Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Institute for Cognitive Systems (ICS)

Energy based limit cycle control for elastically actuated biped robots

Gianluca Garofalo

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Eckehard Steinbach
Prüfer der Dissertation:	
	1. Jun.-Fellow Dr.-Ing. Christian Ott
	2. Prof. Dr. Majid Zamani
	3. Prof. Dr. Bruno Siciliano, Università di Napoli Federico II / Italien

Die Dissertation wurde am 27.09.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 18.03.2017 angenommen.

Preface

Thanks for starting reading this monograph. With the following pages, I will bring you through some of the results of a bit more than five years of work. As anyone who spent a similar amount of effort might know, this journey was full of experiences and a roller coaster of feelings; that kind of things that make one having the impression that a lot has been learned in the meanwhile. This page especially is the last ones that I have written. The last page that in a running metaphor looks a lot like an approaching finishing line, close enough that one stretches to touch it. It is a ritual and I will do my part.

I start thanking the two people who are most to blame for having me working in robotics. Prof. Bruno Siciliano, whose passion while teaching the very first robotic class I ever attended when I was still an undergrad student in Naples, certainly made tick my passion for this discipline and Dr. Christian Ott who gave me the chance to work at the DLR when I had just finished my studies and even my English was at a basic level. I have the privilege to have both of them evaluating my work once again. All the other colleagues are also acknowledged and sincerely thanked for the often enlightening discussions. Among them, for sure a special mention goes to my office mate Johannes Englsberger. His help during these years has gone way further than exchanging ideas about control theory and robotics.

Finally, I want to thank Bice, Massimo and Marco (the three people who had me around everyday for more or less twenty-three years) as well as all the rest of my family, my friends and my girlfriend Catalina because, although they could never help me with a proof or an experiment, they were invaluably supportive and had shown such an interest in things that had bored anyone else to death. Their constant questions about my work and their will to make me explain what I was doing, turned out to be the perfect chance to keep an eye on the "big picture", not to mention the big motivation that it was. They have managed to stand both my relaxed and deadline stressed version without any apparent problem. For that, they must be indeed very talented.

Munich, September 20 Gianluca Garofalo

Abstract

Although at date humanoid robots are still not ready to be used alongside humans in everyday life, clearly legged locomotion is a key feature for robot to posses if they are expected to extensively interact in an environment designed for humans. Legged robots are challenging systems to control due to their hybrid nature, underactuation and unilateral constraints. Additionally, highly dynamic tasks as jumping and running require performant actuation principles. Elastic joints are, therefore, often used due to their capability to store energy. The notion of elastic joint robots has a long history in robotics. Nevertheless, while the joint elasticity was originally treated as a disturbance of the rigid-body dynamics and therefore it had to be minimized during both the mechanical and control design processes, more recent drive concepts like series elastic actuators or variable impedance actuators deliberately introduce elasticity for implementing torque control, increasing physical robustness, or reaching high output velocities. As a consequence, legged robots which use this type of actuation principle have another source of underactuation introduced in the system model.

The main contribution of the thesis is the design of a dynamic state feedback to produce asymptotically stable limit cycles for an elastically actuated robot. The controller produces the desired periodic motion recruiting directly the springs contained in the joints, as opposed to classic controllers for elastic joint robots, which are designed to minimize the effects of the elasticity. To the best of the knowledge of the author, this is the first controller to exploit the presence of elasticity. In an ideal nominal case of frictionless system, the control law will stop the motors after a transient and the oscillations are sustained by the springs. The controller is able to achieve this result by regulating an energy function of the closed loop system to a desired constant value. The energy function is defined using the stiffness of the real springs in the joints, beside the gravitational potential energy and the kinetic energy of the links. The limit cycle of the closed loop system is itself defined by the desired value of the energy function. As a consequence, another interesting feature of the controller is the capability to regulate the robot to a desired equilibrium point (in alternative to the limit cycle), by simply setting to a nonpositive value the desired energy.

The final structure of the controller is obtained after some intermediate steps. While a dynamic state feedback is employed for the multi-joint case, a simple static state feedback is used for regulating the energy of a single elastic joint. Nevertheless, the limitation to a single joint is not required if the controller has to be used for regulation to an equilibrium point. The design of this simpler controller represents another contribution of the thesis.

Before the design of the controllers for elastically actuated robots, the simpler, more common case of rigidly actuated robots is considered. Also for this case, a control law is designed to

solve the problem of orbital stabilization. Similarly, the controller produces the asymptotically stable limit cycle by regulating an energy function. This time, a virtual potential introduced by the controller is used in the definition of the energy, since no elastic elements are present in the system. This contribution is also further extended to a robust version of the controller.

The stability analysis of all the energy based controllers relies on semidefinite Lyapunov functions. A contribution is made concerning this topic by extending the theorems for the stability analysis of equilibrium points to the case of generic bounded invariant sets, i.e. including also limit cycles.

Elasticity specific to the case of legged locomotion is also considered. Several works can be found in the biomechanics field in which the role of elasticity is shown to be crucial to obtain the main features of a locomotion pattern. The spring loaded inverted pendulum (SLIP) is a very simple, yet powerful way to extrapolate center of mass trajectories and vertical ground reaction forces profiles of a wide set of gaits. A contribution of the thesis is to map this simple one mass model on a multi-body robot in order to obtain a walking pattern and insights on the use of elasticity in locomotion. Using an even simpler version of the SLIP model, an energy based approach is shown to be effective to produce a jumping pattern. Another contribution of this work is to incorporate the dynamic state feedback for elastically actuated robots in a state of the art optimization framework, originally designed for balancing humanoid robots. In this way, it is possible to command the robot to initiate and stop a sequence of jumps.

Finally, the last contribution of the thesis is the design of a control law for regulating the end effector of a floating based robot in presence of nonzero generalized momentum. Although the controller is considered for a satellite with a robotic arm, it can be seen as a first result towards improved control of the feet of a humanoid robot during the flight phase. A satellite, in fact, shares the same structure of the dynamic model as a humanoid robot, but the absence of gravity simplifies the problem. This results rely on the findings of this work on the structure of the dynamic model of a floating base system.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The interest of the robotic community in locomotion is already decades old, starting with the pioneering works of Vukobratovic [VB05], Kato [KOK+73], Raibert [Rai86] as well as of McGeer [McG90]. In some sense these authors laid down a dichotomy in the field: on one side fully actuated robots based, for example, on the concept of the zero-moment point (ZMP) and on the other purely passive mechanical systems or with very little actuation based on cyclic behavior and orbital stability. Furthermore, in the biomechanics community Blickhan [Bli89] started proposing template models which can reproduce locomotion patterns by means of elastic elements; not too differently from the pogo stick principle used in the first hopping robots developed by Raibert at the MIT. It was becoming quite clear that the role of the actuation is of paramount importance in highly dynamical systems like legged robots; as intuition might suggest when thinking of tasks like running or jumping.

It is not surprising then if legged robots have been both the application and motivation for the design of innovative actuation systems. Even less of a surprise is that often these new actuator designs include elastic elements in order to achieve efficiency through energy storage (which is clearly inspired by the human capabilities of storing energy in tendons and muscles). Different uses for springs in legged locomotion [Ale90] show that there are two main reasons for considering their presence: energy storage (which means less energy consumption and unwanted heat production) and smaller force at the impact of the foot on the ground (which decrease the risk of damages). The notion of elastic joint robots has also a long history in robotics. Nevertheless, while in the seminal publications by Spong [Spo87] and De Luca [De 88] in the late 80s and early 90s, the joint elasticity was originally treated as a disturbance of the rigid-body dynamics, more recent drive concepts like series elastic actuators [PW95] or variable impedance actuators [WEH11] deliberately introduce elasticity for implementing torque control, increasing physical robustness, or reaching high output velocities.

A common challenge to both fields is coping with the underactuation of the system, which clearly increases the complexity of the controller design. A schematic of the problems can be found in Fig. 1.1. For legged robots the underactuation is a direct consequence of their base free nature, as opposed to the classical base fixed robot manipulators. Legged robots are free systems in space subject to nonconstant unilateral contacts provided by the interaction with the floor. The torques provided by the motors in the joints and the reaction forces exerted by the floor cannot be all directly and arbitrarily chosen by the controller, from which the underactuation of the system

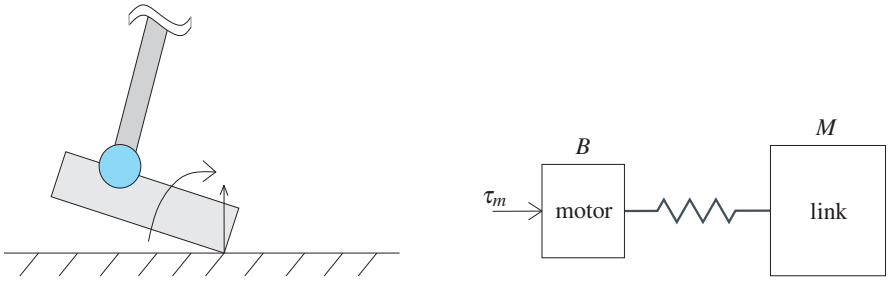

Figure 1.1.: Schematic of the sources of the underactuation.

arise. If the robot can be considered as having a foot motionless during a specific phase of its task, then the model is that of a classical manipulator. The system can even become overactuated if more than one foot is rigidly in contact with the floor, in which case a force distribution problem must be solved. For elastic joint robots, on the other hand, the underactuation is due to the presence of the elastic element itself, which couples the motor to the link. Although the goal is to control the position of the link, the latter is affected by the action of the spring which cannot be directly and arbitrarily chosen by the controller.

A short overview of the existing methods for dealing both with the underactuation problem in legged robots and elastic joint robots is presented.

1.1. The underactuation in biped robots

Biped robots can be modeled as nonsmooth hybrid systems. The nonsmoothness arises from the occurrence of the impacts, while the word hybrid refers to the mix of both a continuous and discrete dynamics due to the presence of algebraic constraints modeling the restriction imposed on the state by the floor. When any of the constraints is active, the model describing the system is a differential-algebraic equation (DAE) [Bro99].

Taking into account the underactuation of a biped robot is a challenging task. Therefore, often simplified model are used for the design of the controller. The linear inverted pendulum model is often used in conjunction with the ZMP in order to prevent the robot from tilting. In [EOR+11], for example, a ZMP controller provides the center of mass (CoM) reference trajectory using the values of the measured and the desired ZMP. The latter is computed stabilizing the unstable part of the CoM dynamics, referred to as capture point in [PCDG06]. Keeping the ZMP within the support polygon ensures that the robot is not tilting over. Nevertheless, the limitations due to the friction cone are not taken into account.

The best solution to deal with the unilateral constraint imposed by the floor is to set up an optimization problem. The latter can be used just at the planning level, for parameterized trajectories or in an online scheme. In the first case, the joint trajectories are expressed typically by splines, whose coefficients are determined solving an optimization problem that guaranties the satisfaction of all the equality and inequality constraints [DCA05, SYLM08]. A classic PD controller can later be used to track these desired trajectories. The drawback of this approach is in the lack of robustness since the feasibility constraints are only taken into account during the planning phase, as well as the necessity of replanning in case of changes in the environment or in general when modifications to the desired trajectories are required. Using parameterized trajectories partially overcomes the latter problem. The idea of the method is to express the trajectories in terms of a set of parameters. Changing the parameters allows to adapt the desired trajectory to achieve different step lengths or walking speeds [WC06]. A similar idea is used in the context of hybrid

zero dynamics of planar biped walkers [WGC+07]. In this case, the optimization provides the parameters of the parameterized trajectories to stabilize the hybrid zero dynamics (HZD), once suitable output functions have been chosen. The HZD is an extension of the zero dynamics [Isi95] to a hybrid system. The additional requirement is that the invariant manifold, on which the zero dynamics is defined, must be also invariant under the impact map of the system. If the convergence of the virtual constraint is exponentially fast, then the asymptotic stability of the HZD implies the asymptotic stability of the whole hybrid system.

The most robust way to use the optimization is to use it for both the planning and the on-line control. The limitations of this approach are in the computation cost and the availability of theoretical results for the stability analysis. For example, the nonlinear model predictive control (NMPC) presents stability properties only for a particular class of nonlinear systems and under certain assumptions [ABQ+99, APE04].

Finally, some balancing controllers, like those in [HOR14, HDO16], resort to an optimization problem in order to obtain a solution to the force redistribution problem that arises when more than one end effector is in contact with the environment. In this case, the controller aims at producing desired contact forces, which are obtained by choosing the necessary torques at the joints.

1.2. Control of elastic joint robots

Elastically actuated robots, in contrast to robot with a flexible link structure, are modeled using a spring connecting the motor to the link that it actuates.

Tomei [Tom91] in the early 90s already showed that elastic joint robots can be stabilized to a desired configuration by a simple PD controller for the motor angle, provided also a compensation of the gravity torque at the equilibrium. Since then, many other controllers have been proposed. Loosely speaking, the gravity compensation was the only term fighting the underactuation, so to achieve better results other authors started using additional feedback information.

In [GHS89], an approximate analysis based on the singular perturbation theory justifies the use of torque measurements and link side positions. In [ASH01], the whole state of the robot (motor position, torques and their first derivatives) is used in the feedback loop. The best performance is theoretically given by decoupling based approaches, which provide a partially or even fully linearized closed loop system and ensure global asymptotic stability also for the tracking case. Nevertheless, passivity based approaches [ASOH07, OASKH08] do not require measurements of the contact forces and therefore provide a high degree of robustness for contacts with unknown environments. Static and dynamic feedback linearization were used in [Spo87, De 88, DL98] in order to solve the tracking problem, both for the reduced and complete model of flexible joint robots. These controllers, however, require as a state vector the link side positions up to their third derivative and/or a very accurate robot model. The situation with backstepping based controllers is similar to that of decoupling based approaches. Backstepping is based on the idea of considering the torque produced by the springs [NT93] as an intermediate input to be used to control the link equations. In a second step, the actual torque input to the motors are designed so as to follow the reference behavior for the previous intermediate input. A comparison of the many works on this topic can be found in [BOL95]. This design process is easily transformed into an adaptive version. In order to cope with parameter uncertainty, adaptive control results for robots with elastic joints include the high-gain (approximate) scheme [Spo89] and the global (but very complex) solution obtained in [LB92]. Moreover, robust control schemes based on sliding mode techniques have been proposed in [SRS88]. On the other hand, singular perturbation based controllers are easy to implement, but their performance is theoretically and practically limited to the case of relatively high joint stiffness. A nonlinear controller based on the two-time scale separation property was

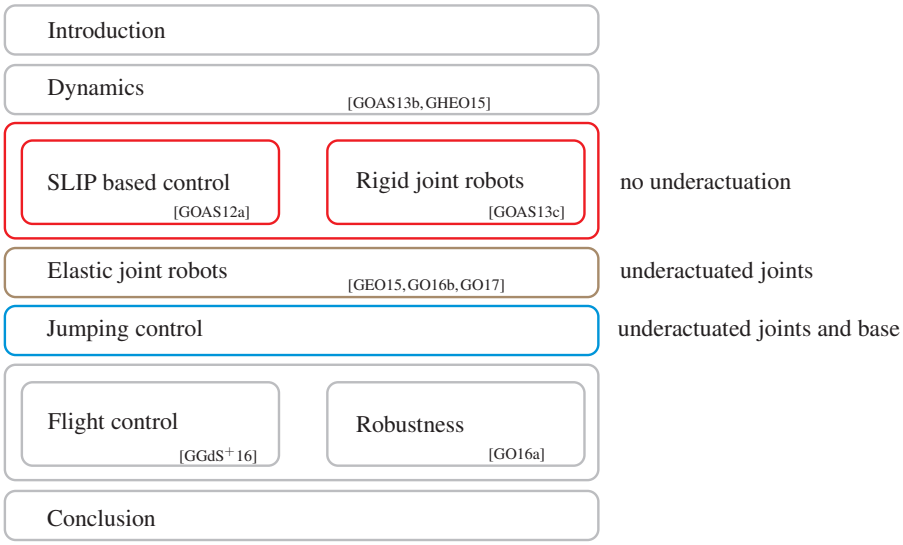

Figure 1.2.: Graphical overview of the thesis and the relation to the publications.

proposed in [KK85].

Finally, the case of Cartesian impedance for elastic joint robots was considered in [Ott08], were singular perturbation, backstepping and passivity approaches are analyzed and compared.

1.3. Overview and organization of the thesis

From what has been said so far it is clear that the analysis and control of legged robots with, additionally, elastically actuated joints are definitely not easy tasks. Even when adding numerous simplifying hypothesis, the problem can be extremely difficult to solve. In this work, a passive behavior of the system with respect to the contact forces is aimed at. The control action will be designed avoiding as much as possible force measurements, which in such a highly dynamic task are likely to be not reliable. At the same time, the periodicity of locomotion patterns is targeted through regulation of an energy function of the system. The motivation behind this choice is the presence of the elastic elements in the joints. Since energy will be stored in the deflected springs, it is clear that taking directly this energy into account to establish a repetitive task of the robot can be one way to exploit the presence of this type of actuation. To the best of the knowledge of the author, this is so far the only approach that clearly shows how to take advantage of such actuation principle. The final result of this work consists in achieving a sequence of jumps with an elastically actuated biped robot. The designed control law does not rely on strong simplifying assumptions, it uses the whole dynamic model of the robot and it is based on the asymptotically stable controller for elastic joints presented in this thesis. Nevertheless, in order to achieve the goal, several intermediate steps are taken. The simplification used in each step will consist in removing one or both the sources of underactuation in the system. The organization of the thesis is sketched in Fig. 1.2, where additionally the papers from Table 1.1 and Table 1.2, that constitute the main references for the chapters, are indicated.

The chapters circled in gray are not directly related to the design steps followed to obtain the controller responsible for producing a jumping sequence with an elastically actuated robot. In chapter 2, the focus is on the dynamic model of the robot and its properties. The information in there give directions for building a **dynamic library**, which is indispensable for testing and implementing the controllers presented in this work. Chapter 7 and chapter 8, instead, provide

solutions to improve the **control of the end effector during the flight phase** and increase the **robustness of the energy control in presence of disturbances**.

In the chapters in red, all the sources of underactuation are removed and a rigidly actuated robot with a fixed base will be considered. Chapter 3 focuses more on the legged robotics side of the problem, providing a solution for **mapping a simple template model on an actual robot**. The goal is to understand how to use the elasticity in locomotion through control strategies inspired by the SLIP template model. In chapter 4 the focus is shifted on the energy regulation. There, the problem of **generating an asymptotically stable limit cycle for a rigidly actuated manipulator** is solved.

When the underactuation due to the elastic joints is considered, new controllers need to be designed. The latter are presented in the chapter 5 (circled in brown) and allow to solve the problem of **orbital stabilization for elastic joint robots**. The control laws are capable of both establishing and damping out controlled oscillations and represents probably the main theoretical contribution of the thesis.

Chapter 6 (circled in blue) takes into account all the forms of underactuation, and a **jumping controller for elastically actuated robots** is presented.

Finally, chapter 9 summarizes the contributions of the thesis and provides directions for future works.

Table 1.1.: List of publications as first author.

