iterates which are always feasible, it appears also to be an order of magnitude more efficient in terms of computation time.

4.4.1 Whole Body Motion

The proposed MPC scheme has been validated also considering whole body motion (see Fig. 4.16). Simulations are performed with an HRP-4 robot [56], which is controlled using the standard inverse dynamics approach [39, 97]. The proposed whole body motion controller employs PD-controllers to track reference trajectory of the CoM produced by MPC, and trajectories of the feet and hands generated using cubic polynomials. Positions of the hands are controlled only during the short intervals when they are approaching the contact points – during the rest of the simulation the reference configuration of the arms is maintained with PD-controllers instead. The whole body motion controller obeys contact friction constraints and constraints due to dynamics and kinematics of the robot. These constraints include the joint torque limits, which, however, are never reached in

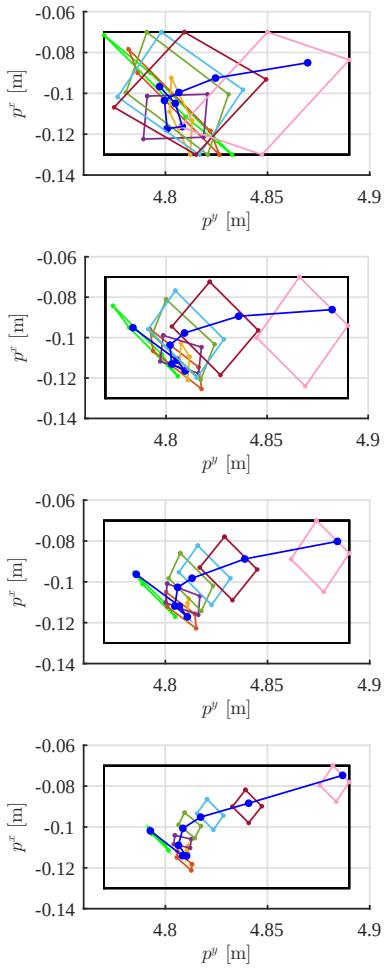


Figure 4.15: The trajectory of the CoP (blue line) is always kept inside the quadrilateral, which is kept inside the support polygon $\mathcal{S}(\boldsymbol{s}_i^{xy})$ (black rectangle). The range of polytopic uncertainty μ_1 and μ_2 is multiplied here by four different values of the coefficient k, from top to bottom: 1, 0.6, 0.4, 0.2.

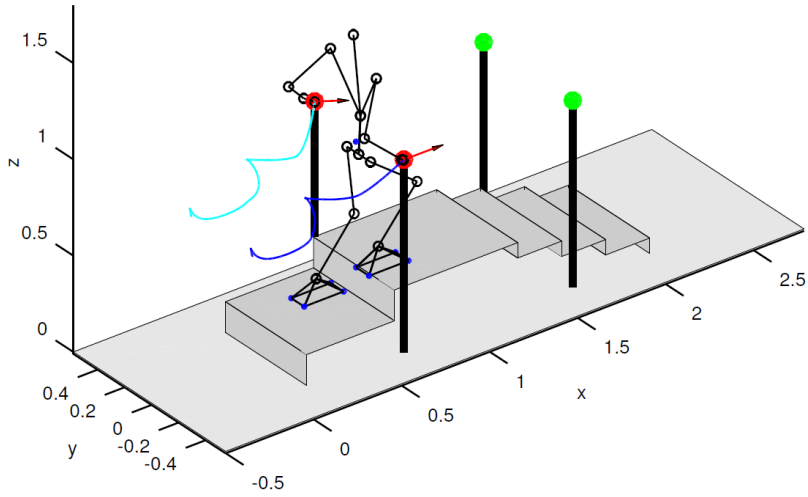


Figure 4.16: HRP-4 humanoid robot walking up stairs with hands support. 3D whole body motion generated online with the proposed NMPC scheme.

these simulations. Additionally, the generation of the reference external wrench, $\boldsymbol{w}_e \in \mathbb{R}^6$, by hand contacts is enforced using an equality constraint

$$\boldsymbol{w}_e = \Sigma_i \boldsymbol{T}_{ei} \boldsymbol{w}_{\boldsymbol{s}_i}, \tag{4.35}$$

where $\boldsymbol{T}_{ei}$ is a wrench transformation matrix [78], and $\boldsymbol{w}_{\boldsymbol{s}_i} \in \mathbb{R}^6$ is the wrench at the i-th contact point. In order to address discrepancy between the frequency of the whole body motion control and the MPC, linear interpolation of the reference values of the external wrench is employed.