Reference	Description
Journal, [GO16a]	Gianluca Garofalo and Christian Ott. Limit cycle control using energy function regulation with friction compensation. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):90–97, 2016
Journal, [GO17]	Gianluca Garofalo and Christian Ott. Energy based limit cycle control of elastically actuated robots. *IEEE Trans. on Automatic Control*, page to appear, 2017
Conference, [GOAS12a]	Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Walking control of fully actuated robots based on the bipedal SLIP model. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1999–2004, Saint Paul, USA, May 2012
Conference, [GOAS13c]	Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Orbital stabilization of mechanical systems through semidefinite Lyapunov functions. In *American Control Conference (ACC)*, pages 5735–5741, Washington DC, USA, June 2013
Conference, [GOAS13b]	Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. On the closed form computation of the dynamic matrices and their differentiations. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2364–2359, Tokyo, Japan, November 2013
Conference, [GHEO15]	Gianluca Garofalo, Bernd Henze, Johannes Englsberger, and Christian Ott. On the inertially decoupled structure of the floating base robot dynamics. In *8th Vienna International Conference on Mathematical Modelling (2015)*, pages 322–327, Vienna, Austria, February 2015
Conference, [GEO15]	Gianluca Garofalo, Johannes Englsberger, and Christian Ott. On the regulation of the energy of elastic joint robots: excitation and damping of oscillations. In *American Control Conference (ACC)*, pages 4825–4831, Chicago, USA, July 2015
Conference, [GO16b]	Gianluca Garofalo and Christian Ott. Steps towards energy efficiency in elastically actuated robots. In Mohammad O. Tokhi and Gurvinder S. Virk, editors, *Advances in Cooperative Robotics: Proceedings of the 19th International Conference on CLAWAR 2016*, pages 780–782, London, UK, September 2016. World Scientific
Poster, [GOAS12b]	Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Walking control of fully actuated robots based on the bipedal SLIP model. In *Proceedings of the Conference on Dynamic Walking*, Pensacola, USA, May 2012
Poster, [GOAS13a]	Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Asymptotically stable limit cycles generation by using nullspace decomposition and energy regulation. In *DGR-Tage*, Munich, Germany, October 2013

Table 1.2.: List of publications as coauthor.

Reference	Description
Journal, [PŽGO14]	Tadej Petrič, Leon Žlajpah, Gianluca Garofalo, and Christian Ott. Walking with adaptive oscillator and dynamic movement primitives. *International Journal of Mechanics and Control*, 15(1):3–10, 2014
Conference, [PŽGO13]	Tadej Petrič, Leon Žlajpah, Gianluca Garofalo, and Christian Ott. Walking control using adaptive oscillators combined with dynamic movement primitives. In *22nd International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, pages 204–211, Portorož, Slovenia, September 2013
Conference, [LGP+13]	Dominic Lakatos, Gianluca Garofalo, Florian Petit, Christian Ott, and Alin Albu-Schäffer. Modal limit cycle control for variable stiffness actuated robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4934–4941, Karlsruhe, Germany, May 2013
Conference, [LGDAS14]	Dominic Lakatos, Gianluca Garofalo, Alexander Dietrich, and Alin Albu-Schäffer. Jumping control for compliantly actuated multi-legged robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4562–4568, Hong Kong, China, May 2014
Conference, [EWO+14]	Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, and Alin Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In *IEEE/RAS Int. Conf. on Humanoid Robots*, pages 916–923, Madrid, Spain, November 2014
Conference, [HWR+14]	Bernd Henze, Alexander Werner, Maximo Alejandro Roa, Gianluca Garofalo, Johannes Englsberger, and Christian Ott. Control applications of toro - a torque controlled humanoid robot. In *IEEE/RAS Int. Conf. on Humanoid Robots*, Madrid, Spain, November 2014
Conference, [GGdS+16]	Alessandro Massimo Giordano, Gianluca Garofalo, Marco de Stefano, Christian Ott, and Alin Albu-Schäffer. Dynamics and control of a free-floating space robot in presence of nonzero linear and angular momenta. In *IEEE Conf. on Decision and Control*, page to appear, 2016

CHAPTER 2

Dynamics

As a result of the steady increasing complexity of robotic systems, researchers have constantly analyzed the dynamic equations of motion of the robot to improve the computational efficiency and gain insights that might lead to more effective control laws.

Some of the milestones are the results provided in [Uic65, SV76] where the recursive Newton-Euler algorithm was formulated and in [OMVH79] where a more efficient version was presented. In [Hol80] it was shown that not only the Newton-Euler approach, but also the Lagrangian formulation could provide an equally efficient algorithm and in [Sil82] the equivalence of the two methods was provided. While the previous works deal with the inverse dynamics problem, in [Ver74] the direct one is considered, so that in the end for both problems algorithms with $O(n)$ complexity are available. Other authors investigated the possibility of using more elegant and efficient tools to write the dynamics. Important are the works of Featherstone [Fea87], Rodriguez [Rod87] and Park et al. [PBP95], where spatial operator algebra and Lie groups are respectively used.

The remainder of this chapter strongly relies on screw theory, therefore in section 2.1 some of the basic concepts are recalled. An influential textbook on this topic is [MLS94], to which the reader is referred for the details. However, the goal of the section is to make the reader familiar with the notation used and moreover to look at the problem from a point of view typical of control theory. As insights about the structure of the equations have turned out to be of great usefulness when dealing with floating base robots, section 2.2.2 collects some classic properties and others introduced in this work. On the other hand, flexible joint robots have risen the need of computing the derivatives with respect to time of all the matrices of the dynamic model, as it is shown in [Spo87, DF11, Ott08]. The solution of other problems and analysis requires, instead, to differentiate the matrices with respect to the state or the dynamic parameters. The controllability analysis of underactuated manipulators motivated in [Mül07] to provide an efficient factorization for the inverse of the inertia matrix, in order to compute its partial derivatives. The derivative with respect to the state of the direct and inverse dynamic function, useful in optimization problems or in state estimation problems, were provided in [SYLM08, SB01]. For these reasons, section 2.3 contributes to the differentiation with respect to time, state and dynamic parameters of the matrices of the dynamic model.

For the reader not familiar with twists and wrenches, the expression of the matrices used in the chapter are provided. To this end, let $\boldsymbol{E}_3 \in \mathbb{R}^{3\times3}$ be the identity matrix, $\boldsymbol{O}_{3\times3} \in \mathbb{R}^{3\times3}$ and $\mathbf{0} \in \mathbb{R}^3$ have all the entries equal to zero, while $\boldsymbol{I}_k$ is the inertia tensor with respect to a frame oriented as

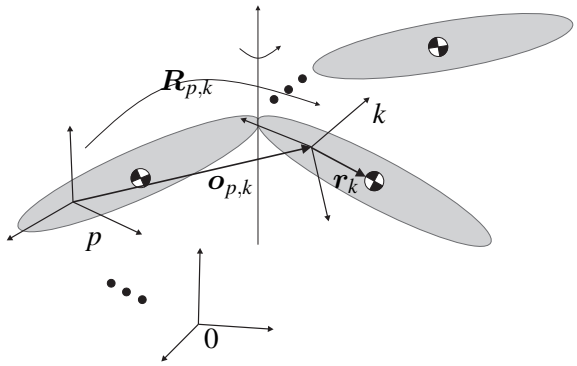

Figure 2.1.: Given the frame p and the frame k, $o_{p,k}$ is the vector connecting their origins and $R_{p,k}$ the rotation matrix to go from p to k.

the body frame and with the origin in the CoM of the rigid body (indicated with r_k). Finally, $\check{a}$ is the skew symmetric matrix such that $\check{a}b = a \times b$. Then, the homogeneous transformation matrix, adjoint and Lie bracket matrices for a given rotation matrix[1] $R_{p,k} \in SO(3)$ and position vector $o_{p,k} \in \mathbb{R}^3$ as in Fig. 2.1 are:

$$T_{p,k} = \begin{bmatrix} R_{p,k} & o_{p,k} \\ 0^T & 1 \end{bmatrix}, \ \mathrm{Ad}_{p,k} = \begin{bmatrix} R_{p,k} & \check{o}_{p,k} R_{p,k} \\ O_{3\times 3} & R_{p,k} \end{bmatrix}, \ \mathrm{adj}_{p,k} = \begin{bmatrix} \check{\omega}_k & \check{v}_k \\ O_{3\times 3} & \check{\omega}_k \end{bmatrix},$$

where the convention for the twist coordinates uses the linear part for the top three rows and the angular part for the bottom ones, i.e. $\nu_k = \begin{bmatrix} v_k^T & \omega_k^T \end{bmatrix}^T$ [MLS94]. All the other matrices are defined accordingly, so the inertia matrix is:

$$\Lambda_k = \begin{bmatrix} m_k E_3 & -m_k \check{r}_k \\ m_k \check{r}_k & I_k - m_k \check{r}_k^2 \end{bmatrix}.$$

Nevertheless, throughout the chapter the explicit expression of the matrices is avoided, so the reader used to the opposite convention will have no problem to follow the equations.

2.1. Screw theory in a nutshell

The goal of this section is to show that a screw motion of a rigid body can be seen as the flow of a special dynamical system. This gives an interpretation of a screw in terms of system analysis theory. To this end, it is useful to recall that given the non autonomous linear system

$$\dot{\chi}(t) = A(t)\chi(t), \tag{2.1}$$

the integral solution can be easily computed when $A(t) = A_0\theta(t)$, where A_0 is a constant matrix and $\theta(t)$ is a scalar function of time. In this case it is possible to write

$$\chi(t) = e^{A_0 \int_0^t \theta(\tau)\mathrm{d}\tau}\chi(0), \tag{2.2}$$

where it is assumed that the initial time is $t_0 = 0$.

[1] Since a rotation matrix has orthonormal columns and determinant equal to 1, it belongs to the special orthonormal group $SO(3)$ of the real (3×3) matrices.

2.1.1. Screw motion

For a rigid body system the position of a point $\boldsymbol{p}^s$ in the spatial frame, i.e. the fixed frame with respect to which the motion of the body is considered, can be computed from the position of the same point $\boldsymbol{p}^b$ in the body frame, once it is known the homogeneous transformation matrix

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{o} \\ \boldsymbol{0}^T & 1 \end{bmatrix} , \tag{2.3}$$

where $\boldsymbol{R}$ is the rotation matrix from the spatial to the body frame and $\boldsymbol{o}$ is the position vector of the origin of the body frame in the spatial frame. When the body frame is moving in time, then

$$\boldsymbol{p}^s(t) = \boldsymbol{T}(t)\boldsymbol{p}^b , \tag{2.4}$$

which differentiated with respect to time leads to

$$\dot{\boldsymbol{p}}^s(t) = \dot{\boldsymbol{T}}(t)\boldsymbol{p}^b = \dot{\boldsymbol{T}}(t)\boldsymbol{T}^{-1}(t)\boldsymbol{p}^s(t) . \tag{2.5}$$

This has exactly the form of the non autonomous linear system in (2.1) with dynamic matrix given by

$$\dot{\boldsymbol{T}}(t)\boldsymbol{T}^{-1}(t) = \begin{bmatrix} \breve{\boldsymbol{\omega}}(t) & -\breve{\boldsymbol{\omega}}(t)\boldsymbol{o}(t) + \dot{\boldsymbol{o}}(t) \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} , \tag{2.6}$$

where $\breve{\boldsymbol{\omega}}(t) = \dot{\boldsymbol{R}}(t)\boldsymbol{R}^{-1}(t)$ is a skew symmetric matrix. It is therefore of interest to consider the case when $\dot{\boldsymbol{T}}(t)\boldsymbol{T}^{-1}(t) = \boldsymbol{T}_0\dot{\theta}(t)$, since using (2.2) and assuming that the reference is chosen in such a way that $\theta(0) = 0$, the solution is

$$\boldsymbol{p}^s(t) = e^{\boldsymbol{T}_0\theta(t)}\boldsymbol{p}^s(0) . \tag{2.7}$$

Introducing the constants $\boldsymbol{\omega}_0$ and $\boldsymbol{d}$, from (2.6) it follows that the goal is to satisfy

$$\breve{\boldsymbol{\omega}}(t) = \breve{\boldsymbol{\omega}}_0\dot{\theta}(t) \tag{2.8}$$

$$-\breve{\boldsymbol{\omega}}(t)\boldsymbol{o}(t) + \dot{\boldsymbol{o}}(t) = \boldsymbol{d}\dot{\theta}(t) . \tag{2.9}$$

Using the identity $-\breve{\boldsymbol{\omega}}_0^2 = \left(\boldsymbol{E}_3 - \boldsymbol{\omega}_0\boldsymbol{\omega}_0^T\right)$, the decomposition of $\boldsymbol{d}$ along the direction of $\boldsymbol{\omega}_0$ and an orthogonal plane can be computed as

$$\boldsymbol{d} = -\breve{\boldsymbol{\omega}}_0^2\boldsymbol{d} + \boldsymbol{\omega}_0\boldsymbol{\omega}_0^T\boldsymbol{d} = -\breve{\boldsymbol{\omega}}_0\boldsymbol{c} + h\boldsymbol{\omega}_0 , \tag{2.10}$$

being $\breve{\boldsymbol{\omega}}_0\boldsymbol{d} = \boldsymbol{c}$ and $\boldsymbol{\omega}_0^T\boldsymbol{d} = h$ also constants. The final result is

$$\dot{\boldsymbol{T}}(t)\boldsymbol{T}^{-1}(t) = \begin{bmatrix} \breve{\boldsymbol{\omega}}_0 & -\breve{\boldsymbol{\omega}}_0\boldsymbol{c} + h\boldsymbol{\omega}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \dot{\theta}(t) , \tag{2.11}$$

where the well known expression of the constant twist associated to a screw motion with pitch h, direction of the axis given by $\boldsymbol{\omega}_0$ and passing through the point $\boldsymbol{c}$ can be recognized.

2.1.2. The Newton-Euler equation in body coordinates

Consider a frame attached to a rigid body (which will be indicated with the index k) in movement with respect to a spatial frame (which will be indicated with the index 0). Using the body twist $\boldsymbol{\nu}_k$ and the body wrench $\boldsymbol{\mu}_k$, the equation of motion can be written as

$$\frac{d}{dt}\left(\mathrm{Ad}_{0,k}^{-T}\,\boldsymbol{\Lambda}_k\boldsymbol{\nu}_k\right) = \mathrm{Ad}_{0,k}^{-T}\,\boldsymbol{\mu}_k\ , \tag{2.12}$$

where $\boldsymbol{\Lambda}_k$ is the constant body inertia matrix and $\mathrm{Ad}_{0,k}$ is the adjoint matrix which uses an element of the Lie group (the homogeneous transformation matrix from the inertial to the body frame) as a linear mapping on the Lie algebra [MLS94]. Computing the time derivative in (2.12), leads to

$$\boldsymbol{\Lambda}_k\dot{\boldsymbol{\nu}}_k - \mathrm{adj}_{0,k}^{T}\,\boldsymbol{\Lambda}_k\boldsymbol{\nu}_k = \boldsymbol{\mu}_k\ , \tag{2.13}$$

where $\frac{d}{dt}\left(\mathrm{Ad}_{0,k}^{-1}\right) = -\,\mathrm{adj}_{0,k}\,\mathrm{Ad}_{0,k}^{-1}$, being $\mathrm{adj}_{0,k}$ the Lie bracket matrix which uses an element of the Lie algebra (the body twist $\boldsymbol{\nu}_k$) as a linear mapping on the Lie algebra itself [MLS94]. Finally, using the property $\mathrm{adj}_{0,k}\,\boldsymbol{\nu}_k = \mathbf{0}$, the term $\boldsymbol{\Lambda}_k\,\mathrm{adj}_{0,k}\,\boldsymbol{\nu}_k$ can added to (2.13) without changing the equation, so that it can be rewritten as

$$\boldsymbol{\Lambda}_k\dot{\boldsymbol{\nu}}_k + \boldsymbol{\Gamma}_k\boldsymbol{\nu}_k = \boldsymbol{\mu}_k\ , \tag{2.14}$$

where $\boldsymbol{\Gamma}_k = \left(\boldsymbol{\Lambda}_k\,\mathrm{adj}_{0,k} - \mathrm{adj}_{0,k}^{T}\,\boldsymbol{\Lambda}_k\right)$. Thanks to this choice, $\boldsymbol{\Gamma}_k$ is skew symmetric ($\boldsymbol{\Gamma}_k = -\boldsymbol{\Gamma}_k^T$) and being $\boldsymbol{\Lambda}_k$ constant, the property $\dot{\boldsymbol{\Lambda}}_k = \boldsymbol{\Gamma}_k + \boldsymbol{\Gamma}_k^T$, known also as passivity property in robotics [SSVO08, Ott08], is satisfied.

2.2. Dynamic model

Floating base robots that are a tree structure of N articulated bodies are considered. The N equations of motions in the form of (2.14), one for each of the N bodies, can be written stacking the twists in $\bar{\boldsymbol{\nu}} = \mathrm{col}\,(\boldsymbol{\nu}_k)$, the wrenches in $\bar{\boldsymbol{\mu}} = \mathrm{col}\,(\boldsymbol{\mu}_k)$ and using the block diagonal matrices $\bar{\boldsymbol{\Lambda}} = \mathrm{blkdiag}\,(\boldsymbol{\Lambda}_k)$ and $\bar{\boldsymbol{\Gamma}} = \mathrm{blkdiag}\,(\boldsymbol{\Gamma}_k)$, where $k = 1,...,N$. Since the bodies are constrained to each other, it exists a mapping in the form $\boldsymbol{\nu}_k = \boldsymbol{J}_k(\boldsymbol{x})\boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^{n_v}$ are the complete velocity coordinates while $\boldsymbol{x}$ indicates the complete configuration coordinates of the robot. The latter is given by the configuration of the base $\boldsymbol{x}_1$ and the joint position $\boldsymbol{q} \in \mathbb{R}^{n_q}$ and together with $\boldsymbol{v}$ constitute the state of the robot. Note that in general $\boldsymbol{x}_1$ might not have necessarily all entries in $\mathbb{R}$ and

$$\dot{\boldsymbol{x}} \neq \boldsymbol{v}\ ,$$

which is the case when, for example, the rotation matrix is used to represent the orientation of the base. Another consequence of the constraints between the N bodies is the possibility to project each of the equations in the form of (2.14) in the space orthogonal to the constraint reaction wrenches. This is possible through $\boldsymbol{J}_k^T(\boldsymbol{x})$, so that

$$\bar{\boldsymbol{J}}^T(\boldsymbol{x})\left[\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{J}}(\boldsymbol{x})\dot{\boldsymbol{v}} + \left(\bar{\boldsymbol{\Gamma}}\bar{\boldsymbol{J}}(\boldsymbol{x}) + \bar{\boldsymbol{\Lambda}}\dot{\bar{\boldsymbol{J}}}(\boldsymbol{x})\right)\boldsymbol{v}\right] = \boldsymbol{u} - \bar{\boldsymbol{J}}^T(\boldsymbol{x})\bar{\boldsymbol{\Lambda}}\bar{\boldsymbol{\gamma}}(\boldsymbol{x})\ , \tag{2.15}$$

where $\bar{\boldsymbol{J}} = \mathrm{col}(\boldsymbol{J}_k)$ and each $\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k_c} - \boldsymbol{\Lambda}_k\boldsymbol{\gamma}_k + \boldsymbol{w}_k$ has been split in

- constraint reaction wrench (which disappears after the projection, i.e $\boldsymbol{J}_k^T\boldsymbol{\mu}_{k_c} = \mathbf{0}$),

- the weight (which is linear in $\bar{\gamma} = \mathrm{col}(\gamma_k)$, being γ_k the body gravitational acceleration),
- all the other external wrench (which after the projection correspond to $\boldsymbol{u}$, i.e $\boldsymbol{J}_k^T \boldsymbol{w}_k = \boldsymbol{u}$).

The terms γ_k can be expressed as the product of a configuration dependent part and a constant part, i.e. $\gamma_k = \mathrm{Ad}_{0,k}^{-1}(\boldsymbol{x})\gamma_0$, where γ_0 is the constant spatial gravitational acceleration[2]. The dynamic model of the robot can be rewritten in a more compact way as

$$\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{Q}^T\boldsymbol{\tau} + \boldsymbol{J}_1^T(\boldsymbol{x})\boldsymbol{w}_1 \ , \tag{2.16}$$

where

$$\boldsymbol{M} = \sum_k \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \boldsymbol{J}_k \ , \tag{2.17a}$$

$$\boldsymbol{C} = \sum_k \boldsymbol{J}_k^T \left[\left(\boldsymbol{\Lambda}_k \, \mathrm{adj}_{0,k} - \mathrm{adj}_{0,k}^T \, \boldsymbol{\Lambda}_k \right) \boldsymbol{J}_k + \boldsymbol{\Lambda}_k \dot{\boldsymbol{J}}_k \right] \ , \tag{2.17b}$$

$$\boldsymbol{g} = \sum_k \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \, \mathrm{Ad}_{0,k}^{-1} \gamma_0 \ , \tag{2.17c}$$

$k = 1,...,N$. In (2.16) $\boldsymbol{M} \in \mathbb{R}^{n_v \times n_v}$ is the symmetric and positive definite inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{n_v \times n_v}$ is the Coriolis matrix and $\boldsymbol{g} \in \mathbb{R}^{n_v}$ is the vector of gravity torques. The input to the system $\boldsymbol{u}$ is given by the wrench acting to the base $\boldsymbol{w}_1 \in \mathbb{R}^6$ (mapped through $\boldsymbol{J}_1^T$) and the torques provided by the motors $\boldsymbol{\tau} \in \mathbb{R}^{n_q}$ (mapped through $\boldsymbol{Q}^T$). The matrix $\boldsymbol{Q} \in \mathbb{R}^{n_q \times n_v}$ selects the joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$ out of all the velocity coordinates, i.e. $\dot{\boldsymbol{q}} = \boldsymbol{Q}\boldsymbol{v}$. In this form, the matrices $\boldsymbol{J}_k$, $\mathrm{Ad}_{0,k}^{-1}$ and $\mathrm{adj}_{0,k}$ are state dependent and are the only quantities needed to compute (2.17). Algorithm 1 is one way to obtain such quantities, propagating the matrices from the root to the leaves of the tree structured robot. It is an iterative procedure equivalent to the outward recursion of the Newton-Euler algorithm and, like the latter, it is based on the coordinate transformation for the velocities, which can be formulated as

$$\boldsymbol{\nu}_k = \mathrm{Ad}_{p,k}^{-1} \boldsymbol{\nu}_p + \boldsymbol{\xi}_k \ , \tag{2.18}$$

where k and p are used to indicate a frame attached to the current link and its parent[3], while $\boldsymbol{\xi}_k$ gives the relative velocity between these two bodies. Writing (2.18) in terms of Jacobian matrices, leads to an equality that, being true $\forall \boldsymbol{v}$, allows to write

$$\boldsymbol{J}_k = \mathrm{Ad}_{p,k}^{-1} \boldsymbol{J}_p + \boldsymbol{\Xi}_k \ . \tag{2.19}$$

where $\boldsymbol{\Xi}_k$ is a matrix determined by the type of interconnection between the bodies[4], such that $\boldsymbol{\xi}_k = \boldsymbol{\Xi}_k \boldsymbol{v}$. This concept, as discussed in [RS87], allows the joints to be completely general, with any number of degrees of freedom up to and including six, which is particularly important to model floating base robots. For a 1 - DoF joint the expression of $\boldsymbol{\Xi}_k$ is constant and particularly simple as $\boldsymbol{\xi}_k$ reduces to (2.11), where instead of $\dot{\theta}$ the correspondent joint velocity $\dot{q}_k$ must be replaced. As a consequence, the differentiation of (2.19) with respect to time, leads to

$$\dot{\boldsymbol{J}}_k = \mathrm{Ad}_{p,k}^{-1} \dot{\boldsymbol{J}}_p - \mathrm{adj}_{p,k} \, \mathrm{Ad}_{p,k}^{-1} \boldsymbol{J}_p \ , \tag{2.20}$$

which can be easily included in Algorithm 1.

[2] The angular part of γ_0 is always zero, while assuming for example that the third axis is the vertical one, then the linear part will be $\begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$, where g is the gravitational acceleration constant.

[3] If the bodies are numbered according to a so called regular numbering scheme [Fea87], it is guaranteed that during the iteration the parent's quantities are computed befere those of its children.

[4] Notice how in (2.19) only "local" informations are needed, meaning that $\mathrm{Ad}_{p,k}^{-1}$ and $\boldsymbol{\Xi}_k$ are given by the connection between the link and its parent.

Algorithm 1 Iterative computation of the kinematics

Require:
LABELING
$\boldsymbol{q}$, $\boldsymbol{v}$, $\boldsymbol{\Xi}_k$ for $k = 1, ..., N$.
$\boldsymbol{J}_0, \dot{\boldsymbol{J}}_0 \leftarrow \boldsymbol{O}$
$\mathrm{Ad}_{0,0}^{-1} \leftarrow \boldsymbol{E}$
for $i = 1$ to N **do**
$\boldsymbol{T}_{p,k} = \mathrm{HTM}(\boldsymbol{q})$
$\mathrm{Ad}_{p,k}^{-1} = \mathrm{INVERSEOFADJOINT}(\boldsymbol{T}_{p,k})$
$\mathrm{adj}_{p,k} = \mathrm{LIEBRAKETMATRIX}(\boldsymbol{\xi}_k)$
$\boldsymbol{J}_k = \mathrm{Ad}_{p,k}^{-1} \boldsymbol{J}_p + \boldsymbol{\Xi}_k$ (and eventually its derivatives)
$\mathrm{Ad}_{0,k}^{-1} = \mathrm{Ad}_{p,k}^{-1} \mathrm{Ad}_{0,p}^{-1}$
$\mathrm{adj}_{0,k} = \mathrm{adj}_{p,k} + \mathrm{Ad}_{p,k}^{-1} \mathrm{adj}_{0,p} \mathrm{Ad}_{p,k}$
end for

2.2.1. The reduced model for elastic joint robots

For the derivation of the model, the following assumptions are made [Spo87]:

Assumption 2.1. *The rotors of the motors are rotational symmetric rigid bodies and the rotation axis and the symmetry axis of each joint coincide.*

Assumption 2.2. *The electric part of the dynamics (i.e. the inner current control loop of the motors) is sufficiently fast such that it can be neglected.*

Assumption 2.3. *The rotational part of the kinetic energy of the rotors is determined only by the relative movement of the rotors with respect to the previous link.*

When these assumptions are satisfied, underactuated robots with elastic joints can be modeled as

$$\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{Q}^T \boldsymbol{\tau}(\boldsymbol{\phi}) + \boldsymbol{J}_1^T(\boldsymbol{x})\boldsymbol{w}_1 \tag{2.21a}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau}(\boldsymbol{\phi}) = \boldsymbol{\tau}_m \,, \tag{2.21b}$$

where $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \in \mathbb{R}^{n_q}$ are also part of the state of the system being $\boldsymbol{\theta}$ the motor position as compared to the link position $\boldsymbol{q}$. The deflections of the springs which connect the motors to the links are $\boldsymbol{\phi} = \boldsymbol{\theta} - \boldsymbol{q}$ and $\boldsymbol{\tau}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} U_k(\boldsymbol{\phi}) \in \mathbb{R}^{n_q}$ is the torque[5] produced by the springs with $U_k(\boldsymbol{\phi})$ the correspondent potential function. The gravity torque vector can also be expressed using a gravitational potential $U_g(\boldsymbol{x})$ as $\boldsymbol{g}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} U_g(\boldsymbol{x})$. Finally, $\boldsymbol{\tau}_m \in \mathbb{R}^{n_q}$ is the input to the system provided by the motors.

In case the robot contains not only rotational joints, the subset $\mathbb{X}$ in which all the prismatic joints are kept bounded will always be considered. In this case, it is well known that the following proposition holds [CSB96]

Proposition 2.1. *For every matrix norm there exists a $\beta > 0$ such that*

$$\left\| \frac{\partial^2 U_g(\boldsymbol{x})}{\partial \boldsymbol{x}^2} \right\| < \beta, \ \forall \boldsymbol{x} \in \mathbb{X} \,. \tag{2.22}$$

[5]The symbol $\nabla_{\boldsymbol{x}}(\cdot)$ is used to replace $\left(\frac{\partial(\cdot)}{\partial \boldsymbol{x}}\right)^T$, in order to ease the notation.

For the design of the control law, an additional assumption is required:

Assumption 2.4. *The potential function $U_k(\phi)$ satisfies the following three conditions*

- $U_k(\phi)$ *is a nonlinear scalar function* (A1)
- $\alpha_1 \|\boldsymbol{\xi}\|^2 < \boldsymbol{\xi}^T \dfrac{\partial^2 U_k(\phi)}{\partial \phi^2} \boldsymbol{\xi} < \alpha_2 \|\boldsymbol{\xi}\|^2 , \ \forall \boldsymbol{\xi} \in \mathbb{Q}$ (A2)
- $\beta < \alpha_1$ (A3)

where, since β must be positive, clearly also $\alpha_1, \alpha_2 > 0$. Assumption (A3) in particular, as explained in [Ott08], states that the robot should be designed properly. Namely, the binding forces should grow faster than the diverging forces between $\boldsymbol{\theta}$ and $\boldsymbol{q}$, meaning that the joint stiffness should be sufficiently high to prevent the robot from falling down under the load of its own weight.

2.2.2. Properties of the dynamic model

As mentioned at the beginning of the chapter, the properties of the dynamic models can simplify the design of control algorithms. In this section, the properties that will be used in the remainder of the thesis are presented.

Passivity

Since in section 2.1.2 it was possible to add a term to (2.13) without changing the equation, it is clear that the choice of the matrix $\boldsymbol{C}$ (whose derivation follows from that) is not unique. The term $\boldsymbol{C}\boldsymbol{v}$ represents the centrifugal and Corioilis effects, which are quadratic velocity terms that can be factorized in infinitely many ways. Between this multitude of choices, the one used satisfies once again the passivity property. In fact, since each $\boldsymbol{J}_k^T \left(\boldsymbol{\Lambda}_k \operatorname{adj}_{0,k} - \operatorname{adj}_{0,k}^T \boldsymbol{\Lambda}_k \right) \boldsymbol{J}_k$ is a skew symmetric matrix, then

$$\boldsymbol{C} + \boldsymbol{C}^T = \sum_k \dot{\boldsymbol{J}}_k^T \boldsymbol{\Lambda}_k \boldsymbol{J}_k + \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \dot{\boldsymbol{J}}_k = \dot{\boldsymbol{M}} ,$$

$k = 1, ..., N$, which is equivalent to say that $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew-symmetric. As a consequence, given any $\boldsymbol{a} \in \mathbb{R}^{n_q}$ then $\boldsymbol{a}^T \left(\dot{\boldsymbol{M}} - 2\boldsymbol{C} \right) \boldsymbol{a} = \boldsymbol{0}$. The special case $\boldsymbol{a} = \boldsymbol{v}$ is satisfied by any choice of $\boldsymbol{C}$ since it is a result of the principle of conservation of energy [SSVO08]. The passivity property is a stronger property than $\boldsymbol{v}^T \left(\dot{\boldsymbol{M}} - 2\boldsymbol{C} \right) \boldsymbol{v} = \boldsymbol{0}$, since the matrices themselves depend on $\boldsymbol{v}$. Even requiring to satisfy the passivity property there still exist infinite many $\boldsymbol{C}$. Another way to obtain such a matrix is to use, for example, the Christoffel symbols of the first type, or the factorization used in [Nie90].

Generalized momentum and conservation laws

As it was firstly shown in [KKK+03], it is always possible to express the generalized momentum $\boldsymbol{h} \in \mathbb{R}^6$, i.e. the linear and the angular momentum stacked together, as

$$\boldsymbol{h} = \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{v} , \tag{2.23}$$

where $\boldsymbol{A}(\boldsymbol{x})$ is called the centroidal momentum matrix in [OGL13] if the generalized momentum is expressed in a frame attached to the center of mass (CoM) and with the axis aligned with the inertial frame. In the following, the momentum will be always expressed in this frame.

Since the linear momentum $\boldsymbol{p} \in \mathbb{R}^3$ is directly related to the velocity of the center of mass (CoM) $\dot{\boldsymbol{x}}_{CoM}$, i.e. $\boldsymbol{p} = m\dot{\boldsymbol{x}}_{CoM}$ where m is the total mass of the robot, it is clear that a relationship must exist between the centroidal momentum matrix $\boldsymbol{A}(\boldsymbol{x})$ and the Jacobian matrix $\boldsymbol{J}_{CoM}(\boldsymbol{x})$ mapping the velocity coordinates into the velocity of the CoM of the robot, i.e. $\dot{\boldsymbol{x}}_{CoM} = \boldsymbol{J}_{CoM}(\boldsymbol{x})\boldsymbol{v}$. To this end, the centroidal momentum matrix is partitioned as $\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{A}_p^T(\boldsymbol{x}) & \boldsymbol{A}_l^T(\boldsymbol{x})\end{bmatrix}^T$, so that

$$\boldsymbol{p} = \boldsymbol{A}_p(\boldsymbol{x})\boldsymbol{v} \tag{2.24}$$

$$\boldsymbol{l} = \boldsymbol{A}_l(\boldsymbol{x})\boldsymbol{v}\ , \tag{2.25}$$

where $\boldsymbol{l} \in \mathbb{R}^3$ is the angular momentum. Equating the two expressions of $\boldsymbol{p}$ it follows that

$$\boldsymbol{A}_p(\boldsymbol{x}) = m\boldsymbol{J}_{CoM}(\boldsymbol{x})\ , \tag{2.26}$$

since the equality must be satisfied for every possible choice of $\boldsymbol{v}$. In view of (2.26) and knowing that $\boldsymbol{g}(\boldsymbol{x})$ is obtained from the mapping of the force $\boldsymbol{f}_g = m\,g\,\boldsymbol{e}_g$ through $\boldsymbol{J}_{CoM}^T(\boldsymbol{x})$, the gravity torques can be written as $\boldsymbol{g}(\boldsymbol{x}) = g\boldsymbol{A}_p^T(\boldsymbol{x})\boldsymbol{e}_g$, where g is the gravitational constant and $\boldsymbol{e}_g$ a unit vector pointing upwards.

For a system described by the dynamic model (2.16), it is well known that the following property holds

Proposition 2.2 (Newton−Euler equations)**.** *Let $\boldsymbol{\mu}_{CoM}$ be the total wrench acting at the CoM obtained from the combination of all the external wrenches, then*

$$\dot{\boldsymbol{h}} = \boldsymbol{\mu}_{CoM}\ . \tag{2.27}$$

From the previous property and the expression of the generalized momentum given in (7.2), the following corollaries can be derived

Proposition 2.3 (Conservation of generalized momentum)**.** *Assume that the robot is free floating ($\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$) and no external wrenches are acting on it, then*

$$\boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{v}} + \dot{\boldsymbol{A}}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} = \boldsymbol{0}\ . \tag{2.28}$$

Proposition 2.4 (Conservation of angular momentum)**.** *Assume that the resulting total wrench at the CoM is a pure force, then*

$$\boldsymbol{A}_l(\boldsymbol{x})\dot{\boldsymbol{v}} + \dot{\boldsymbol{A}}_l(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} = \boldsymbol{0}\ . \tag{2.29}$$

Block diagonal structure

A consequences of these basic mechanical principles is that there exists a coordinate transformation that can inertially decouple the equations of motion of a floating base robot, i.e. it transforms the inertia matrix into a block diagonal matrix. The transformation is based on orthogonality relationships between matrices. This result includes and extends those in [HHC07, ORH11], allowing to separate the linear and angular centroidal dynamics from the joint dynamics. To easily derive the results, special cases are considered, although the results are valid in general.

Consider a free floating rigidly actuated robot with no external wrenches acting on it, so that the only input are the torques provided by the motors. Replacing $\dot{\boldsymbol{v}}$ from (2.16) into (2.28) and setting $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{w}_1$ to zero leads to

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{Q}^T\boldsymbol{\tau} - \left(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v}) - \dot{\boldsymbol{A}}(\boldsymbol{x},\boldsymbol{v})\right)\boldsymbol{v} = \boldsymbol{0}\ . \tag{2.30}$$

Equation (2.30) holds for every possible choice of $\boldsymbol{v}$ and $\boldsymbol{\tau}$, therefore it is clear that the following conditions must be satisfied

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{Q}^T = \boldsymbol{0} \tag{2.31}$$

$$\left(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v}) - \dot{\boldsymbol{A}}(\boldsymbol{x},\boldsymbol{v})\right)\boldsymbol{v} = \boldsymbol{0}\,. \tag{2.32}$$

Consider the velocity coordinate transformation, inspired by the one used in [OKN08],

$$\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{h}\\ \dot{\boldsymbol{q}}\end{bmatrix} = \boldsymbol{A}_Q(\boldsymbol{x})\boldsymbol{v} \qquad\qquad \boldsymbol{A}_Q(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{A}(\boldsymbol{x})\\ \boldsymbol{Q}\end{bmatrix}\,, \tag{2.33}$$

with the inverse transformation given by[6]

$$\boldsymbol{v} = \boldsymbol{A}_Q^{-1}(\boldsymbol{x})\boldsymbol{\xi} \qquad\qquad \boldsymbol{A}_Q^{-1}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{A}^{+_M}(\boldsymbol{x}) & \boldsymbol{Q}^{+_M}(\boldsymbol{x})\end{bmatrix}\,. \tag{2.34}$$

Pre-multiplying (2.16) by $\boldsymbol{A}_Q^{-T}(\boldsymbol{x})$, using

$$\dot{\boldsymbol{v}} = \boldsymbol{A}_Q^{-1}(\boldsymbol{x})\left(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{A}}_Q(\boldsymbol{x})\boldsymbol{A}_Q^{-1}(\boldsymbol{x})\boldsymbol{\xi}\right) \tag{2.35}$$

and the orthogonality relationship expressed in (2.31) leads to the equations of motion in the new coordinates

$$\underbrace{\begin{bmatrix}\boldsymbol{\Lambda}_h(\boldsymbol{x}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Lambda}_q(\boldsymbol{x})\end{bmatrix}}_{\boldsymbol{\Lambda}(\boldsymbol{x})}\dot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix}\boldsymbol{\Gamma}_{hh}(\boldsymbol{x},\boldsymbol{v}) & \boldsymbol{\Gamma}_{hq}(\boldsymbol{x},\boldsymbol{v})\\ -\boldsymbol{\Gamma}_{hq}^T(\boldsymbol{x},\boldsymbol{v}) & \boldsymbol{\Gamma}_{qq}(\boldsymbol{x},\boldsymbol{v})\end{bmatrix}}_{\boldsymbol{\Gamma}(\boldsymbol{x},\boldsymbol{A}_Q^{-1}(\boldsymbol{x})\boldsymbol{\xi})}\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{\tau}\end{bmatrix}\,, \tag{2.36}$$

where

$$\boldsymbol{\Lambda}_h(\boldsymbol{x}) = \left(\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{A}^T(\boldsymbol{x})\right)^{-1} \tag{2.37}$$

$$\boldsymbol{\Lambda}_q(\boldsymbol{x}) = \left(\boldsymbol{Q}\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{Q}^T\right)^{-1} \tag{2.38}$$

$$\boldsymbol{\Gamma}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{\xi} = \boldsymbol{\Lambda}(\boldsymbol{x})\begin{bmatrix}\boldsymbol{A}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v}) - \dot{\boldsymbol{A}}(\boldsymbol{x},\boldsymbol{v})\\ \boldsymbol{Q}\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\end{bmatrix}\boldsymbol{v} \quad = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Q}^{+_M T}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v}\end{bmatrix}\,. \tag{2.39}$$

Equation (2.39) shows that, although the generalized momentum will influence the joint dynamics through $\boldsymbol{\Gamma}(\boldsymbol{x},\boldsymbol{v})$ since in general $-\boldsymbol{\Gamma}_{hq}^T(\boldsymbol{x},\boldsymbol{v})\boldsymbol{h} \neq \boldsymbol{0}$, the inverse is not true. As a matter of fact, from (2.32) and (2.39) follows that

$$\boldsymbol{\Gamma}_{hh}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{h} + \boldsymbol{\Gamma}_{hq}(\boldsymbol{x},\boldsymbol{v})\dot{\boldsymbol{q}} = \boldsymbol{0}\,, \tag{2.40}$$

which can be seen as a natural consequence of the conservation of the generalized momentum.