The simulations can be visualized in a video[4], which consists of four parts: the first two illustrate walking on a flat ground, while the last two demonstrate walking up and down stairs. Due to the fact that the reference external wrench acts directly on the CoM, there is a relative freedom in choosing the contact configurations. In order to demonstrate this the reference external wrench uses a single hand contact in the first and third parts of the video, while in the second and fourth parts exactly the same external wrenches is produced using two hand contacts.

In the case of walking on a flat ground, tuning of the MPC and whole body motion controller amounts to finding kinematically feasible contact point positions, which can be relatively easy parameterized with the CoM position. More

[4] `https://youtu.be/NSDiSRPnxZ0`

difficulties are experienced while working on walking on stairs application. Most of them are related to infeasibility of the reference trajectories due to the mechanical limits of the knee (going up) and ankle (going down) joints. The latter appears to be a particularly severe limitation, which makes walking downstairs with flat feet a much more challenging task than going upstairs. The issues with the joint limits are alleviated by tuning the swing foot trajectories and the constraint on the maximal distance between the feet and the CoM, in addition, relatively low height of the steps (8 cm) is considered for downstairs climbing, while higher height (15 cm) is considered for upstairs climbing.

Joint torques with and without hand contacts are evaluated to measure possible reduction in energy consumption because of weight distribution. In case of slower motion, and including an additional objective in the cost function to increase the vertical component of the external force, the average of joint torques decreases, but high peaks appear. Even if the result is not strongly reliable because further tuning is required, these peaks are reduced with the integration of hands support.

4.5 Conclusion

This Chapter initially introduces the analogies between dynamic nonprehensile manipulation and the generation of multi-contact locomotion tasks for a humanoid robot, and motivates the investigation of motion planning and control techniques for multi-contact locomotion, stating the limitations of the existing approaches. The model predictive control is presented as a state of the art approach for the generation of walking patterns satisfying the condition of dynamic balance. Subsequently, building on the insight that variations in the vertical direction of motion are much less relevant than variations in the posterior-anterior directions for a walking task, a NMPC approach is proposed to generate a reference motion for a humanoid robot walking upstairs and downstairs while interacting with the environment with the hand contacts. With this purpose, a novel expression of the dynamic feasibility constraint is proposed, including variable height of the CoM, and modeling the external wrench due to the hand contacts with the additional supports. The designed NMPC has been assimilated to a Newton method ensuring dynamic feasibility at each iteration.

Drawing a conclusion, in this Chapter, following the same procedure as in [18], and constraining each iteration of the Newton method to fall within the limits of the specified polytopic uncertainty, the linear approximation of the NMPC is made robust to polytopic uncertainties. A specific form of the dynamic feasibility constraint is used, where a CoP is introduced for foot contacts on the ground, while other external forces are considered as a generic force and torque acting on the CoM, as proposed in [3]. In this way, this approach works as well for different

multi-contact situations, since the generated external wrench is not dependent on the specific contact configuration. The simulation of a walking motion up and down stairs with additional hand support and a critical discussion of the numerical results demonstrate the applicability of the method in case of whole body motion generation, with remarkable computational efficiency.

Chapter 5

Conclusion

5.1 Main Contribution

The main results on this Thesis are related to dynamic nonprehensile manipulation and legged locomotion tasks. In particular, new strategies for control of nonprehensile planar rolling manipulation, motion planning of 3D impact manipulation, and motion planning of multi-contact walking tasks, are proposed.

- Regarding control of nonprehensile planar rolling manipulation, the contribution is related to both modeling and control. A novel IDA-PBC for planar rolling manipulation tasks between two arbitrary shapes, considering both symmetric and asymmetric planar rolling manipulation, is designed. This approach differs from the standard feedback linearization where a linear dynamics is imposed at the expense of canceling all the nonlinear dynamics of the system. In addition, a systematic approach to simplify the IDA-PBC design is presented, transforming the set of PDEs resulting from the matching conditions in a set of algebraic equations. The control method has been applied to separable and non-separable Hamiltonian systems. Numerical tests confirm the validity of the control approach.

- A framework to deal with motion planning of impact manipulation is proposed in this manuscript. Firstly, the batting primitive performed by a semi-humanoid robot is here considered. The proposed nonlinear estimation techniques improve the control accuracy, while dealing with the real-time constraints. A coordinate-free, smooth, optimal motion plan, that minimizes the acceleration functional of the robot end-effector, is proposed. The batting paths are tracked by a semi-humanoid robot through a closed-loop kinematic inversion. Numerical tests are implemented to compare differ-

ent metrics to define the optimal impact configuration of the ball and the paddle system. Furthermore, the batting primitive is exploited to generate the dual-hand ball juggling task. An optimal planner for this nonprehensile dual-hand manipulation is given by a sequence of nonlinear minimum least squares problems. The approach is validated in the Matlab environment in connection with the V-Rep graphic simulation platform.

- Finally, for generating walking motions in multi-contact situations, a Newton method providing feasible iterates for a nonlinear MPC scheme is designed. It is based on a linear approximation which is made robust to polytopic uncertainties, constraining each iteration of the Newton method to fall within the limits of these polytopic uncertainties. The simulation of a walking motion up and down stairs with additional hand support demonstrates the applicability of the method, with remarkable computational efficiency. This approach works as well for different multi-contact situations, since the generated external wrench is not dependent on the specific contact configuration.

5.2 Future Research

Several ideas to improve the obtained results, and to investigate future lines of research, are detailed in the following.

- The proposed simplified IDA-PBC approach for planar rolling manipulation still presents a bottleneck related to the solution of the kinetic energy matching equation, that is based on the work [93]. An analytic analysis of the singular situation, that can appear in the solution of the kinetic energy matching equation, might lead to a more general result than the one obtained managing numerically such singularity. Furthermore, an experimental validation of the approach is necessary to confirm the validity of the proposed modeling and control methodology. As future work, also the application of the approach to other planar rolling tasks, like the butterfly task [71, 25, 113], might be investigated.

- Regarding motion planning for nonprehensile impact manipulation, as future work, the approach should be experimentally validated on a physical robotic prototype. Besides, a number of juggling patterns could be implemented exploiting again the batting primitive. The framework is now able to juggle by throwing with one ball, but it should be straightforward to add a second ball, assuming to have a more extended field of view.

- About the nonlinear MPC approach for multi-contact walking tasks, the proposed method relies on predefined footsteps and hand supports. Automatic footstep placement on horizontal surfaces could be included easily using the methods introduced in [48], but more complex cases of uneven ground and multi-contact situations require much more advanced planning methods [11, 29, 35, 117]. Nevertheless, the next step is to evaluate how a lexicographic approach could be introduced, to automatically decide when it is necessary to use the additional hand support, as in [109].

An interesting research line that is gaining more and more interest in the robotic research community is the investigation of a framework that may build a bridge between walking gaits and some other robotic field, such as object manipulation [10, 14, 19, 85].

One of the concept that underpin the research flow developed in this Thesis is the identification of connections between nonprehensile manipulation and dynamic walking. Both of them intrinsically involve: fast and hybrid dynamics, multiple non-coplanar contacts, and interaction between complex shapes. As future research, the investigation of possible mathematical mappings between dynamic nonprehensile manipulation and multi-contact dynamic walking might help in reusing some of the theory already developed within the robotic manipulation field into the robotic walking scenario, and vice-versa.

The proposed IDA-PBC approach, for example, could be applied to deal with the dynamic walking tasks. The control technique could be applied to the planar system which resembles the physics of walking, i.e. the inverted pendulum. Similarly, the proposed nonlinear MPC approach could be further investigated for dynamic manipulation tasks, conceiving the constrained optimal control problem that generates kinematically and dynamically feasible, robust, and smooth motions, in a task-specific way.

Bibliography

[1] E.W. Aboaf, S. Drucker, and C.G. Atkeson, *Task-level robot learning: Juggling a tennis ball more accurately*, IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 1989, pp. 1290–1295.

[2] L. Acosta, J.J. Rodrigo, J. Mendez, G. N. Marichal, and M. Sigut, *Ping-pong player prototype*, IEEE Robotics & Automation Magazine **10** (2003), no. 4, 44–52.

[3] D.J. Agravante, A. Cherubini, A. Sherikov, P.B. Wieber, and A. Kheddar, *Human-humanoid collaborative carrying*, working paper or preprint, 2016.

[4] A. Akbarimajd and M.N. Ahmadabadi, *Manipulation by juggling of planar polygonal objects using two 3-DoF manipulators*, IEEE/ASME international conference on advanced intelligent mechatronics (Zurich, CH), 2007, pp. 1–6.

[5] R. L. Andersson, *A robot ping-pong player: experiment in real-time intelligent control*, MIT press, 1988.

[6] ———, *Aggressive trajectory generator for a robot ping-pong player*, IEEE Control Systems Magazine **9** (1989), no. 2, 15–21.