The structure can be further explored to show additional inertial separation. The condition expressed in (2.31) can be interpreted in such a way that the torques produced by the motors are internal forces for the free floating robot and therefore they produce no change in the generalized momentum. Similarly, assume that the resulting total wrench at the CoM is a pure force $\boldsymbol{f}_{CoM}$, which maps via $\boldsymbol{J}_{CoM}^T(\boldsymbol{x})$. Replacing $\dot{\boldsymbol{v}}$ from (2.16) into (2.29) and setting $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{\tau}$ to zero leads to

$$\frac{1}{m}\boldsymbol{A}_l(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{A}_p^T(\boldsymbol{x})\boldsymbol{f}_{CoM} - \left(\boldsymbol{A}_l(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x},\boldsymbol{v}) - \dot{\boldsymbol{A}}_l(\boldsymbol{x},\boldsymbol{v})\right)\boldsymbol{v} = 0\,, \tag{2.41}$$

[6] Given a matrix $\boldsymbol{T}$, $\boldsymbol{T}^{+_M}$ denotes the dynamically consistent weighted pseudo inverse defined as $\boldsymbol{T}^{+_M}(\boldsymbol{x}) := \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{T}^T\left(\boldsymbol{T}\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{T}^T\right)^{-1}$.

where, additionally, (2.26) has been used. This condition, holding for every possible choice of $\boldsymbol{v}$ and $\boldsymbol{f}_{CoM}$, leads to

$$\boldsymbol{A}_l(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{A}_p^T(\boldsymbol{x}) = \boldsymbol{0}\,. \tag{2.42}$$

In view of this orthogonality relationship, $\boldsymbol{\Lambda}_h(\boldsymbol{x})$ is itself block diagonal. In particular,

$$\boldsymbol{\Lambda}_h(\boldsymbol{x}) = \left(\begin{bmatrix}\boldsymbol{A}_p(\boldsymbol{x})\\ \boldsymbol{A}_l(\boldsymbol{x})\end{bmatrix}\boldsymbol{M}^{-1}(\boldsymbol{x})\begin{bmatrix}\boldsymbol{A}_p^T(\boldsymbol{x}) & \boldsymbol{A}_l^T(\boldsymbol{x})\end{bmatrix}\right)^{-1} = \begin{bmatrix}\boldsymbol{\Lambda}_p(\boldsymbol{x}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Lambda}_l(\boldsymbol{x})\end{bmatrix}\,, \tag{2.43}$$

with

$$\boldsymbol{\Lambda}_p(\boldsymbol{x}) = \left(\boldsymbol{A}_p(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{A}_p^T(\boldsymbol{x})\right)^{-1} \tag{2.44}$$

$$\boldsymbol{\Lambda}_l(\boldsymbol{x}) = \left(\boldsymbol{A}_l(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{A}_l^T(\boldsymbol{x})\right)^{-1}\,. \tag{2.45}$$

This has an impact also on the pseudo-inverse of the centroidal momentum matrix $\boldsymbol{A}^{+_M}(\boldsymbol{x})$ and consequently on the gravity term. The first can be expressed as

$$\boldsymbol{A}^{+_M}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})\begin{bmatrix}\boldsymbol{A}_p^T(\boldsymbol{x}) & \boldsymbol{A}_l^T(\boldsymbol{x})\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_p(\boldsymbol{x}) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Lambda}_l(\boldsymbol{x})\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_p^{+_M}(\boldsymbol{x}) & \boldsymbol{A}_l^{+_M}(\boldsymbol{x})\end{bmatrix}\,, \tag{2.46}$$

where (2.43) - (2.45) have been taken into account, from which immediately follows

$$\boldsymbol{A}_Q^{-T}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{A}_p^{+_M T}(\boldsymbol{x})\\ \boldsymbol{A}_l^{+_M T}(\boldsymbol{x})\\ \boldsymbol{Q}^{+_M T}(\boldsymbol{x})\end{bmatrix} g\boldsymbol{A}_p^T(\boldsymbol{x})\boldsymbol{e}_g = \begin{bmatrix}g\,\boldsymbol{e}_g\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix}\,. \tag{2.47}$$

Combining all the previous results, (2.16) can be written in the new coordinates as

$$\begin{bmatrix}\boldsymbol{\Lambda}_p & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Lambda}_l & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda}_q\end{bmatrix}\dot{\boldsymbol{\xi}} + \begin{bmatrix}\boldsymbol{\Gamma}_{pp} & \boldsymbol{\Gamma}_{pl} & \boldsymbol{\Gamma}_{pq}\\ -\boldsymbol{\Gamma}_{pl}^T & \boldsymbol{\Gamma}_{ll} & \boldsymbol{\Gamma}_{lq}\\ -\boldsymbol{\Gamma}_{pq}^T & -\boldsymbol{\Gamma}_{lq}^T & \boldsymbol{\Gamma}_{qq}\end{bmatrix}\boldsymbol{\xi} + \begin{bmatrix}g\,\boldsymbol{e}_g\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{\tau}\end{bmatrix} + \boldsymbol{A}_Q^{-T}\boldsymbol{J}_1^T\boldsymbol{w}_1\,, \tag{2.48}$$

which is still eligible of one last refinement. Assume, in fact, that no contact wrenches are acting on the system ($\boldsymbol{w}_1 = \boldsymbol{0}$). Taking into account (2.40), the first three lines of the model are

$$\boldsymbol{\Lambda}_p(\boldsymbol{x})\dot{\boldsymbol{p}} = -g\,\boldsymbol{e}_g\,. \tag{2.49}$$

On the other hand, from (2.27) it follows that $\dot{\boldsymbol{p}} = -m\,g\,\boldsymbol{e}_g$ must hold. Comparing the two expressions of $\dot{\boldsymbol{p}}$, it can be concluded that

$$\boldsymbol{\Lambda}_p = \frac{1}{m}\boldsymbol{E}_3\,, \tag{2.50}$$

where $\boldsymbol{E}_3$ is the 3×3 identity matrix. Note that, because of the passivity property, it also follows that $\boldsymbol{\Gamma}_{pp}(\boldsymbol{x},\boldsymbol{v})$ is skew-symmetric.

It is important to notice that the physical units of the first six equations and the remaining ones are not the same. This is not surprising, since a mix of velocities and generalized momentum is used as part of the new state.

2.3. Further computations

The content of this section is intended for the reader who needs to differentiate the matrices of the dynamic model.

2.3.1. Parameters differentiation

In order to compute the derivative of (2.17) with respect to the dynamic parameters, the linearity of the matrices in the parameters themselves is considered.

Proposition 2.5. *If $\boldsymbol{v}_k$ and $\boldsymbol{\omega}_k$ are the linear and angular part of the twist $\boldsymbol{\nu}_k$ respectively, then the generalized momentum $\boldsymbol{\Lambda}_k\boldsymbol{\nu}_k$ can be written as*

$$\boldsymbol{\Lambda}_k\boldsymbol{\nu}_k = \boldsymbol{A}'(\boldsymbol{\nu}_k)\boldsymbol{\pi}_k \,, \tag{2.51}$$

where

$$\boldsymbol{\pi}_k = \begin{bmatrix} m_k & m_k\boldsymbol{r}_k^T & I_{xx_k} & I_{yy_k} & I_{zz_k} & I_{xy_k} & I_{xz_k} & I_{yz_k} \end{bmatrix}^T \,, \tag{2.52}$$

and $\boldsymbol{r}_k$ is the position of the center of mass of the $k-th$ body with respect to its own frame (see Fig. 2.1). Moreover, the rows of $\boldsymbol{A}'(\boldsymbol{\nu}_k) \in \mathbb{R}^{6\times 10}$ for the linear momentum are

$$\boldsymbol{A}'_p = \begin{bmatrix} \boldsymbol{v}_k & \check{\boldsymbol{\omega}}_k & \boldsymbol{O}_{3\times 6} \end{bmatrix} \in \mathbb{R}^{3\times 10} \,, \tag{2.53}$$

and the ones for the angular momentum are

$$\boldsymbol{A}'_l = \begin{bmatrix} \boldsymbol{0} & -\check{\boldsymbol{v}}_k & \check{\check{\boldsymbol{\omega}}}_k \end{bmatrix} \in \mathbb{R}^{3\times 10} \,, \tag{2.54}$$

where $\boldsymbol{O}_{3\times 6} \in \mathbb{R}^{3\times 6}$ is a matrix of zeros and

$$\check{\check{\boldsymbol{a}}} = \begin{bmatrix} a_x & 0 & 0 & a_y & a_z & 0 \\ 0 & a_y & 0 & a_x & 0 & a_z \\ 0 & 0 & a_z & 0 & a_x & a_y \end{bmatrix} \,. \tag{2.55}$$

Proposition 2.6 (Parameters differentiation)**.** *Given the expression of the dynamic matrices in* (2.17) *and the factorization presented in Proposition 2.5, then the derivatives of* (2.17) *with respect to the dynamic parameters are*

$$\frac{\partial \boldsymbol{M}^j}{\partial \boldsymbol{\pi}_{k_h}} = \boldsymbol{J}_k^T \boldsymbol{A}'(\boldsymbol{J}_k^j)^h \,, \tag{2.56a}$$

$$\frac{\partial \boldsymbol{C}^j}{\partial \boldsymbol{\pi}_{k_h}} = \boldsymbol{J}_k^T \left(\boldsymbol{A}'(\mathrm{adj}_{0,k}\,\boldsymbol{J}_k^j + \dot{\boldsymbol{J}}_k^j)^h - \mathrm{adj}_{0,k}^T\,\boldsymbol{A}'(\boldsymbol{J}_k^j)^h \right) \,, \tag{2.56b}$$

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\pi}_{k_h}} = \boldsymbol{J}_k^T \boldsymbol{A}'(\mathrm{Ad}_{0,k}^{-1}\boldsymbol{\gamma}_0)^h \,, \tag{2.56c}$$

where the superscripts j and h indicate the $j-th$ and $h-th$ column of the corresponding matrix.

Proof. Only the computations for the inertia matrix will be considered, since the same can be repeated for the other matrices. From (2.17) one can write

$$\boldsymbol{M}^j = \sum_k \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \boldsymbol{J}_k^j \,. \tag{2.57}$$

From the factorization in Proposition 2.5 it follows that $\boldsymbol{\Lambda}_k\boldsymbol{J}_k^j = \boldsymbol{A}'(\boldsymbol{J}_k^j)\boldsymbol{\pi}_k$. Differentiating the expression with respect to $\boldsymbol{\pi}_{k_h}$, the proposition is proven. □

From (2.17) and Proposition 2.5 is also possible to easily compute the Slotine-Li regressor matrix $\boldsymbol{Y}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{v}_r,\dot{\boldsymbol{v}}_r)$. Using $\boldsymbol{\pi} = \mathrm{col}(\boldsymbol{\pi}_k)$, the identity $\boldsymbol{M}\dot{\boldsymbol{v}}_r + \boldsymbol{C}\boldsymbol{v}_r + \boldsymbol{g} = \boldsymbol{Y}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{v}_r,\dot{\boldsymbol{v}}_r)\boldsymbol{\pi}$ holds, where

$$\boldsymbol{Y}\boldsymbol{\pi} = \sum_k \boldsymbol{J}_k^T \left(\boldsymbol{A}'(\boldsymbol{\alpha}_k) - \mathrm{adj}_{0,k}^T \boldsymbol{A}'(\boldsymbol{J}_k\dot{\boldsymbol{q}}_r)\right)\boldsymbol{\pi}_k \,, \tag{2.58}$$

$\boldsymbol{\alpha}_k = \boldsymbol{J}_k\dot{\boldsymbol{v}}_r + \mathrm{adj}_{0,k}\boldsymbol{J}_k\boldsymbol{v}_r + \dot{\boldsymbol{J}}_k\boldsymbol{v}_r + \boldsymbol{\gamma}_k$ and $k = 1,...,N$. Rewriting (2.58) in terms of stacking matrices leads to the expression of the regressor matrix

$$\boldsymbol{Y} = \bar{\boldsymbol{J}}^T \,\mathrm{blkdiag}\left(\boldsymbol{A}'(\boldsymbol{\alpha}_k) - \mathrm{adj}_{0,k}^T \boldsymbol{A}'(\boldsymbol{J}_k\boldsymbol{v}_r)\right) \,, \tag{2.59}$$

$k = 1,...,N$. Given the Slotine-Li regressor matrix, the classic regressor matrix can be obtained replacing $\boldsymbol{v}_r = \boldsymbol{v}$ and $\dot{\boldsymbol{v}}_r = \dot{\boldsymbol{v}}$.

2.3.2. State differentiation

Although quite involved, the key point for the computation of the derivatives with respect to the state is to find a convenient factorization for $\mathrm{adj}_{0,k}$; similarly to what was done in Sec. 2.3.1 for the momentum[7]. From (2.17), it follows that the derivatives of $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{g}$ can be computed if the derivative of $\boldsymbol{J}_k$, $\dot{\boldsymbol{J}}_k$, $\mathrm{Ad}_{0,k}^{-1}$ and $\mathrm{adj}_{0,k}$ are available.

Proposition 2.7. *Given the identity $\boldsymbol{J}_k\dot{\boldsymbol{q}} = \sum_h \boldsymbol{J}_k^h\dot{\boldsymbol{q}}_h$, it follows that*

$$\frac{\partial\,\mathrm{Ad}_{0,k}^{-1}}{\partial\boldsymbol{q}_h} = -\,\mathrm{adj}_{\boldsymbol{J}_k^h}\mathrm{Ad}_{0,k}^{-1} \,, \tag{2.60}$$

$$\frac{\partial\,\mathrm{adj}_{0,k}}{\partial\dot{\boldsymbol{q}}_h} = \mathrm{adj}_{\boldsymbol{J}_k^h} \,, \tag{2.61}$$

where $\mathrm{adj}_{\boldsymbol{J}_k^h}$ is the matrix computed using $\boldsymbol{J}_k^h \in \mathbb{R}^6$, instead of $\boldsymbol{\nu}_k \in \mathbb{R}^6$, while

$$\frac{\partial\,\mathrm{adj}_{0,k}}{\partial\boldsymbol{q}_h} = \mathrm{adj}_{\frac{\partial\boldsymbol{J}_k}{\partial\boldsymbol{q}_h}\dot{\boldsymbol{q}}} \,, \tag{2.62}$$

where in $\mathrm{adj}_{\frac{\partial\boldsymbol{J}_k}{\partial\boldsymbol{q}_h}\dot{\boldsymbol{q}}}$ this time $\boldsymbol{\nu}_k \in \mathbb{R}^6$ has been replaced by $\frac{\partial\boldsymbol{J}_k}{\partial\boldsymbol{q}_h}\dot{\boldsymbol{q}} \in \mathbb{R}^6$.

Proof. Firstly, it can be verified that the following identities hold

$$\frac{\partial\,\mathrm{Ad}_{0,k}^{-1}}{\partial\boldsymbol{q}}\dot{\boldsymbol{q}} = \dot{\mathrm{Ad}}_{0,k}^{-1} = -\,\mathrm{adj}_{0,k}\mathrm{Ad}_{0,k}^{-1} \,, \tag{2.63}$$

$$\mathrm{adj}_{0,k} = \sum_h \mathrm{adj}_{\boldsymbol{J}_k^h\dot{\boldsymbol{q}}_h} = \sum_h \mathrm{adj}_{\boldsymbol{J}_k^h}\dot{\boldsymbol{q}}_h \,. \tag{2.64}$$

Substituting (2.64) in (2.63), one obtains an equality that, being true $\forall\dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$, leads to (2.60). On the other hand the differentiation of (2.64) with respect to $\dot{\boldsymbol{q}}_h$ results in (2.61), while the one with respect to $\boldsymbol{q}_h$ results in (2.62). □

[7] The results of this section must be intended only for the cases when the differentiation with the respect to the state is actually meaningful. This is always the case when a classical manipulator with 1 - DoF joints is considered.

Proposition 2.8 (State differentiation)**.** *Given the expression of the dynamic matrices in* (2.17) *and the factorization presented in Proposition 2.7, then the derivatives of* (2.17) *with respect to the state are*

$$\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}_h} = \sum_k \frac{\partial \boldsymbol{J}_k^T}{\partial \boldsymbol{q}_h} \boldsymbol{\Lambda}_k \boldsymbol{J}_k + \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}_h} , \tag{2.65a}$$

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}_h} = \sum_k \frac{\partial \boldsymbol{J}_k^T}{\partial \boldsymbol{q}_h} \left(\boldsymbol{\Psi}_k \boldsymbol{J}_k + \boldsymbol{\Lambda}_k \dot{\boldsymbol{J}}_k \right) + \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \frac{\partial \dot{\boldsymbol{J}}_k}{\partial \boldsymbol{q}_h} + \boldsymbol{J}_k^T \left[\left(\boldsymbol{\Lambda}_k \operatorname{adj}_{\frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}_h} \dot{\boldsymbol{q}}} - \operatorname{adj}^T_{\frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}_h} \dot{\boldsymbol{q}}} \boldsymbol{\Lambda}_k \right) \boldsymbol{J}_k + \boldsymbol{\Psi}_k \frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}_h} \right] , \tag{2.65b}$$

$$\frac{\partial \boldsymbol{C}}{\partial \dot{\boldsymbol{q}}_h} = \sum_k \boldsymbol{J}_k^T \left[\left(\boldsymbol{\Lambda}_k \operatorname{adj}_{\boldsymbol{J}_k^h} - \operatorname{adj}^T_{\boldsymbol{J}_k^h} \boldsymbol{\Lambda}_k \right) \boldsymbol{J}_k + \boldsymbol{\Lambda}_k \frac{\partial \dot{\boldsymbol{J}}_k}{\partial \dot{\boldsymbol{q}}_h} \right] , \tag{2.65c}$$

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}_h} = \sum_k \left(\frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}_h}^T \boldsymbol{\Lambda}_k - \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \operatorname{adj}_{\boldsymbol{J}_k^h} \right) \operatorname{Ad}^{-1}_{0,k} \boldsymbol{\gamma}_0 , \tag{2.65d}$$

where $k = 1, \ldots, N$*. In the previous relationships,* $\boldsymbol{C}$ *is the only matrix which depends on both* $\boldsymbol{q}$ *and* $\dot{\boldsymbol{q}}$*. Notice that for the only term differentiated with respect to* $\dot{\boldsymbol{q}}$*, the following holds*

$$\frac{\partial \dot{\boldsymbol{J}}_k}{\partial \dot{\boldsymbol{q}}} = \frac{\partial}{\partial \dot{\boldsymbol{q}}} \left(\frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} \right) = \frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{q}} . \tag{2.66}$$

2.3.3. Time differentiation

Obviously the derivatives of (2.17) with respect to time could be computed using the chain rule and (2.65). Nevertheless, if the derivatives with respect to the state are not required for the considered application, it is more efficient to directly differentiate (2.17) with respect to time. This is quite clear considering that the derivative of a matrix with respect to a vector is an order three tensor, while the derivative with respect to a scalar is also a matrix.