[7] Y. Aoustin and A.M. Formal'skii, *Beam-and-ball system under limited control: Stabilization with large basin of attraction*, IEEE American Control Conference (St Louis, MO, USA), 2009, pp. 555–560.

[8] Y. Aoustin and A.M. Formal'sky, *An original circular ball-and-beam system: stabilization strategy under saturating control with large basin of attraction*, European Control Conference (Kos, GR), 2007, pp. 4833–4838.

[9] T. Aoyama, T. Takaki, Q. Gu, and I Ishii, *Control scheme of nongrasping manipulation based on virtual connecting constraint*, IEEE International Conference on Robotics and Automation (Stockholm, SE), 2016, pp. 3819–3824.

[10] T. Asfour, J. Borras, C. Mandery, P. Kaiser, E.E. Aksoy, and M. Grotz, *On the dualities between grasping and whole-body loco-manipulation tasks*, International Symposium on Robotics Research (Genova, I), 2015.

[11] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, *Model preview control in multi-contact motion-application to a humanoid robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Chicago,IL, USA), 2014, pp. 4030–4035.

[12] G. Batz, K.K. Lee, D. Wollherr, and M. Buss, *Robot basketball: A comparison of ball dribbling with visual and force/torque feedback*, IEEE International Conference on Robotics and Automation (Kobe, J), 2009, pp. 514–519.

[13] G. Batz, U. Mettin, A. Schmidts, M. Scheint, D. Wollherr, and A.S. Shiriaev, *Ball dribbling with an underactuated continuous-time control phase: Theory & experiments*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, TW, 2010, pp. 2890–2895.

[14] B. Beigzadeh, M.N. Ahmadabadi, A. Meghdari, and A. Akbarimajd, *A dynamic object manipulation approach to dynamic biped locomotion*, Robotics and Autonomous Systems **56** (2008), no. 7, 570–582.

[15] A. Bicchi and A. Marigo, *Dexterous grippers: Putting nonholonomy to work for fine manipulation*, The International Journal of Robotics Research **21** (2002), no. 5-6, 427–442.

[16] J.J. B. Biemond, N. Van de Wouw, W.H. Heemels, and H. Nijmeijer, *Tracking control for hybrid systems with state-triggered jumps*, IEEE Transactions on Automatic Control **58** (2013), no. 4, 876–890.

[17] J. Borras and T. Asfour, *A whole-body pose taxonomy for loco-manipulation tasks.*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Hamburg, DE), 2015, pp. 1578–1585.

[18] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.B. Wieber, *A robust linear MPC approach to online generation of 3D biped walking motion*, IEEE-RAS International Conference on Humanoid Robots (Seoul, KR), 2015, pp. 595–601.

[19] T. Bretl and S. Lall, *Testing static equilibrium for legged robots*, IEEE Transactions on Robotics **24** (2008), no. 4, 794–807.

[20] B. Brogliato and A.Z. Rio, *On the control of complementary-slackness juggling mechanical systems*, IEEE Transactions on Automatic Control **45** (2000), no. 2, 235–246.

[21] M. Buehler, D.E. Koditschek, and P.J. Kindlmann, *Planning and control of robotic juggling and catching tasks*, The International Journal of Robotics Research **13** (1994), no. 2, 101–118.

[22] M. Buhler, D.E. Koditschek, and P.J. Kindlmann, *A family of robot control strategies for intermittent dynamical environments*, Control Systems Magazine, IEEE **10** (1990), no. 2, 16–22.

[23] S. Caron and A. Kheddar, *Multi-contact walking pattern generation based on model preview control of 3d com accelerations*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 550–557.

[24] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, *A versatile and efficient pattern generator for generalized legged locomotion*, IEEE International Conference on Robotics and Automation (Stockholm, SE), 2016, pp. 3555–3561.

[25] M. Cefalo, L. Lanari, and G. Oriolo, *Energy-based control of the butterfly robot*, International IFAC Symposium on Robot Control (Bologna, I), 2006, pp. 1–6.

[26] P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano, *Robotic ball catching with an eye-in-hand single-camera system*, IEEE Transactions on Control Systems Technology **23** (2015), no. 5, 1657–1671.

[27] H. Dai, *Robust multi-contact dynamical motion planning using contact wrench set*, Ph.D. thesis, Massachusetts Institute of Technology, 2016.

[28] H. Dai and R. Tedrake, *Planning robust walking motion on uneven terrain via convex optimization*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 579–586.

[29] R. Deits and R. Tedrake, *Footstep planning on uneven terrain with mixed-integer convex optimization*, IEEE-RAS International Conference on Humanoid Robots (Madrid, ES), 2014, pp. 279–286.

[30] M. Diehl, *Real-time optimization for large scale nonlinear processes*, Ph.D. thesis, Heidelberg University, 2001.

[31] M.P. do Carmo, *Riemannian Geometry*, Mathematics (Boston, Mass.), Birkhäuser, 1992.

[32] A. Donaire, F. Ruggiero, L. R. Buonocore, V. Lippiello, and B. Siciliano, *Passivity-based control for a rolling-balancing system: The nonprehensile disk-on-disk*, IEEE Transactions on Control System Technology (2016), to appear.

[33] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and control of complex physical systems: The port-Hamiltonian approach*, Springer Science & Business Media, 2009.

[34] V. Duindam and S. Stramigioli, *Modeling the kinematics and dynamics of compliant contact*, IEEE International Conference on Robotics and Automation (Taipei, TW), 2003, pp. 4029–4034.

[35] A. Escande, A. Kheddar, and S. Miossec, *Planning contact points for humanoid robots*, Robotics and Autonomous Systems **61** (2013), no. 5, 428–442.

[36] E. Farnioli, M. Gabiccini, and A. Bicchi, *Toward whole-body loco-manipulation: Experimental results on multi-contact interaction with the walk-man robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Daejeon,KR), 2016, pp. 1372–1379.

[37] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl, *qpOASES: A parametric active-set algorithm for quadratic programming*, Mathematical Programming Computation **6** (2014), no. 4, 327–363.

[38] F. Fontana, P. Reist, and R. D'Andrea, *Control of a swinging juggling robot*, IEEE European Control Conference (Zurich, CH), 2013, pp. 2317–2322.

[39] Y. Fujimoto and A. Kawamura, *Proposal of biped walking control based on robust hybrid position/force control*, IEEE International Conference on Robotics and Automation (Minneapolis, MN,USA), vol. 3, Apr 1996, pp. 2724–2730 vol.3.

[40] F. Gomez-Estern, R. Ortega, F.R. Rubio, and J. Aracil, *Stabilization of a class of underactuated mechanical systems via total energy shaping*, IEEE Conference on Decision and Control (Orlando, FL, USA), 2001, pp. 1137–1143.

[41] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, *Using probabilistic movement primitives for striking movements*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 502–508.

[42] F. Gordillo, J. Aracil, and F. Gómez-Estern, *Stabilization of autonomous oscillations and the hopf bifurcation in the ball and beam*, IEEE Conference on Decision and Control (Las Vegas, NV, USA), 2002, pp. 3924–3925.

[43] S. Haddadin, K. Krieger, M. Kunze, and A. Albu-Schaffer, *Exploiting potential energy storage for cyclic manipulation: An analysis for elastic dribbling*

with an anthropomorphic robot, IEEE/RSJ International Conference on Intelligent Robots and Systems (San Francisco, CA, USA), 2011, pp. 1789–1796.

[44] V. Hagenmeyer, S. Streif, and M. Zeitz, *Flatness-based feedforward and feedback linearisation of the ball & plate lab experiment*, IFAC-Symposium on Nonlinear Control Systems (Stuttgart, DE), 2004.

[45] K. Harada, H. Hirukawa, F. Kanehiro, K. Fujiwara, K. Kaneko, S. Kajita, and M. Nakamura, *Dynamical balance of a humanoid robot grasping an environment*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Sendai, J), 2004, pp. 1167–1173.

[46] J. Hauser, S. Sastry, and P. Kokotovic, *Nonlinear control via approximate input-output linearization: The ball and beam example*, IEEE Transactions on automatic control **37** (1992), no. 3, 392–398.

[47] ———, *Nonlinear control via approximate input-output linearization: The ball and beam example*, IEEE Transactions on Automatic Control **37** (1992), no. 3, 392–398.

[48] A. Herdt, H. Diedam, P.B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, *Online walking motion generation with automatic footstep placement*, Advanced Robotics **24** (2010), no. 5-6, 719–737.

[49] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, *Trajectory generation for multi-contact momentum control*, IEEE-RAS International Conference on Humanoid Robots (Seoul, KR), 2015, pp. 874–880.

[50] D. Hristu Varsakelis, *The dynamics of a forced sphere plate mechanical system*, IEEE Transactions on Automatic Control **46** (2001), no. 5, 678–686.

[51] Y. Huang, D. Büchler, B. Schölkopf, and J. Peters, *Jointly learning trajectory generation and hitting point prediction in robot table tennis*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 650–655.