Proposition 2.9 (Time differentiation)**.** *Given the expression of the dynamic matrices in* (2.17)*, then the derivatives of* (2.17) *with respect to time are*

$$\dot{\boldsymbol{M}} = \sum_k \dot{\boldsymbol{J}}_k^T \boldsymbol{\Lambda}_k \boldsymbol{J}_k + \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \dot{\boldsymbol{J}}_k , \tag{2.67a}$$

$$\dot{\boldsymbol{C}} = \sum_k \dot{\boldsymbol{J}}_k^T \left(\boldsymbol{\Psi}_k \boldsymbol{J}_k + \boldsymbol{\Lambda}_k \dot{\boldsymbol{J}}_k \right) + \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \ddot{\boldsymbol{J}}_k + \boldsymbol{J}_k^T \left[\left(\boldsymbol{\Lambda}_k \dot{\operatorname{adj}}_{0,k} - \dot{\operatorname{adj}}^T_{0,k} \boldsymbol{\Lambda}_k \right) \boldsymbol{J}_k + \boldsymbol{\Psi}_k \dot{\boldsymbol{J}}_k \right] , \tag{2.67b}$$

$$\dot{\boldsymbol{g}} = \sum_k \left(\dot{\boldsymbol{J}}_k^T \boldsymbol{\Lambda}_k - \boldsymbol{J}_k^T \boldsymbol{\Lambda}_k \operatorname{adj}_{0,k} \right) \operatorname{Ad}^{-1}_{0,k} \boldsymbol{\gamma}_0 , \tag{2.67c}$$

where $k = 1, \ldots, N$*.*

From a computational point of view, roughly speaking, each additional differentiation of the dynamic matrices with respect to time, requires the propagation of an additional derivative of $\dot{\boldsymbol{J}}_k$ according to the Algorithm 1 and the knowledge of an additional time derivative of $\boldsymbol{v}$.

2.4. Summary

In this chapter, the dynamic equations of a system of interconnected rigid bodies have been considered. The formulation of the Newton-Euler equation in terms of twists and wrenches was the starting point to obtain a closed form expressions for the dynamic matrices. An algorithm using

a “Newton-Euler like” outward recursion (whose computation is very efficient being the first step of the Newton-Euler algorithm) was presented in order to compute the quantities of interest. Additionally, the reduced model for elastically actuated robot was discussed, together with all the assumptions leading to it. Finally, some properties of the dynamic model were introduced. It has been shown that a coordinate transformation can produces a decoupled structure of the equations of motion of a floating base robot. The transformation is derived based on first principles of mechanics and therefore has a clear physical interpretation.

CHAPTER 3

The Spring Loaded Inverted Pendulum

The spring loaded inverted pendulum (SLIP) is a template model which, despite its simplicity, has been proven to produce accurate CoM patterns and contact force profiles for different kind of locomotions [SGGB02, GSB06]. In section 3.1.1 the biped variant of this model is presented, which is also the one that will be considered throughout the chapter. As mentioned in chapter 1 the underlying assumptions will be to have no underactuation at all in the system. As no flight phases are allowed by the simplifying hypothesis, walking patterns will be considered, which motivates the choice of considering the biped variant of the model.

The contribution of this chapter is to provide methods for mapping the simple SLIP model on a multi-body robot, in order to obtain a walking pattern and insights on the use of elasticity in locomotion. The goal is not to assume to have a robot designed to match the behavior of the SLIP model [AS10, KKH10], nor to track the behavior of a passive walker [AB06], but to fully actuate the robot in order to reproduce, via the controller, the behavior of a passive walker to obtain a periodic gait, as sketched in Fig. 3.1. In this way, the benefits of an open loop stable limit cycle are utilized for a robot not tailored to a specific set of periodic walking patterns, but capable of a variety of motions. Additionally, predefined trajectory for the CoM and foot steps are replaced by a target self stabilizing dynamics, i.e. the SLIP behavior.

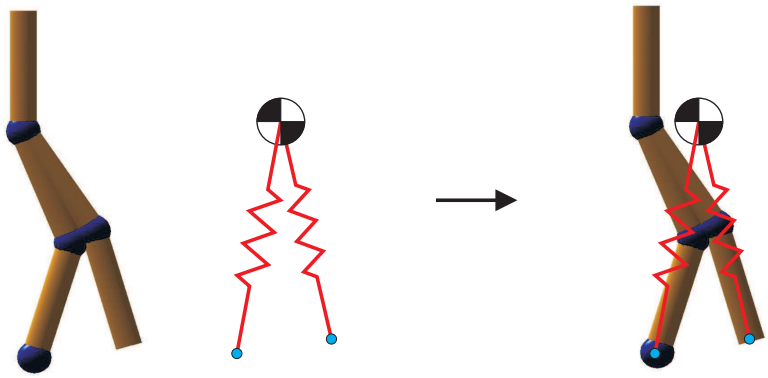

Figure 3.1.: The input torques can reshape the dynamics of the robot in order to make it as close as possible to the SLIP model.

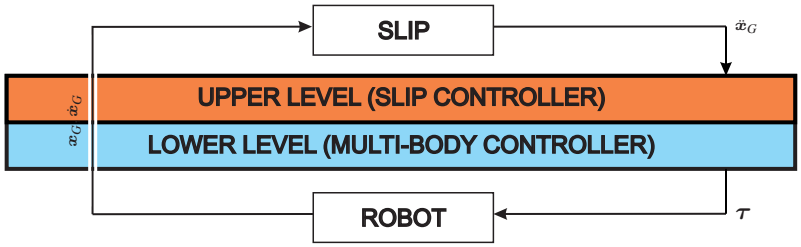

Figure 3.2.: The controller is the interface between the virtual SLIP model and the real robot.

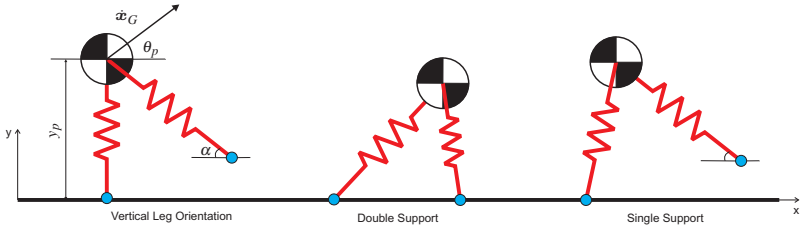

Figure 3.3.: SLIP model in the vertical leg orientation (VLO) condition, double support phase and single support phase.

On the other hand, due to its simplicity, the SLIP model is way far from a complete dynamic description of the entire body motion. The effects of the swing leg are completely neglected in this simple model, while they turn out to have a big influence on the torques of the robot. The foot trajectory of the swing leg is considered in section 3.2, the double support phase in section 3.1.2.

3.1. Modeling

Loosely speaking, the controller proposed in the next section plays the role of an interface between the dynamics of the SLIP model and the one of the robot, making it possible to map the behavior of the first onto the latter, as shown in Fig. 3.2. Before describing the two levels of which the controller consists, it is therefore interesting to consider the dynamic model of the single not interconnected levels.

3.1.1. Dynamic model of the SLIP

As it is shown in Fig. 3.3, the SLIP model is simply given by two massless linear springs connected to each other at the hip, where all the mass m is concentrated. Due to this assumption the swing leg can instantaneously change its configuration when it takes off from the ground, to reach the desired angle of attack α for the next step.

Using $\boldsymbol{f}_L(\boldsymbol{x}_G)$ and $\boldsymbol{f}_R(\boldsymbol{x}_G)$ to denote the spring forces acting at the CoM $\boldsymbol{x}_G$ due to the left and right leg respectively, the dynamic equation of the system can be written as:

$$\ddot{\boldsymbol{x}}_G = \frac{1}{m}\Big(\boldsymbol{f}_L(\boldsymbol{x}_G) + \boldsymbol{f}_R(\boldsymbol{x}_G)\Big) + \boldsymbol{g}_0 \,, \tag{3.1}$$

where $\boldsymbol{g}_0$ is the gravitational acceleration of the Earth. Introducing $\boldsymbol{x}_{F_i}$ with $i = \{Left, Right\}$ for the foot contact point, then:

$$\boldsymbol{f}_i(\boldsymbol{x}_G) = k\Big(l_0 - \|\boldsymbol{x}_G - \boldsymbol{x}_{F_i}\|\Big)\frac{\boldsymbol{x}_G - \boldsymbol{x}_{F_i}}{\|\boldsymbol{x}_G - \boldsymbol{x}_{F_i}\|}, \qquad i = \{L,R\} \,, \tag{3.2}$$

where, k is the stiffness constant and l_0 the rest length of the springs. In the following, the foot contact point will be simply referred to as $\boldsymbol{x}_F$, but obviously $\boldsymbol{x}_F = \boldsymbol{x}_{F_L}$ or $\boldsymbol{x}_F = \boldsymbol{x}_{F_R}$ depending on the stance phase.

Depending on the values of the parameters in the dynamic model, it is possible to obtain self stabilizing periodic walking patterns which, as it was already mentioned, exhibit CoM trajectories and ground reaction forces remarkably similar to human data. Extensive works can be found in the literature which provide methods to choose such parameters in order to modify walking speed and step length, or to increase the basin of attraction of the asymptotically stable limit cycle [EGB09, RBS10, KW11, MSC12].

The classic approach to study the stability of the SLIP model is to consider a Poincaré section [Kha02] and check for the eigenvalues of the monodromy matrix. A particularly convenient section is the one corresponding to the condition of vertical leg orientation (VLO) [RBS10], see Fig. 3.3. In this case, the conservative nature of the SLIP model allows to reduce by one the dimension of the state space. It is sufficient to use the CoM height y_p, its position along the forward direction and the angle of the CoM velocity respect to the floor θ_p as coordinates, since the magnitude of the velocity can be derived from the value of the constant total energy of the system (the massless legs will not dissipate any energy at the impacts). Additionally, as the position along the forward direction has no influence on the periodic behavior of the system, it will be neglected in the analysis of the limit cycle. Therefore, the Poincaré section has dimension two instead of dimension four of the original system.

3.1.2. Dynamic model of the multi-body robot

The dynamic model of the five revolute joints planar robot considered here can be obtained as in chapter 2. The following simplifications will be considered:

Assumption 3.1. *The stance foot is rigidly connected to the floor, until it is commanded to leave the ground.*

Assumption 3.2. *The feet are massless.*

Assumption 3.3. *The impact of the foot with the floor is inelastic.*

Due to the first assumption, the model of the robot is the one of a classic robotic manipulator with a branched structure. It can be derived either from the free floating model (2.16) upon introduction of the stance foot constraint, or using directly a minimal set of coordinates, i.e. the joint angles represented in Fig. 3.4. The first method is completely analogous to what will be done in section 3.2 in order to derived the model for the double support phase. For convenience, the model is repeated here:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}\,, \tag{3.3}$$

where in this case $\boldsymbol{q}, \boldsymbol{g}(\boldsymbol{q}), \boldsymbol{\tau} \in \mathbb{R}^5$ and $\boldsymbol{M}(\boldsymbol{q}), \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{5\times 5}$.

Due to the possibility of having either a single or double support phase, the model is complete when also the transitions between these two are considered. These transitions are caused by the two events touch down and take off of the legs, as shown in Fig. 3.5. Moreover, because of the impact, an energy loss will occur at the touch down instants. This can be modeled considering that, at these instants, also an impulsive force is acting on the system and that it will produce an instantaneous change of the velocity (but not of the position) [MW06]. This means that the system will exhibit a variation of the generalized momentum in the joint space that can be expressed as:

$$\Delta(\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}) = \boldsymbol{M}(\boldsymbol{q})\Delta\dot{\boldsymbol{q}} = \boldsymbol{J}_c^T(\boldsymbol{q})\boldsymbol{\lambda}\,, \tag{3.4}$$

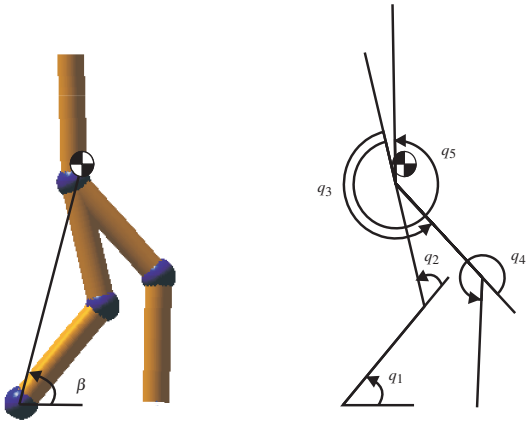

Figure 3.4.: Multi-body robot and joint angles.

where $\boldsymbol{J}_c(\boldsymbol{q}) \in \mathbb{R}^{2\times 5}$ is the Jacobian matrix that maps the joint velocities into the velocity along the constrained directions, while $\boldsymbol{\lambda} \in \mathbb{R}^2$ is the reaction along the same directions. If the impact is modeled as an inelastic impact, then the following relation must be considered:

$$\boldsymbol{J}_c(\boldsymbol{q})\dot{\boldsymbol{q}}^+ = \boldsymbol{0} \qquad \Longleftrightarrow \qquad \boldsymbol{J}_c(\boldsymbol{q})\Delta\dot{\boldsymbol{q}} = -\boldsymbol{J}_c(\boldsymbol{q})\dot{\boldsymbol{q}}^- \,, \tag{3.5}$$

where the superscripts $^-$ and $^+$ are used to express the quantities before and after the impact, respectively. Solving the system of equations (3.4) and (3.5) leads to:

$$\boldsymbol{\lambda} = -\left(\boldsymbol{J}_c\boldsymbol{M}^{-1}\boldsymbol{J}_c^T\right)^{-1}\boldsymbol{J}_c\dot{\boldsymbol{q}}^- \tag{3.6}$$

$$\dot{\boldsymbol{q}}^+ = \left(\boldsymbol{E}_5 - \boldsymbol{J}_c^{+M}\boldsymbol{J}_c\right)\dot{\boldsymbol{q}}^- \,, \tag{3.7}$$

where $\boldsymbol{E}_5 \in \mathbb{R}^{5\times 5}$ is the identity matrix and $\boldsymbol{J}_c^{+M} = \boldsymbol{M}^{-1}\boldsymbol{J}_c^T\left(\boldsymbol{J}_c\boldsymbol{M}^{-1}\boldsymbol{J}_c^T\right)^{-1}$. In (3.7), it is possible to recognize a nullspace projection matrix [SSVO08], which projects the joint velocities before the impact into the null space of the Jacobian matrix $\boldsymbol{J}_c$. Therefore, an impact modeled in this way can be interpreted saying that only the component of $\dot{\boldsymbol{q}}$ that is causing zero velocity at the contact point is kept after the impact. The energy loss is given by:

$$\begin{aligned} 2\Delta H &= \dot{\boldsymbol{q}}^{+^T}\boldsymbol{M}\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^{-^T}\boldsymbol{M}\dot{\boldsymbol{q}}^- \\ &= \dot{\boldsymbol{q}}^{-^T}\left[\left(\boldsymbol{E} - \boldsymbol{J}_c^{+M}\boldsymbol{J}_c\right)^T\boldsymbol{M}\left(\boldsymbol{E} - \boldsymbol{J}_c^{+M}\boldsymbol{J}_c\right) - \boldsymbol{M}\right]\dot{\boldsymbol{q}}^- \\ &= -\dot{\boldsymbol{q}}^{-^T}\boldsymbol{J}_c^T\left(\boldsymbol{J}_c\boldsymbol{M}^{-1}\boldsymbol{J}_c^T\right)^{-1}\boldsymbol{J}_c\dot{\boldsymbol{q}}^- \,, \end{aligned} \tag{3.8}$$

where $\boldsymbol{J}_c(\boldsymbol{q})\dot{\boldsymbol{q}}^-$ is the velocity of the impact point right before the impact, in other words the pseudo kinetic energy related to the impact point is the energy that will be lost in the impact.

The touch down event initiates the double support phase. Assuming to have no sliding between the feet and the rigid floor, an additional constraint must be added to the model in this phase (just like has been previously done for modeling the impact):

$$\boldsymbol{J}_c(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0} \,, \tag{3.9}$$

which can also be expressed as:

$$\boldsymbol{J}_{H_L}(\boldsymbol{q}_L)\dot{\boldsymbol{q}}_L = \boldsymbol{J}_{H_R}(\boldsymbol{q}_R)\dot{\boldsymbol{q}}_R \,, \tag{3.10}$$

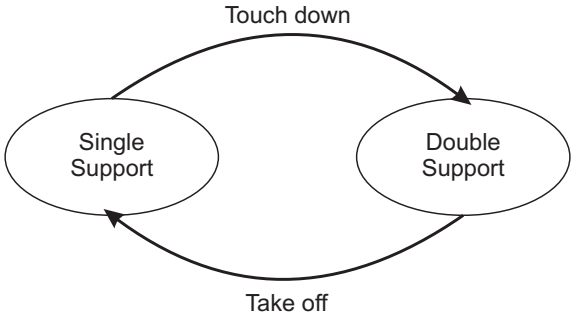

Figure 3.5.: Automaton describing the evolution of the discrete part of the system.

where $\boldsymbol{J}_{H_i}(\boldsymbol{q}_i)$, $i = \{L,R\}$ gives the velocity of the hip with respect to the correspondent foot and $\boldsymbol{q}_i$, $i = \{L,R\}$ are the joint variables of the correspondent leg. Assuming that at least one of the two Jacobian matrices is nonsingular[1], then:

$$\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{H_i}^{-1}\left(\boldsymbol{q}_i\right)\boldsymbol{J}_{H_j}\left(\boldsymbol{q}_j\right)\dot{\boldsymbol{q}}_j, \tag{3.11}$$

and is possible to write $\dot{\boldsymbol{q}} = \boldsymbol{\Psi}(\boldsymbol{q})\dot{\boldsymbol{q}}_a$, being:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{q}}_i \\ \dot{\boldsymbol{q}}_j \\ \dot{q}_5 \end{bmatrix} \qquad \boldsymbol{\Psi}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{H_i}^{-1}\boldsymbol{J}_{H_j} & 0 \\ \boldsymbol{E} & 0 \\ \boldsymbol{0} & 1 \end{bmatrix} \qquad \dot{\boldsymbol{q}}_a = \begin{bmatrix} \dot{\boldsymbol{q}}_j \\ \dot{q}_5 \end{bmatrix}, \tag{3.12}$$

where q_5 is the joint angle of the trunk not affected by the change between single and double support. A relationship for the joint acceleration can be obtained differentiating (3.12) with respect to time. Substituting $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ in (3.3), and pre-multiply by $\boldsymbol{\Psi}^T(\boldsymbol{q})$, the model of the robot during the double support phase is obtained:

$$\boldsymbol{M}_a(\boldsymbol{q})\ddot{\boldsymbol{q}}_a + \boldsymbol{C}_a(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_a + \boldsymbol{g}_a(\boldsymbol{q}) = \boldsymbol{\tau}_a\,, \tag{3.13}$$

where $\boldsymbol{M}_a = \boldsymbol{\Psi}^T\boldsymbol{M}\boldsymbol{\Psi}$, $\boldsymbol{C}_a = \boldsymbol{\Psi}^T\left(\boldsymbol{C}\boldsymbol{\Psi} + \boldsymbol{M}\dot{\boldsymbol{\Psi}}\right)$, $\boldsymbol{g}_a = \boldsymbol{\Psi}^T\boldsymbol{g}$ and $\boldsymbol{\tau}_a = \boldsymbol{\Psi}^T\boldsymbol{\tau}$.

The flow maps, which describe the continuous dynamics of the system with a set of ordinary differential equations together with the set of discrete states and their transition constitute the complete model description of this switching system [Lib03].