[52] A.M. Johnson, G.C. Haynes, and D.E. Koditschek, *Standing self-manipulation for a legged robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Vilamoura, PT), 2012, pp. 272–279.

[53] S. Kajita and B. Espiau, *Legged robots*, The hadbook of robotics, Springer, 2008.

[54] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, *Biped walking pattern generation by using preview control of zero-moment point*, IEEE International Conference on Robotics and Automation (Taipei, TW), 2003, pp. 1620–1626.

[55] ———, *Resolved momentum control: Humanoid motion planning based on the linear and angular momentum*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas, NV,USA), 2003, pp. 1644–1650.

[56] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira, *Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body*, IEEE/RSJ International Conference on Intelligent Robots and Systems (San Francisco, CA, USA), 2011, pp. 4400–4407.

[57] B. Kiss, J. Lévine, and B. Lantos, *On motion planning for robotic manipulation with permanent rolling contacts*, International Journal of Robotics Research **21** (2002), no. 5–6, 443–461.

[58] K. Koyanagi, H. Hirukawa, S. Hattori, M. Morisawa, S. Nakaoka, K. Harada, and S. Kajita, *A pattern generator of humanoid robots walking on a rough terrain using a handrail*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Nice, FR), 2008, pp. 2617–2622.

[59] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Souères, and K. Mombaur, *Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations*, IEEE-RAS International Conference on Humanoid Robots (Seoul, KR), 2015, pp. 684–689.

[60] C. Lanczos, *The variational principles of mechanics*, Courier Corporation, 2012.

[61] K.K. Lee, G. Batz, and D. Wollherr, *Basketball robot: Ball-on-plate with pure haptic information*, IEEE International Conference on Robotics and Automation (Pasadena, CA, USA), 2008, pp. 2410–2415.

[62] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, *Generation of whole-body optimal dynamic multi-contact motions*, The International Journal of Robotics Research **32** (2013), no. 9-10, 1104–1119.

[63] V. Lippiello and F. Ruggiero, *3D monocular robotic ball catching with an iterative trajectory estimation refinement*, IEEE International Conference on Robotics and Automation (Saint Paul, MN, USA), 2012, pp. 3950–3955.

[64] V. Lippiello, F. Ruggiero, and B. Siciliano, *The effects of shapes in input-state linearization for stabilization of nonprehensile planar rolling dynamic manipulation*, IEEE Robotics and Automation Letters **1** (2016), no. 1, 492–499.

[65] C. Liu, Y. Hayakawa, and A. Nakashima, *Racket control and its experiments for robot playing table tennis*, IEEE International Conference on Robotics and Biomimetics (Guangzhou, CN), 2012, pp. 241–246.

[66] M.I.A. Lourakis, *levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++*, [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004, [Accessed on 31 Jan. 2005.].

[67] K.M. Lynch and C.K. Black, *Recurrence, controllability, and stabilization of juggling*, IEEE Transactions on Robotics and Automation **17** (2001), no. 2, 113–124.

[68] K.M. Lynch and M.T. Mason, *Dynamic underactuated nonprehensile manipulation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Osaka, J), 1996, pp. 889–896.

[69] ———, *Stable pushing: Mechanics, controllability, and planning*, The International Journal of Robotics Research **15** (1996), no. 6, 533–556.

[70] ———, *Dynamic nonprehensile manipulation: Controllability, planning, and experiments*, The International Journal of Robotics Research **18** (1999), no. 1, 64–92.

[71] K.M. Lynch, N. Shiroma, H. Arai, and K. Tanie, *The roles of shape and motion in dynamic manipulation: The butterfly example*, IEEE International Conference on Robotics and Automation (Lueven, BE), 1998, pp. 1958–1963.

[72] D. Mansour, A. Micaelli, A. Escande, and P. Lemerle, *A new optimization based approach for push recovery in case of multiple noncoplanar contacts*, IEEE-RAS International Conference on Humanoid Robots (Bled, SI), 2011, pp. 331–338.

[73] M.T. Mason and K.M. Lynch, *Dynamic manipulation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Tokyo, J), 1993, pp. 152–159.

[74] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, *Constrained model predictive control: Stability and optimality*, Automatica **36** (2000), no. 6, 789–814.

[75] D.J. Montana, *The kinematics of contact and grasp*, The International Journal of Robotics Research **7** (1988), no. 3, 17–32.

[76] K. Muelling, *Modeling and learning of complex motor tasks: A case study with robot table tennis*, Ph.D. thesis, Technische Universität, 2013.