3.2. Swing leg treatment

The presence of the swing leg is the element that clearly mostly differs between the SLIP model and the considered planar robot. While the SLIP model defines the CoM trajectory and the target angle of attack for the next step, it is still necessary to define a trajectory for the swing leg, since this information is completely missing in the SLIP model. In order to be as close as possible to the virtual system that the robot is trying to reproduce, a good choice could be to just plan the relative position of the swing foot with respect to the CoM, ensuring that the foot will have reached the desired relative position (defined by the desired angle of attack) before the touch down event. Furthermore, the angle $\beta(\boldsymbol{x}_G)$ related to the line from the stance foot to the CoM, shown in Fig. 3.4, is used as a parameter for the trajectory. This is a standard choice for biped robots

[1] If both are singular the system is in a complete rest.

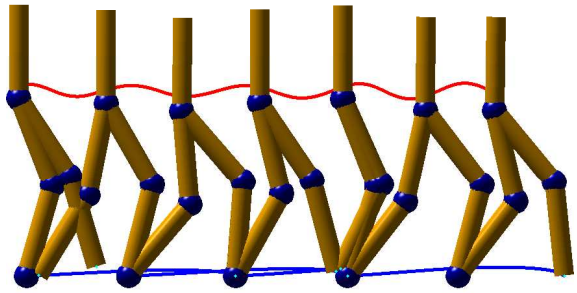

Figure 3.6.: Consecutive steps of the robot. The red and blue lines are respectively the CoM and swing foot trajectory.

based on hybrid zero dynamics as it guarantees a coordination between the forward movement of the robot and the position of the swing leg [WGC+07]. Then the desired foot position is given by

$$\boldsymbol{x}_{F_d} = \boldsymbol{x}_G + \boldsymbol{\Pi}(\beta(\boldsymbol{x}_G)) \,, \tag{3.14}$$

where $\boldsymbol{\Pi}(\beta(\boldsymbol{x}_G))$ is the interpolating function describing the shape of the foot trajectory relative to the CoM. Since it is also necessary to ensure the continuity of the desired trajectory, a cubic spline with desired initial and final values and two "virtual instants" is used [SSVO08]. The desired foot position cannot be used as a desired value for a conventional trajectory tracking controller since it depends on the current state. Instead, we formulate a virtual constraint [WGC+07] by defining the error term

$$\boldsymbol{\xi} = \boldsymbol{x}_F - \boldsymbol{x}_G - \boldsymbol{\Pi}(\beta(\boldsymbol{x}_G)) \,, \tag{3.15}$$

which should be kept equal to $\boldsymbol{\xi}_d = \mathbf{0}$ by the controller. The angle β, instead of the time, is used to parameterize the offset of the swing foot with respect to the CoM. As a result the swing foot position is dependent on the CoM and no information about the time necessary to complete one step is needed. Notice that, except for the swing leg, the planning in the control design process has been replaced requiring a desired dynamic behavior rather than desired trajectories. Knowing the actual position and velocity of the CoM of the robot, the expected acceleration is the one that a SLIP model would have had in that state.

Figure 3.6 shows overlapped configurations of the robot during the path. The red line represent the CoM trajectory, while the blue line the swing foot trajectory.

3.3. Control

The control action is acting both on the virtual SLIP model and the actual robot. In the following, they will be referred to as upper level control and lower level control respectively. The upper level control modifies the dynamics of the original SLIP model. The result of this action is to provide a target dynamics for the lower level control. The latter represents the real control action, as it will determine the torques that the motors need to provide in order to actuate the robot.

3.3.1. Upper Level: Total energy control

While the SLIP template is a conservative system, the multi-body robot will have energy losses at the impacts, as computed in (3.8). The upper level control is responsible to cope with this difference. The energy control law acts based on the state of the SLIP model and is not only

injecting energy in the system when the total level decreases, but it also dissipates energy when the level increases. This is different from what was done in [HRHS10], where an energy correction law is not continuously acting (based on the error between the desired level and the measured one), but intervening before the impact based on a precomputed value of the energy loss.

The starting point is to consider the expression of the power:

$$\dot{H} = \dot{\boldsymbol{x}}_G^T \Big(m\ddot{\boldsymbol{x}}_G - \boldsymbol{f}_L - \boldsymbol{f}_R - m\boldsymbol{g}_0 \Big) \,. \tag{3.16}$$

Since the power is the derivative of the energy and the desired value of the energy is constant, the problem of regulating the energy to the desired value can be solved via an energy controller that ensures:

$$\dot{H} = -K_H \Big(H - H_d \Big) \,, \tag{3.17}$$

where K_H is a positive gain and H_d is the desired constant total energy. This is possible if the resulting acceleration on the CoM is the one of a pure SLIP model plus an additional term with the following expression:

$$\Delta \boldsymbol{g} = -\frac{1}{m} \frac{\dot{\boldsymbol{x}}_G}{\|\dot{\boldsymbol{x}}_G\|^2} K_H \Big(H - H_d \Big) \,. \tag{3.18}$$

Because of this additional term, the SLIP model can be imagined as walking under the effect of a virtual state dependent gravity field. The term $\Delta \boldsymbol{g}$ is always in the direction of the CoM velocity, increasing or decreasing the kinetic energy in order to keep the total energy to the desired value. Equation (3.1) thus is modified to

$$\ddot{\boldsymbol{x}}_G = \frac{1}{m} \Big(\boldsymbol{f}_L(\boldsymbol{x}_G) + \boldsymbol{f}_R(\boldsymbol{x}_G) \Big) + \boldsymbol{g}_v \,, \tag{3.19}$$

with $\boldsymbol{g}_v = \boldsymbol{g}_0 + \Delta \boldsymbol{g}$.

3.3.2. Lower level: Multi-Body Control

In the following two different approaches will be presented in order to make the robot behave as close as possible to the simple virtual model (3.19). The first approach is based on an exact feedback linearization of the robot dynamics and can be seen as an extension of the control law presented in [HRHS10] from an hopper to a bipedal SLIP for walking. The second approach, presented in section 3.3.2, aims at a simpler control law which requires less information about the system.

Feedback linearization

The SLIP model can be mapped on the real robot dynamics at an acceleration level. The idea is to ensure that the acceleration of the CoM of the real robot is the same as in the SLIP model. The control law can be derived following the same steps needed to derive the classic impedance control law with feedback linearization.

Three main tasks can be identified:

- CoM acceleration $\ddot{\boldsymbol{x}}_G$
- virtual constraint for the swing foot trajectory $\boldsymbol{\xi}$
- trunk orientation $\theta_T = q_1 + q_2 + q_5$.

The corresponding velocities can be computed from

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t(\boldsymbol{q})\dot{\boldsymbol{q}} \,, \tag{3.20}$$

where $\dot{\boldsymbol{x}}_t = \begin{bmatrix} \dot{\boldsymbol{x}}_G^T & \dot{\boldsymbol{\xi}}^T & \dot{\theta}_T^T \end{bmatrix}^T$ is the velocity in the task space and $\boldsymbol{J}_t(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_G^T & \boldsymbol{J}_\xi^T & \boldsymbol{J}_\theta^T \end{bmatrix}^T$ the corresponding Jacobian matrix. Considering the time derivative of (3.20) and substituting $\ddot{\boldsymbol{q}}$ from (3.3), one gets:

$$\ddot{\boldsymbol{x}}_t = \boldsymbol{J}_t \boldsymbol{M}^{-1}\Big(\boldsymbol{\tau} - \boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{g}\Big) + \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}} \,. \tag{3.21}$$

At this point with the following choice of the torques[2]:

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{J}_t^{-1}\Big(\ddot{\boldsymbol{x}}_{t_d} - \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}}\Big) + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} \,, \tag{3.22}$$

it is guaranteed that $\ddot{\boldsymbol{x}}_t = \ddot{\boldsymbol{x}}_{t_d}$. Finally, choosing

$$\ddot{\boldsymbol{x}}_{t_d} := \begin{bmatrix} \frac{1}{m}\Big(\boldsymbol{f}_L(\boldsymbol{x}_G) + \boldsymbol{f}_R(\boldsymbol{x}_G)\Big) + \boldsymbol{g}_v \\ -\boldsymbol{K}_\xi\boldsymbol{\xi} - \boldsymbol{D}_\xi\dot{\boldsymbol{\xi}} \\ \ddot{\theta}_{T_d} - K_T\tilde{\theta}_T - D_T\dot{\tilde{\theta}}_T \end{bmatrix} \,, \tag{3.23}$$

where $K_T, D_T > 0$, $\boldsymbol{K}_\xi, \boldsymbol{D}_\xi \in \mathbb{R}^{2\times 2}$ are positive definite matrices and $\tilde{\theta}_T = \theta_T - \theta_{T_d}$, ensures that $\ddot{x}_G$ corresponds exactly to (3.19) and that $\boldsymbol{\xi} \to \boldsymbol{0}$, $\theta_T \to \theta_{T_d}$ as $t \to \infty$.

Multi-priority control law

Since three different types of tasks are considered, one can separate them and use simpler control laws to fulfill a single task. The required torques can be added using a null space projector. A higher priority will be assigned to the CoM task, since the main goal is to reproduce a SLIP like motion. A further simplification for this task is to just produce the forces of virtual springs attached to the CoM. Once it has been understood from the SLIP model that the presence of compliant elements is useful, it is convenient to design simpler control laws than the feedback linearization while the self-stabilizing behavior will be augmented by the upper level SLIP stabilization. In other words, the goal for this task is to map the virtual spring forces in joint torques accordingly to the transpose of the Jacobian matrix that relates the joint velocity to the CoM velocity (without compensation of the nonlinear terms):

$$\boldsymbol{\tau}_G = \boldsymbol{J}_G^T\Big(\boldsymbol{f}_L(\boldsymbol{x}_G) + \boldsymbol{f}_R(\boldsymbol{x}_G) + m\Delta\boldsymbol{g}\Big) \,, \tag{3.24}$$

where obviously also the extra component of the virtual gravity field must be taken into account.

Assuming, without loss of generality, that $\dot{\theta}_{T_d} = 0$, for the other tasks simple PD control laws with gravity compensation can be used:

$$\boldsymbol{\tau}_{PD} = \begin{bmatrix} \boldsymbol{J}_\xi^T & \boldsymbol{J}_\theta^T \end{bmatrix} \begin{bmatrix} -\boldsymbol{K}_\xi\boldsymbol{\xi} - \boldsymbol{D}_\xi\dot{\boldsymbol{\xi}} + \boldsymbol{g}_\xi \\ -K_T\tilde{\theta}_T - D_T\dot{\theta}_T + \boldsymbol{g}_\theta \end{bmatrix} \,, \tag{3.25}$$

where $\boldsymbol{g}_\xi$ and $\boldsymbol{g}_\theta$ are the projections of the gravity torques in the corresponding space.

This control law is known to be asymptotically stable and less sensitive to model uncertainties than feedback linearization [PP88]. At this point adding the resulting torques will cause the two

[2] In the following it will be assumed to be far enough from the singular configurations. Further on in the section, additional torques will be introduced in order to cope with this problem.

tasks to interfere with each other. To avoid this, the torques computed for the second task will be projected in the null space of the transpose of the Jacobian matrix of the higher priority task. This means that the torques produced to fulfill the second task will produce no force on the CoM which will mainly move under the effect of the gravity field and the virtual springs. Obviously this also means that the second priority task will be fulfilled as long as it is not interfering with the other one.

Before concluding this section it is worth to mention that also torques that push the robot away from the singularity should always be considered. The classic approach is to consider the kinematic manipulability measure [Yos90]:

$$m_{kin_i}(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}_{H_i}\boldsymbol{J}_{H_i}^T)} \qquad i = \{L,R\} \tag{3.26}$$

and then choosing a singularity avoidance potential. In addition it is possible to consider a damping term also for this task. The idea is that, considering a quadratic form for the singularity avoidance potential, the corresponding torque will be similar to a spring like term where the transpose of $\boldsymbol{J}_{kin}(\boldsymbol{q}) = \frac{\partial m_{kin}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ is responsible for projecting the force from one space to the other. The singularity avoidance torques are:

$$\boldsymbol{\tau}_{SA_i} = \begin{cases} -\boldsymbol{J}_{kin_i}^T\left(K_{SA}\tilde{m}_{kin_i} + D_{SA}\boldsymbol{J}_{kin_i}\dot{\boldsymbol{q}}\right) & m_{kin_i} \leqslant m_0 \\ \boldsymbol{0} & m_{kin_i} > m_0 \end{cases}, \tag{3.27}$$

where $\tilde{m}_{kin_i} = m_{kin_i} - m_0$ and m_0 is a threshold which indicates that the robot is close enough to the singular configuration to activate the corresponding torques.

The final expression of the torques is then:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{SA_L} + \boldsymbol{\tau}_{SA_R} + \boldsymbol{\tau}_G + \left(\boldsymbol{E} - \boldsymbol{J}_G^T\boldsymbol{J}_G^{+_M T}\right)\boldsymbol{\tau}_{PD}\,. \tag{3.28}$$

Double support and redistribution of torques

As seen in 3.1.2 the expression of the dynamic model during the double support is formally equal to the one obtained for the single support. This means that the control laws computed before can be directly applied also in this case provided to consider the matrices of the new dynamic model instead of the old ones and to remove the swing foot coordinates from the task space. In this way, it is possible to compute the torques $\boldsymbol{\tau}_a$ in (3.13) and not all the torques $\boldsymbol{\tau}$. To obtain the latter the relation $\boldsymbol{\tau}_a = \boldsymbol{\Psi}^T\boldsymbol{\tau}$ between the torques is taken into account, from which:

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{\Psi}^{+_W T}\boldsymbol{\tau}_a + \left(\boldsymbol{E} - \boldsymbol{\Psi}^{+_W T}\boldsymbol{\Psi}^T\right)\boldsymbol{\tau}_d \\ \boldsymbol{\Psi}^{+_W T} &= \boldsymbol{W}^{-1}\boldsymbol{\Psi}\left(\boldsymbol{\Psi}^T\boldsymbol{W}^{-1}\boldsymbol{\Psi}\right)^{-1}, \end{aligned}$$

where $\boldsymbol{W}$ is a positive definite matrix, which induces a weighted norm for the torques. It is useful to penalize a big value of some torques (e.g. ankle torque) more than the others. In addition, the associated null space projector is used to keep the torques as close as possible to some desired value.

3.4. Simulations and stability analysis

In the following simulation study, the cyclic behavior of the SLIP model is compared with the one of the multi-body robot using the two proposed controllers. The parameters of the SLIP are

Table 3.1.: SLIP model parameters.

mass	$m = 80\,\mathrm{kg}$
angle of attack	$\alpha = 1.2\,\mathrm{rad}$
rest length	$l_0 = 1\,\mathrm{m}$
stiffness	$k = 15696\,\mathrm{N/m}$
total energy	$H = 824\,\mathrm{J}$

Table 3.2.: Multy-body robot parameters.

	shank	thigh	trunk
mass	5 kg	5 kg	60 kg
inertia	0.01 kg m^2	0.01 kg m^2	1 kg m^2
length	0.48 m	0.48 m	0.48 m

adopted from the simulation results in [RBS10] and are shown in Table 3.1. The parameters of the multi-body dynamics are shown in Table 3.2. The CoM of each segment was set to the center of the segment. Moreover a closed loop inverse kinematic (CLIK) algorithm [SSVO08] is used in order to ensure that both system will start with the same initial conditions and in the same configuration (VLO configuration).

The simulation environment is implemented in MATLAB/Simulink ® using a sampling time of 0.1 ms. Figure 3.7 shows the vertical motion of the CoM for the SLIP model and the controlled multi-body system with the controllers from section 3.3.2. Even the feedback linearization based controller deviates from the trajectories of the SLIP model due to the energy loss at each impact. One can see that the deviation of the simpler control law is larger, but it still results in a quasi-periodic walking motion.

The vertical ground reaction force for the SLIP model and the forces due to the virtual springs for the two controllers are shown in Fig. 3.8 - 3.10, with higher variations in the force for the multi-body model.

The effect of the energy control approach (3.18) is presented in Fig. 3.11 with the feedback linearization based controller and different values of the controller gain K_H. One can see that without the energy controller (i.e. for $K_H = 0$) the multi-body system instantaneously loses energy

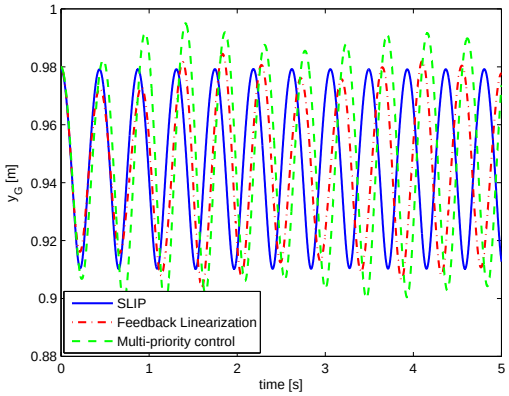

Figure 3.7.: Vertical motion of the CoM for the two multi-body controllers compared with the SLIP dynamics.

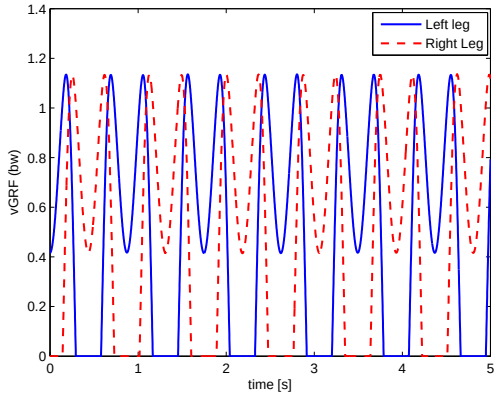

Figure 3.8.: Vertical ground reaction force of the SLIP model.

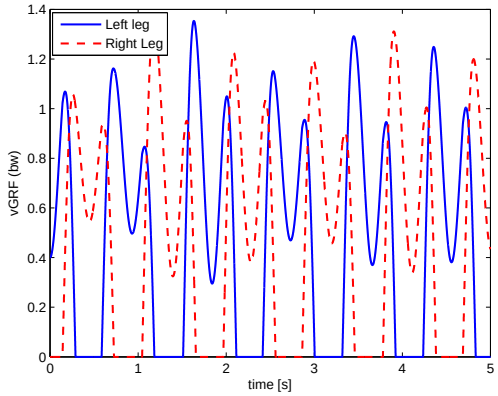

Figure 3.9.: Virtual springs forces of the multi-body system with the feedback linearization based controller.

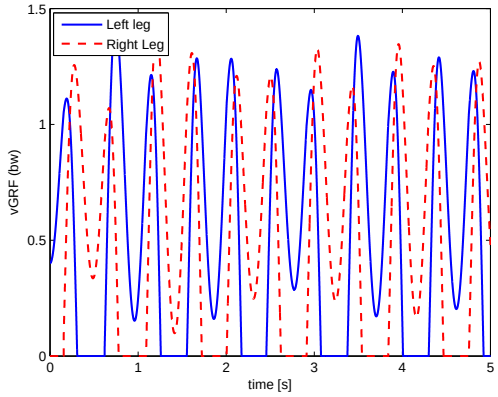

Figure 3.10.: Virtual springs forces of the multi-body system with the simplified control law.

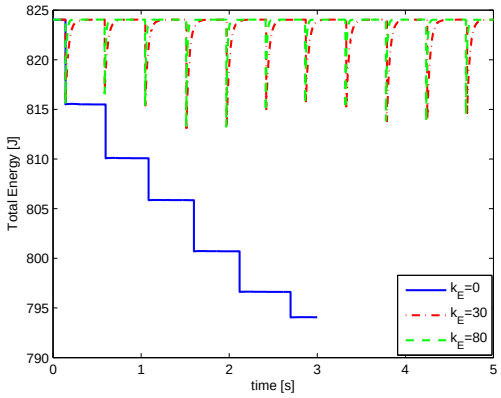

Figure 3.11.: Energy control in the multi-body model using the feedback linearization based controller.