[77] M. Müller, S. Lupashin, and R. D'Andrea, *Quadrocopter ball juggling*, IEEE/RSJ International Conference on Intelligent Robots and Systems (San Francisco, CA, USA), 2011, pp. 5113–5120.

[78] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*, CRC press, 1994.

[79] A. Nakashima, D. Ito, and Y. Hayakawa, *An online trajectory planning of struck ball with spin by table tennis robot*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Besançon, FR), 2014, pp. 865–870.

[80] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa, *Modeling of rebound phenomenon of a rigid ball with friction and elastic effects*, IEEE American Control Conference (Baltimore, MD, USA), 2010, pp. 1410–1415.

[81] R. Naldi and R.G. Sanfelice, *Passivity-based control for hybrid systems with applications to mechanical systems exhibiting impacts*, Automatica **49** (2013), no. 5, 1104–1116.

[82] R. Ortega, A. Donaire, and J.G. Romero, *Passivity - based control of mechanical systems*, Feedback Stabilization of Controlled Dynamical Systems (2016), to appear.

[83] R. Ortega, M.W. Spong, F. Gómez-Estern, and G. Blankenstein, *Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment*, IEEE Transactions on Automatic Control **47** (2002), no. 8, 1218–1233.

[84] R. Ortega, A. Van Der Schaft, B. Maschke, and G. Escobar, *Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems*, Automatica **38** (2002), no. 4, 585–596.

[85] M.A. Ott, C.and Roa and G. Hirzinger, *Posture and balance control for biped robots based on contact force optimization*, IEEE-RAS International Conference on Humanoid Robots (Bled, SI), 2011, pp. 26–33.

[86] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, *A convex model of momentum dynamics for multi-contact motion generation*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 842–849.

[87] M. Posa and R. Tedrake, *Direct trajectory optimization of rigid body dynamical systems through contact*, Algorithmic foundations of robotics, Springer, 2013, pp. 527–542.

[88] P. Reist and R. D'Andrea, *Bouncing an unconstrained ball in three dimensions with a blind juggling robot*, IEEE International Conference on Robotics and Automation (Kobe, J), 2009, pp. 1774–1781.

[89] ———, *Design and analysis of a blind juggling robot*, IEEE Transactions on Robotics **28** (2012), no. 6, 1228–1243.

[90] E. Rohmer, S.P.N. Singh, and M. Freese, *V-REP: A versatile and scalable robot simulation framework*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Tokyo, J), 2013, pp. 1321–1326.

[91] R. Ronsse, P. Lefevre, and R. Sepulchre, *Rhythmic feedback control of a blind planar juggler*, IEEE Transactions on Robotics **23** (2007), no. 4, 790–802.

[92] R. Ronsse and R. Sepulchre, *Feedback control of impact dynamics: The bouncing ball revisited*, IEEE Conference on Decision and Control (San Diego, CA, USA), 2006, pp. 4807–4812.

[93] M. Ryalat and D.S. Laila, *A simplified IDA–PBC design for underactuated mechanical systems with applications*, European Journal of Control **27** (2016), 1–16.

[94] J.C. Ryu, F. Ruggiero, and K. M. Lynch, *Control of nonprehensile rolling manipulation: Balancing a disk on a disk*, IEEE International Conference on Robotics and Automation (St. Paul, MN, USA), 2012, pp. 3232–3237.

[95] J.C. Ryu, F. Ruggiero, and K. M. Lynch, *Control of nonprehensile rolling manipulation: Balancing a disk on a disk*, IEEE Transactions on Robotics **29** (2013), no. 5, 1152–1161.

[96] K. Ryu and Y. Oh, *Balance control of ball-beam system using redundant manipulator*, IEEE International Conference on Mechatronics (Instanbul, TR), 2011, pp. 403–408.

[97] L. Saab, O.E. Ramos, F. Keith, N. Mansard, P. Souères, and J.Y. Fourquet, *Dynamic whole-body motion generation under rigid contacts and other unilateral constraints*, IEEE Transactions on Robotics **29** (2013), no. 2, 346–362.

[98] R. G. Sanfelice, A.R. Teel, and R. Sepulchre, *A hybrid systems approach to trajectory tracking control for juggling systems*, IEEE Conference on Decision and Control (New Orleans, LA, USA), 2007, pp. 5282–5287.

[99] S. Satpute, R. Mehra, F. Kazi, and N.M. Singh, *Geometric–PBC approach for control of circular ball and beam system*, International Symposium on Mathematical Theory of Networks and Systems (Groningen, NL), 2014, pp. 1238–1243.