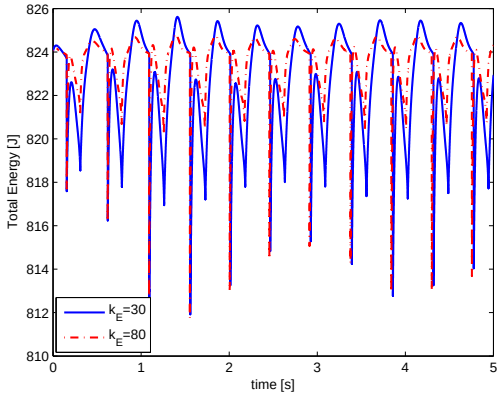

Figure 3.12.: Energy control in the multi-body model using the simplified controller.

at each step due to the impact. The red dash-dotted line and the green dashed line show the total energy for two different controller gains. This evaluation verifies that the proposed energy control method efficiently compensates for the energy loss and the energy is recovered according to the time constant in (3.17). Figure 3.12 shows the results with the simpler control law from section 3.3.2. Even if there is a larger variation in the total energy due to the uncompensated nonlinear terms in the closed loop dynamics, also for this controller the energy loss at the impact is compensated.

As the goal is to reproduce the SLIP behavior, also the stability analysis will be conducted as it is usually done for the latter. In other words, the Poincaré map corresponding to the VLO configuration will be considered for both systems. To be precise, it should be mentioned that a more formally correct way to analyze the system is to consider a three-dimensional Poincaré map. For the multi-body robot, in fact, the total energy is not exactly constant, but only controlled to be constant. Therefore, it should be considered as the third coordinate on the Poincaré section. Nevertheless, if the total energy control is fast enough to ensure that the total energy does not deviate far from its desired value the approximation of considering a constant energy level is reasonable. The behavior of the two controllers on the Poincaré section is shown in Fig. 3.13 and Fig. 3.14. One can see that both controllers converge to a fixed point, which however is different from the original fixed point of the SLIP model, represented with a red circle in the figures.

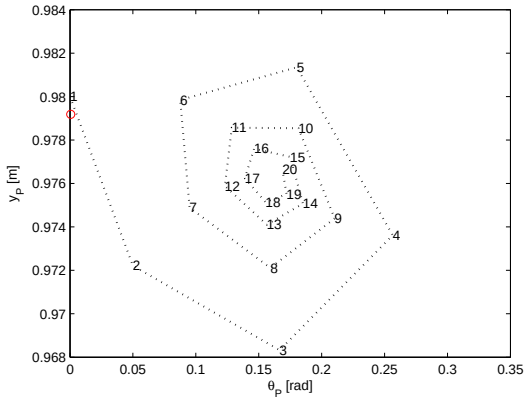

Figure 3.13.: Poincaré section of the multi-body model with the feedback linearization based controller. The red circle is the fixed point for the original SLIP model.

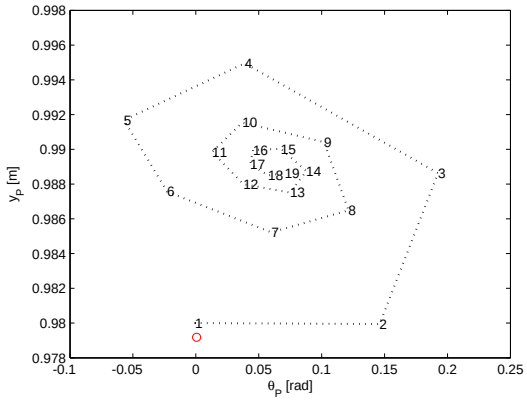

Figure 3.14.: Poincaré section of the multi-body model with the simplified controller. The red circle is the fixed point for the original SLIP model.

3.5. Summary

In this chapter, the bipedal SLIP model was utilized as a template for the control of a multi-body robot. The controller acts on two levels. In order to compensate for the energy loss that is inherent to the impacts at the foot touch down, an energy control algorithm for the SLIP model was proposed. In contrast to previous solutions in the literature, this controller is based on a continuous regulation of the energy and does not need any prediction of the energy loss. The controlled SLIP dynamics is then used as a desired dynamics for a lower level controller designed for the multi-body robot. Two solutions of different complexity were proposed and compared. A multi-body robot controller based on feedback linearization results in a very close emulation of the SLIP behavior. Moreover, a simpler control approach was discussed, which avoids the computational burden and parameter sensitivity problems of the feedback linearization based controller. The proposed controllers were shown in simulations to produce an asymptotically stable limit cycle behavior similar to the SLIP dynamics. Due to the energy loss at the foot touch down and the nonlinearities in the dynamics, the fixed point on the Poincaré section, however, differs from the original fixed point of the SLIP model. Assumptions were made in order to simplify the problem, for example to avoid considering the zero moment point (ZMP) restrictions. Further on in the thesis, these assumptions will be removed.

CHAPTER 4

Energy regulation for rigidly actuated robots

In this chapter the hypothesis of removing the underactuation due to both the contact and the elastic joints is still holding. As a result, the robot prototype to consider is that of a standard robot manipulator with dynamic equation

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{g}(\boldsymbol{q})=\boldsymbol{\tau}\ . \tag{4.1}$$

The contribution of this chapter is the design of a control law for orbital stabilization of rigidly actuated manipulators. The design concept of the controller is sketched in Fig. 4.1 and it is shared with its elastic counterpart presented in chapter 5. It has the controller from section 3.3.1 as a "predecessor" and it is based on the idea of forcing the system to reach a subset of the state space and then producing a limit cycle by regulating an energy function to a desired value. In some sense, the content of this chapter represents a first step towards the design of a controller which can produce a limit cycle taking directly into account the energy stored by the springs of an elastically actuated robot.

For repetitive tasks, it is usually more important to stay on a prescribed orbit in the state space, rather than following the exact position in time along the desired curve. For these applications, tracking a trajectory might not be the best solution and orbital stabilization might be instead considered. The problem of orbital stabilization for rigid joint robots has been addressed, for example,

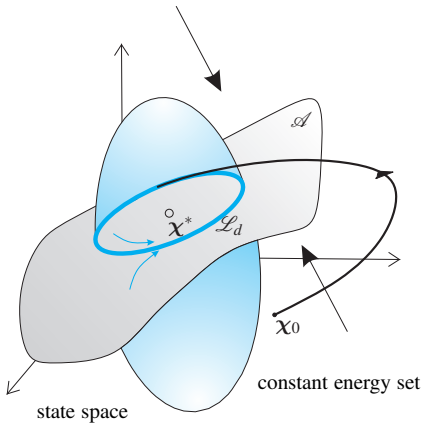

Figure 4.1.: Conceptual illustration of the idea behind the energy based controllers.

in [DS03, DS04, TF09, CEU02]. In [CEU02] the concepts of virtual constraint and feedback linearization are used to obtain a closed loop system that generates its own periodic stable motion, while in [DS03, DS04] a passive control action is designed which allows to decouple the motion along a vector field from the remaining one. The system is then forced to follow an integral curve of this vector field via a passive control law. In case of a closed integral field, the system thus converges to a closed orbit in the configuration space. In [TF09], additionally, a non-passive control action was proposed to achieve regulation of the final velocity along the vector field. Compared to [DS03, DS04, TF09], in the proposed approach the limit cycle is defined by a virtual energy function and the problem is formulated based on the nullspace decomposition introduced in [PCY99] and used for nullspace compliance control in [Ott08, OKN08]. The power-conserving nullspace decoupling from [OKN08] generalizes the nominal control law in [DS03, DS04]. The two can be shown to be equivalent in case of a 1 - dimensional nullspace, however, the approach from [OKN08] can also be applied in case of a higher-dimensional nullspace. Compared to [CEU02] the passivity property of the system is used to avoid altering the original dynamics of the system through feedback linearization. Moreover, the design is split in the generation of the limit cycle and satisfaction of the virtual constraints. These, in turn, are responsible for the subspace in which the system will oscillate.

The stability analysis of the limit cycle relies on semidefinite Lyapunov functions, which can be found in appendix A. Using the conditional stability, the limit cycle can be designed for a lower dimensional dynamical system describing how the original one evolves on a chosen submanifold and the corresponding velocity space. Even if designed assuming a lower dimensional system, the limit cycle is an asymptotically stable solution for the whole system.

4.1. Motivating example

The main idea behind the control laws based on the regulation of an energy function that are proposed in this work can be easily exemplified considering a simple mass-spring system, whose dynamic equation is given by

$$m\ddot{q} + kq = u \,, \tag{4.2}$$

with mass $m > 0$, stiffness constant $k > 0$, state $\chi = (q, \dot{q})$ and input u. Choosing $u = 0$, the system has an equilibrium point in $\chi^* = (0,0)$. This type of equilibrium point is called center [Kha02], since for each initial condition $\chi_0 = (q_0, \dot{q}_0)$ the resulting trajectory will be a closed orbit around the equilibrium point, that is the level set $\mathscr{L}_0$ of the Hamiltonian $H(\chi) = \frac{1}{2}\left(m\dot{q}^2 + kq^2\right)$, defined as $\mathscr{L}_0 = \{\chi \mid H(\chi) = H(\chi_0)\}$.

The difference between these closed orbits and limit cycles is that they are not isolated. Forcing the system to reach always a desired value of the Hamiltonian, will result in the presence of a limit cycle. This is possible by choosing $u = -K_H \dot{q}\tilde{H}(\chi)$, obtaining the closed loop system

$$m\ddot{q} + K_H \dot{q}\tilde{H}(\chi) + kq = 0 \,, \tag{4.3}$$

where $K_H > 0$, $\tilde{H}(\chi) = H(\chi) - H_d$ and $H_d > 0$ is the desired value of the Hamiltonian.

Proof. Firstly, the invariance of the set $\mathscr{L}_d = \{\chi \mid H(\chi) = H_d\}$ is shown. To this end, the derivative of $\tilde{H}(\chi)$ along the flow of the system is computed, resulting in $\dot{\tilde{H}}(\chi) = -K_H \dot{q}^2 \tilde{H}(\chi)$, which proves that $\tilde{H}$ will not change when starting from $\mathscr{L}_d$. As second step, the stability of $\mathscr{L}_d$ is shown considering the Lyapunov function $V(\chi) = \frac{1}{2}\tilde{H}^2(\chi)$, with derivative $\dot{V}(\chi) = \tilde{H}(\chi)\dot{\tilde{H}}(\chi) = -K_H \dot{q}^2 \tilde{H}^2(\chi) \leq 0$. For the asymptotic stability, LaSalle's invariance theorem [Kha02] is applied,

which allows to conclude that the positive limit set Ł$^+$ is given by the largest positive invariant set M within the set $E = \{\boldsymbol{\chi} \mid \dot{q} = 0 \, or \, H(\boldsymbol{\chi}) = H_d\}$, i.e. the set where $\dot{V}(\boldsymbol{\chi}) = 0$. Beside $\mathscr{L}_d$, also $\boldsymbol{\chi}^*$ is an invariant set contained in M. Invoking Chetaev instability theorem [Kha02] using the function $H(\boldsymbol{\chi})$ and considering a neighborhood of the equilibrium point in which $0 < H < H_d$, it can be concluded that $\boldsymbol{\chi}^*$ is an unstable equilibrium point. Therefore every solution, except for the trivial one $\boldsymbol{\chi}(t) = \boldsymbol{\chi}^*$, $\forall t \geq 0$, approaches $\mathscr{L}_d$ as $t \to \infty$. □

For multibody systems, ensuring that the Hamiltonian $H(\boldsymbol{\chi}) \to H_d$ is not enough to conclude the existence of a limit cycle. The difference is that for a 1 - DoF system the set $\mathscr{L}_d$ is an isolated closed[1] orbit in the state space (corresponding to a limit cycle), which is not true for a n - DoF system. For this reason, in the following the system will be forced to evolve on a 1 - dimensional submanifold of the configuration space.

4.2. Coordinate transformation

Since the goal is to force the system to evolve on a submanifold of the configuaration space, a function $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{q})$ is used to describe the submanifold. In fact, assuming that the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{(n-1)\times n}$ of the mapping $\boldsymbol{y} : \mathbb{R}^n \to \mathbb{R}^{n-1}$ is a full rank matrix, then $\boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}$ defines a 1 - dimensional submanifold of the configuration space [Fra04]. Accordingly, the dynamics of the system can be written with a new set of coordinates, as in [PCY99, Ott08, OKN08]. First a nullspace base matrix[2] $\boldsymbol{Z}(\boldsymbol{q}) \in \mathbb{R}^{1\times n}$ is computed, allowing to obtain the directions orthogonal to the submanifold, then $\boldsymbol{Z}(\boldsymbol{q})$ is used to compute a dynamically consistent[3] nullspace projector $\boldsymbol{N}(\boldsymbol{q}) \in \mathbb{R}^{1\times n}$, which will be part of the extended Jacobian matrix $\boldsymbol{J}_N(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$, such that

$$\begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} = \boldsymbol{J}_N(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{q}) \\ \boldsymbol{N}(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} \, , \tag{4.4}$$

where $\boldsymbol{N}(\boldsymbol{q}) = \Big(\boldsymbol{Z}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{Z}^T(\boldsymbol{q})\Big)^{-1}\boldsymbol{Z}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})$ and v is an additional nullspace velocity. With this choice, the extended Jacobian matrix $\boldsymbol{J}_N(\boldsymbol{q})$ is non singular and the inverse is given by

$$\boldsymbol{J}_N^{-1}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}^{+_M}(\boldsymbol{q}) & \boldsymbol{Z}^T(\boldsymbol{q}) \end{bmatrix} \, , \tag{4.5}$$

where $\boldsymbol{J}^{+_M}(\boldsymbol{q})$ denotes the dynamically consistent weighted pseudo inverse defined as

$$\boldsymbol{J}^{+_M}(\boldsymbol{q}) = \boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\Big(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\Big)^{-1} \, . \tag{4.6}$$

The joint velocity can thus be computed from the Cartesian velocity and the nullspace velocity via

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M}(\boldsymbol{q})\dot{\boldsymbol{y}} + \boldsymbol{Z}^T(\boldsymbol{q})v \, . \tag{4.7}$$

From (4.4) and (4.7) it is straightforward to rewrite (4.1) in the extended velocity coordinates as

$$\boldsymbol{\Lambda}(\boldsymbol{q}) \begin{bmatrix} \ddot{\boldsymbol{y}} \\ \dot{v} \end{bmatrix} + \boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} = \boldsymbol{J}_N^{-T}(\boldsymbol{q})\Big(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q})\Big) \, , \tag{4.8}$$

[1] It is the level set of a positive definite and radially unbounded function.
[2] I.e. it fulfills the condition $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{Z}^T(\boldsymbol{q}) = \boldsymbol{0}$.
[3] I.e. it fulfills the condition $\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{N}^T(\boldsymbol{q}) = \boldsymbol{0}$.

where the matrix $\boldsymbol{\Lambda}(\boldsymbol{q})$ is block diagonal, due to the use of a dynamically consistent nullspace projector matrix, whose expression is

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{\Lambda}_y(\boldsymbol{q}) & \mathbf{0} \\ \mathbf{0} & \Lambda_n(\boldsymbol{q}) \end{bmatrix}$$

$$\boldsymbol{\Lambda}_y(\boldsymbol{q}) = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\right)^{-1} \qquad \Lambda_n(\boldsymbol{q}) = \boldsymbol{Z}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{Z}^T(\boldsymbol{q})$$

while for $\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ (omitting the dependencies)

$$\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{\Gamma}_y(\boldsymbol{q}, \dot{\boldsymbol{q}}) & \boldsymbol{\Gamma}_{yn}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{\Gamma}_{ny}(\boldsymbol{q}, \dot{\boldsymbol{q}}) & \Gamma_n(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix}$$

$$\boldsymbol{\Gamma}_y = \boldsymbol{\Lambda}_y\left(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}\right)\boldsymbol{J}^{+_M} \qquad \boldsymbol{\Gamma}_{yn} = \boldsymbol{\Lambda}_y\left(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}\right)\boldsymbol{Z}^T$$

$$\boldsymbol{\Gamma}_{ny} = -\boldsymbol{\Gamma}_{yn}^T \qquad \Gamma_n = \Lambda_n\left(\boldsymbol{N}\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{N}}\right)\boldsymbol{Z}^T \ .$$

The control input $\boldsymbol{\tau}$ can be split accordingly in the components $\boldsymbol{\tau}_y$ and τ_n along the two spaces

$$\boldsymbol{\tau} = \boldsymbol{J}_N^T(\boldsymbol{q}) \begin{bmatrix} \boldsymbol{\tau}_y \\ \tau_n \end{bmatrix} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\tau}_y + \boldsymbol{N}^T(\boldsymbol{q})\tau_n \ , \tag{4.9}$$

so that (omitting the dependencies) the system can be written in the form

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M}\dot{\boldsymbol{y}} + \boldsymbol{Z}^T v \tag{4.10}$$

$$\ddot{\boldsymbol{y}} = \boldsymbol{\Lambda}_y^{-1}\left(-\boldsymbol{\Gamma}_y\dot{\boldsymbol{y}} - \boldsymbol{\Gamma}_{yn}v - \boldsymbol{J}^{+_M T}\boldsymbol{g} + \boldsymbol{\tau}_y\right) \tag{4.11}$$

$$\dot{v} = \Lambda_n^{-1}\left(\boldsymbol{\Gamma}_{yn}^T\dot{\boldsymbol{y}} - \Gamma_n v - \boldsymbol{Z}\boldsymbol{g} + \tau_n\right) \tag{4.12}$$

4.3. Control law

Loosely speaking, the control input $\boldsymbol{\tau}$ will be chosen such that $n-1$ constraints are satisfied and, at the same time, a limit cycle for the "remaining dynamics" is generated, i.e. designing the input $\boldsymbol{\tau}_y$ to keep the system on the submanifold and the corresponding velocity space and the input τ_n to produce the oscillation on the submanifold itself. In addition, a power conserving term is included to shift energy from the constraint space into the nullspace. The idea of having such an energy shifting term is used also in the Port - Hamiltonian framework between subsystems [Str15].

4.3.1. Design

Since no elastic elements are present in the case under consideration a virtual potential $U(\boldsymbol{q})$ is introduced through the controller, which will clearly have an influence on the shape of the limit cycle, as shown in section 4.4. The desired value H_d also influences the shape of the limit cycle. As a rule of thumb, one can consider that when the regulation goal is achieved and the kinetic energy is zero, the desired energy is equal to the virtual potential energy and therefore it directly influences the maximum amplitude of the oscillations.

For a given desired configuration $\boldsymbol{q}_d \in \mathbb{R}^n$, such that $\boldsymbol{y}(\boldsymbol{q}_d) = \mathbf{0}$, let $U(\boldsymbol{q})$ be a positive definite C^1 function on the 1 - dimensional submanifold defined by $\boldsymbol{y}(\boldsymbol{q}) = \mathbf{0}$ and having its minimum at $\boldsymbol{q}_d$. In other words, $\boldsymbol{q}_d$, $U(\boldsymbol{q})$ and $\boldsymbol{y}(\boldsymbol{q})$ are chosen such that $U(\boldsymbol{q})$ has a unique constrained local minimum at $\boldsymbol{q}_d$, i.e.

- $\boldsymbol{y}(\boldsymbol{q}_d) = \mathbf{0}$

- $U(\boldsymbol{q}) \geq 0, \forall \boldsymbol{q} : \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}$
- $U(\boldsymbol{q}) = 0 \Longleftrightarrow \boldsymbol{q} = \boldsymbol{q}_d, \forall \boldsymbol{q} : \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}$.