[100] S. Schaal and C.G. Atkeson, *Open loop stable control strategies for robot juggling*, IEEE International Conference on Robotics and Automation (Atlanta, GE, USA), 1993, pp. 913–918.

[101] ———, *Memory-based robot learning*, IEEE International Conference on Robotics and Automation (San Diego, CA, USA), 1994, pp. 2928–2933.

[102] ———, *Robot juggling: Implementation of memory-based learning*, Control Systems **14** (1994), no. 1, 57–71.

[103] T. Senoo, A. Namiki, and M. Ishikawa, *Ball control in high-speed batting motion using hybrid trajectory generator*, IEEE International Conference on Robotics and Automation (Orlando, FL, USA), 2006, pp. 1762–1767.

[104] D. Serra, *Robot control for nonprehensile dynamic manipulation tasks*, International Conference on Informatics in Control, Automation and Robotics, Doctoral Constortium (Lisbon,PT), 2016, pp. 3–12.

[105] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P.B. Wieber, *A Newton method with always feasible iterates for nonlinear model predictive control of walking in a multi-contact situation*, IEEE-RAS International Conference on Humanoid Robots (Cancun, MX), 2016, pp. 932–937.

[106] D. Serra, F. Ruggiero, V. Lippiello, and B. Siciliano, *A nonlinear least squares approach for nonprehensile dual-hand robotic ball juggling*, World Congress of the International Federation of Automatic Control (Tolouse, FR), 2017, p. to appear.

[107] D. Serra, A. C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, *Time–optimal paths for a robotic batting task*, Springer Lecture Notes in Electrical Engineering (2017), to appear.

[108] D. Serra, A.C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, *An optimal trajectory planner for a robotic batting task: The table tennis example*, International Conference on Informatics in Control, Automation and Robotics (Lisbon, PT), 2016, pp. 90–101.

[109] A. Sherikov, D. Dimitrov, and P.B. Wieber, *Balancing a humanoid robot with a prioritized contact force distribution*, IEEE-RAS International Conference on Humanoid Robots (Seoul, KR), 2015, pp. 223–228.

[110] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer-Verlag, 2010.

[111] A. Simpkins and E. Todorov, *Complex object manipulation with hierarchical optimal control*, IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL) (Paris, FR), 2011, pp. 338–345.

[112] Y. Sun, R. Xiong, Q. Zhu, J. Wu, and J. Chu, *Balance motion generation for a humanoid robot playing table tennis*, IEEE-RAS International Conference on Humanoid Robots (Bled, SI), 2011, pp. 19–25.

[113] M. Surov, A. Shiriaev, L. Freidovich, S. Gusev, and L. Paramonov, *Case study in non-prehensile manipulation: Planning perpetual rotations for "Butterfly" robot*, IEEE International Conference on Robotics and Automation (Seattle, WA, USA), 2015, pp. 1484—1489.

[114] T. Tabata and Y. Aiyama, *Passing manipulation by 1 degree-of-freedom manipulator-catching manipulation of tossed object without impact*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas, NV, USA), vol. 3, 2003, pp. 2920–2925.

[115] A. Teel and L. Praly, *Tools for semiglobal stabilization by partial state and output feedback*, SIAM Journal on Control and Optimization **33** (1995), no. 5, 1443–1488.

[116] X. Tian, J.H. Koessler, and R.G. Sanfelice, *Juggling on a bouncing ball apparatus via hybrid control*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Tokyo, J), 2013, pp. 1848–1853.

[117] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettré, *A reachability-based planner for sequences of acyclic contacts in cluttered environments*, International Symposium on Robotics Research (Sestri Levante, I), 2015.

[118] P.B. Wieber, *Trajectory free linear model predictive control for stable walking in the presence of strong perturbations*, IEEE-RAS International Conference on Humanoid Robots (Genova, I), 2006, pp. 137–142.

[119] P.B. Wieber, R. Tedrake, and S. Kuindersma, *Modeling and control of legged robots*, The hadbook of robotics, Springer, 2015.

[120] M. Zefran, V. Kumar, and C.B. Croke, *On the generation of smooth three-dimensional rigid body motions*, IEEE Transactions on Robotics and Automation **14** (1998), no. 4, 576–589.

[121] Y. Zhang, R. Xiong, Y. Zhao, and J. Chu, *An adaptive trajectory prediction method for ping-pong robots*, Intelligent Robotics and Applications, Springer, 2012, pp. 448–459.