Consequently, the configuration $\boldsymbol{q}_d$ can be obtained as

$$\begin{aligned} \boldsymbol{q}_d = \arg\min_{\boldsymbol{q}} \; & U(\boldsymbol{q}) \\ \text{s.t. } & \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0} \; . \end{aligned} \tag{4.13}$$

The necessary condition for optimality of the minimization problem, obtained using the Lagrange multipliers $\boldsymbol{\lambda}$, is

$$\nabla_{\boldsymbol{q}} U(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\lambda} = 0 \; , \tag{4.14}$$

so that pre-multiplying by the nullspace base matrix $\boldsymbol{Z}(\boldsymbol{q})$, the condition

$$n(\boldsymbol{q}) := \boldsymbol{Z}(\boldsymbol{q})\nabla_{\boldsymbol{q}} U(\boldsymbol{q}) = 0 \tag{4.15}$$

is obtained, where $n(\boldsymbol{q})$ can be seen as a local nullspace coordinate [Bai85]. Concluding, $\boldsymbol{q}_d$ is the unique configuration with coordinates $\boldsymbol{y}(\boldsymbol{q}_d) = \boldsymbol{0}$, $n(\boldsymbol{q}_d) = 0$.

At this point, similarly to section 4.1 where $\frac{1}{2}kq^2$ had the role of $U(\boldsymbol{q})$, the "energy - like" function

$$H(\boldsymbol{\chi}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + U(\boldsymbol{q}) \; , \tag{4.16}$$

is defined, which consists of the physical kinetic energy and the virtual potential energy. Defining $\tilde{H} = H - H_d$, the goal is to regulate H to the desired value H_d, i.e. $\tilde{H} \to 0$ as $t \to \infty$. Before that, the system has to be forced to evolve on the submanifold. This can be achieved by choosing

$$\boldsymbol{\tau}_y = \boldsymbol{J}^{+_M T}\boldsymbol{g} + \boldsymbol{\Gamma}_{yn} v - \boldsymbol{D}_y \dot{\boldsymbol{y}} - \boldsymbol{K}_y \boldsymbol{y} \; , \tag{4.17}$$

since an opportune choice of $\boldsymbol{K}_y, \boldsymbol{D}_y$ guarantees that the closed loop dynamics

$$\boldsymbol{\Lambda}_y \ddot{\boldsymbol{y}} + \Big(\boldsymbol{\Gamma}_y + \boldsymbol{D}_y\Big)\dot{\boldsymbol{y}} + \boldsymbol{K}_y \boldsymbol{y} = \boldsymbol{0} \tag{4.18}$$

converges and stays on the submanifold. Computing the derivative of $H(\boldsymbol{\chi})$ along the flow of the system, when it is evolving on the submanifold, leads to

$$\dot{H} = v\Big(- \boldsymbol{Z}(\boldsymbol{q})\boldsymbol{g}(\boldsymbol{q}) + \tau_n + \boldsymbol{Z}(\boldsymbol{q})\nabla_{\boldsymbol{q}} U(\boldsymbol{q})\Big) \; . \tag{4.19}$$

This motivates the choice $\tau_n = \boldsymbol{Z}\boldsymbol{g} - K_H \tilde{H} v - \boldsymbol{Z}\nabla_{\boldsymbol{q}} U$ in order to achieve a similar dynamics for the energy as in section 4.1.

A precise formulation of the controller and its stability analysis is provided in the next section.

4.3.2. Stability analysis

Given the system (4.1) with input torque $\boldsymbol{\tau}$, let $K_H, K_\alpha \in \mathbb{R}$ be positive scalars and $\boldsymbol{K}_y, \boldsymbol{D}_y \in \mathbb{R}^{(n-1)\times(n-1)}$ two symmetric and positive definite matrices, then the proposed nonlinear static state feedback controller (omitting the dependencies)

$$\boldsymbol{\tau} = \boldsymbol{g} + \boldsymbol{J}_N^T \left(\begin{bmatrix} -\boldsymbol{D}_y & \boldsymbol{\Gamma}_{yn} - K_\alpha \dot{\boldsymbol{y}} v \\ -\boldsymbol{\Gamma}_{yn}^T + K_\alpha v \dot{\boldsymbol{y}}^T & -K_H \tilde{H} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_y \boldsymbol{y} \\ \boldsymbol{Z}\nabla_{\boldsymbol{q}} U \end{bmatrix} \right) \; , \tag{4.20}$$

leads to the closed loop system:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M}\dot{\boldsymbol{y}} + \boldsymbol{Z}^T v \tag{4.21a}$$

$$\boldsymbol{\Lambda}_y\ddot{\boldsymbol{y}} + \Big(\boldsymbol{\Gamma}_y + \boldsymbol{D}_y + K_\alpha v^2 \boldsymbol{E}\Big)\dot{\boldsymbol{y}} + \boldsymbol{K}_y\boldsymbol{y} = \boldsymbol{0} \tag{4.21b}$$

$$\Lambda_n\dot{v} + \Big(\Gamma_n + K_H\tilde{H} - K_\alpha\dot{\boldsymbol{y}}^T\dot{\boldsymbol{y}}\Big)v + \boldsymbol{Z}\nabla_{\boldsymbol{q}}U = \boldsymbol{0} \tag{4.21c}$$

with state vector $\boldsymbol{\chi} = (\boldsymbol{q}, \dot{\boldsymbol{y}}, v)$ and where $\boldsymbol{E} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the identity matrix.

Theorem 4.1 (Rigid case)**.** *The nonlinear autonomous system* (4.21)*, depending on H_d, has an asymptotically stable solution consisting of*

(a) the equilibrium point $\boldsymbol{\chi}^* = (\boldsymbol{q}_d, \boldsymbol{0}, 0)$, for $H_d = 0$

(b) the limit cycle defined by $\mathscr{L}_d = \big\{\boldsymbol{\chi} \mid \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0},\ \tilde{H}(\boldsymbol{\chi}) = 0\big\}$, for $H_d > 0$.

Proof. The proof is based on the theorems and lemmas collected in appendix A.

Consider the C^1 function of the state

$$V_y(\boldsymbol{\chi}) = \frac{1}{2}\left(\dot{\boldsymbol{y}}^T\boldsymbol{\Lambda}_y(\boldsymbol{q})\dot{\boldsymbol{y}} + \boldsymbol{y}^T(\boldsymbol{q})\boldsymbol{K}_y\boldsymbol{y}(\boldsymbol{q})\right)\ , \tag{4.22}$$

such that

$$\begin{cases} V_y(\boldsymbol{\chi}) = 0 & \forall \boldsymbol{\chi} \in \mathscr{A} \\ V_y(\boldsymbol{\chi}) > 0 & \forall \boldsymbol{\chi} \notin \mathscr{A}\ , \end{cases}$$

where $\mathscr{A} = \{\boldsymbol{\chi} \mid \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}\}$. Since $\mathscr{L}_d \subset \mathscr{A}$, $V_y(\boldsymbol{\chi})$ is a candidate semidefinite Lyapunov function to prove the stability of $\mathscr{L}_d$ for the system (4.21). Computing its derivative along the flow of the system, leads to $\dot{V}_y(\boldsymbol{\chi}) = -\dot{\boldsymbol{y}}^T\Big(\boldsymbol{D}_y + K_\alpha v^2\boldsymbol{E}\Big)\dot{\boldsymbol{y}} \leq 0$. The set $\mathscr{A}$ is the largest positively invariant set within $E_y = \{\boldsymbol{\chi} \mid \dot{\boldsymbol{y}} = \boldsymbol{0}\}$, i.e. the set where $\dot{V}_y(\boldsymbol{\chi}) = 0$. In fact, from (4.21b) it is clear that if $\boldsymbol{y}(\boldsymbol{q}) \neq \boldsymbol{0}$ then the system leaves E_y.

If $\mathscr{L}_d$ can be proven to be asymptotically stable conditionally to $\mathscr{A}$, then all the requirements of Theorem A.3 are satisfied and $\mathscr{L}_d$ will turn out to be asymptotically stable for the whole system (4.21). The set $\mathscr{L}_d$ is invariant for (4.21), since it is conditionally invariant to $\mathscr{A}$, which is itself an invariant set. To verify that, consider the system

$$\dot{\boldsymbol{q}} = \boldsymbol{Z}^T v \tag{4.23a}$$

$$\Lambda_n\dot{v} + \Big(\Gamma_n + K_H\tilde{H}\Big)v + \boldsymbol{Z}\nabla_{\boldsymbol{q}}U = 0\ , \tag{4.23b}$$

with $\boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}$, i.e. (4.21) conditionally to $\mathscr{A}$. Remembering that $n(\boldsymbol{q}_d) = 0$, the invariance is clear for $\boldsymbol{\chi}^*$, while for $\mathscr{L}_d$ the derivative of $\tilde{H}(\boldsymbol{\chi})$ along the flow of the system needs to be computed. The latter results in $\dot{\tilde{H}}(\boldsymbol{\chi}) = -K_H v^2\tilde{H}(\boldsymbol{\chi})$ and proves that $\tilde{H}$ will not change when starting from $\mathscr{L}_d$. As in Section 4.1, the Lyapunov function $V_n(\boldsymbol{\chi}) = \frac{1}{2}\tilde{H}^2(\boldsymbol{\chi})$, with derivative $\dot{V}_n(\boldsymbol{\chi}) = \tilde{H}(\boldsymbol{\chi})\dot{\tilde{H}}(\boldsymbol{\chi}) = -K_H v^2\tilde{H}^2(\boldsymbol{\chi}) \leq 0$, shows the stability of $\mathscr{L}_d$ for (4.23). For the asymptotic stability, LaSalle's invariance theorem allows to conclude that the positive limit set $Ł_n^+$ is given by the largest positive invariant set M_n within the set $E_n = \{\boldsymbol{\chi} \mid v = 0\, or\, H(\boldsymbol{\chi}) = H_d\}$, i.e. the set where $\dot{V}_n(\boldsymbol{\chi}) = 0$. Beside $\mathscr{L}_d$, also $\boldsymbol{\chi}^*$ is an invariant set contained in M_n, but, given the choice made for $U(\boldsymbol{q})$, $H(\boldsymbol{\chi})$ is positive definite conditionally to $\mathscr{A}$. Therefore, in case $H_d = 0$, it can be easily verified that $\mathscr{L}_d$ coincides with $\boldsymbol{\chi}^*$, which becomes the only positive limit set and

therefore result (a) is obtained. On the other hand, when $H_d > 0$, invoking Chetaev instability theorem [Kha02] using the function $H(\boldsymbol{\chi})$ and considering a neighborhood of the equilibrium point in which $0 < H < H_d$, it can be concluded that $\boldsymbol{\chi}^*$ is an unstable equilibrium point. Therefore every solution, except for the trivial one $\boldsymbol{\chi}(t) = \boldsymbol{\chi}^*$, $\forall t \geq 0$, approaches $\mathscr{L}_d$ as $t \to \infty$. The set $\mathscr{L}_d$ is uniquely determined by one parameter, e.g. the value of $n(\boldsymbol{q})$, and therefore it is a closed[4] orbit in the state space, i.e. a limit cycle, proving result (b). □

Namely, decreasing the value of H_d, the asymptotically stable limit cycle collapses on the unstable equilibrium point, originating an asymptotically stable equilibrium point.

4.3.3. Controller discussion

The torque generated by the controller can be rewritten as

$$\boldsymbol{\tau} = \boldsymbol{g} + \boldsymbol{J}_N^T \left(\boldsymbol{F}_\Gamma + \boldsymbol{F}_\alpha \right) + \boldsymbol{J}^T \boldsymbol{F}_y + \boldsymbol{N}^T \left(\boldsymbol{F}_U + \boldsymbol{F}_H \right), \tag{4.24}$$

where

$$\boldsymbol{F}_\Gamma = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Gamma}_{yn} \\ -\boldsymbol{\Gamma}_{yn}^T & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} \qquad \boldsymbol{F}_\alpha = \begin{bmatrix} \boldsymbol{0} & -K_\alpha \dot{\boldsymbol{y}} v \\ K_\alpha v \dot{\boldsymbol{y}}^T & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} \tag{4.25}$$

$$\boldsymbol{F}_y = -\boldsymbol{D}_y \dot{\boldsymbol{y}} - \boldsymbol{K}_y \boldsymbol{y} \tag{4.26}$$

$$\boldsymbol{F}_U = \boldsymbol{Z} \nabla_{\boldsymbol{q}} U \qquad \boldsymbol{F}_H = -K_H \tilde{H} v \quad . \tag{4.27}$$

As shown in Fig. 4.2, three contribution can be distinguished: a dynamic reshaping of the system, the torque responsible for forcing the system to evolve on a 1 - dimensional submanifold of the configuration space and finally the one creating the asymptotically stable limit cycle. The last two are dynamically decoupled between them, while the dynamic reshaping part of the controller compensates for the gravity torque and the coupling terms in the Coriolis matrix and shifts energy from the constraint space to the nullspace. It is worth to notice that the entry $-\boldsymbol{\Gamma}_{yn}^T + K_\alpha v \dot{\boldsymbol{y}}^T$ is not necessary for the stability analysis, since it has no influence when the system has reached the set $\mathscr{A}$. Nevertheless, in this way, a power conserving term[5] is obtained, i.e. $\dot{\boldsymbol{q}}^T \boldsymbol{J}_N^T \left(\boldsymbol{F}_\Gamma + \boldsymbol{F}_\alpha \right) = 0$. Although $\boldsymbol{F}_\Gamma$ compensates for the coupling terms in the Coriolis matrix, it does not decouple the dynamics in the two spaces. The remaining blocks of the Coriolis matrix are, in fact, still function of the whole state. Finally, through K_α the rate at which the energy $K_\alpha v^2 \dot{\boldsymbol{y}}^T \dot{\boldsymbol{y}}$ flows from the constraint space to the nullspace can be controlled.

The problem of forcing the system to evolve along the constraint submanifold and the corresponding velocity space can be reformulated as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\tau}_y + \boldsymbol{N}^T(\boldsymbol{q})\tau_n \tag{4.28}$$

$$\text{s.t.} \quad \boldsymbol{\Lambda}_y(\boldsymbol{q})\ddot{\boldsymbol{y}} + \boldsymbol{\Gamma}_y(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{y}} + \boldsymbol{D}_y\dot{\boldsymbol{y}} + \boldsymbol{K}_y\boldsymbol{y} = \boldsymbol{0} \tag{4.29}$$

While substituting the dynamic equation (4.28) in the constraint (4.29) provides the input $\boldsymbol{\tau}_y$ needed to keep the system on the constraint submanifold and the corresponding velocity space, as in (4.17), premultiplying (4.28) by $\boldsymbol{Z}(\boldsymbol{q})$ and using $\dot{\boldsymbol{q}} = \boldsymbol{Z}^T(\boldsymbol{q})v$ gives the dynamic of the system when evolving along them[6], as in (4.23). Since $\boldsymbol{Z}(\boldsymbol{q})$ has been used instead of the usual nullspace projector directly a minimum set of equation describing the dynamics in the nullspace

[4] It is the level set of a positive definite and radially unbounded function.

[5] This is a generalization of the nominal control in [DS03, DS04].

[6] It is possible to show that $\Lambda_n \left(\boldsymbol{N}\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{N}} \right) \boldsymbol{Z}^T = \boldsymbol{Z} \left(\boldsymbol{C}\boldsymbol{Z}^T + \boldsymbol{M}\dot{\boldsymbol{Z}}^T \right)$

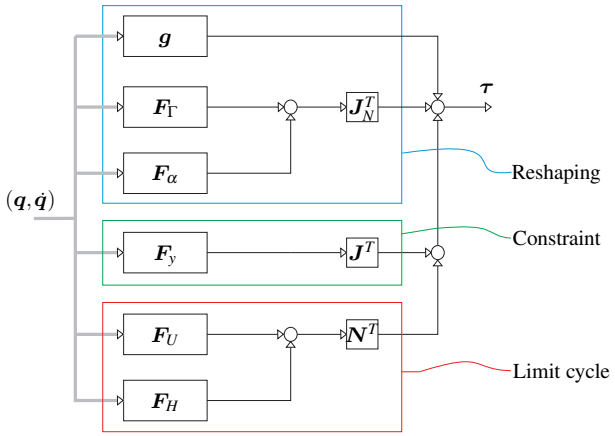

Figure 4.2.: Control law for rigidly actuated robots.

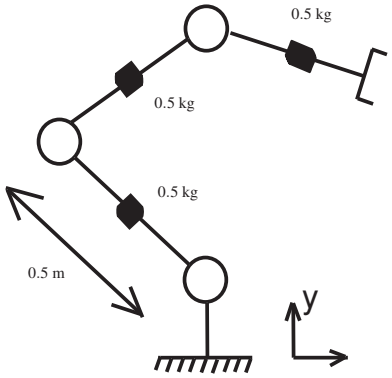

Figure 4.3.: Simulation model in the start configuration.

are obtained, i.e. one equation instead of n in this specific case. In other words, requiring to have dynamic matrices in (4.29) derived from the ones in (4.28) results in the proposed control law for the constraints. On the other hand, if in (4.29) a constant inertia matrix is chosen and accordingly $\mathbf{\Gamma}_y = \mathbf{0}$, then the feedback linearization approach is obtained [CEU02]. The result is not surprising at all since, in this way, the dynamics that describes how the system is forced on the constraint submanifold and the corresponding velocity space is completely altered.

4.4. Simulation

In this section, a simulated 3-DOF planar robot, shown in Fig. 5.14 in the starting configuration, is used to validate the results. The computation of all the components for the dynamics and kinematics was performed using the algorithm and formulas from chapter 2 in MATLAB/Simulink ®.

Two types of simulations have been performed. In the first, the problem of generating the limit cycle in the joint space was analyzed, while in the second the Cartesian space was considered.

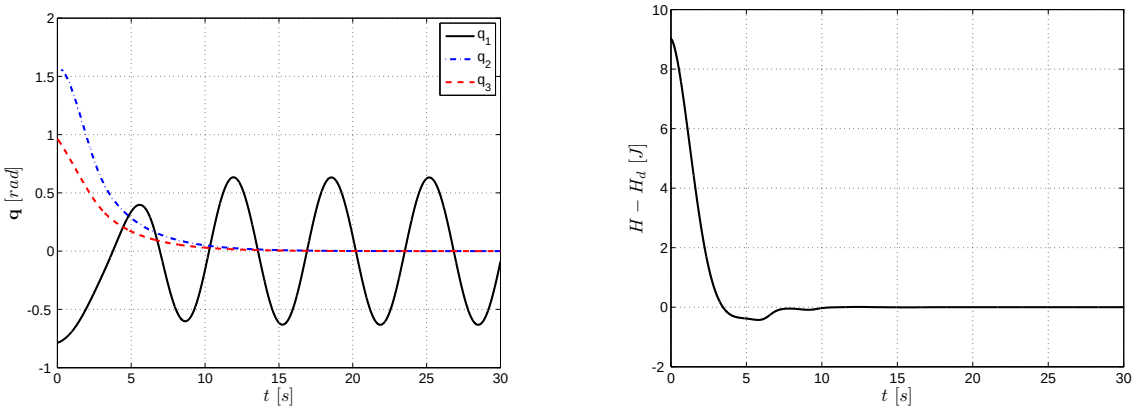

Figure 4.4.: While the last two joints reach the desired values, the first one will oscillate around the equilibrium position. The energy error converges to zero as well.

4.4.1. Limit cycle in joint space

The goal is to make the arm oscillate around the axis of the first joint in a completely stretched configuration. To this end, the constraint submanifold is defined as

$$\begin{bmatrix} q_2 - q_{d2} \\ q_3 - q_{d3} \end{bmatrix} = \mathbf{0} \,, \tag{4.30}$$

where $q_{d2} = q_{d3} = 0$ correspond to the strecthed configuration. The virtual potential function is chosen as in [OKN08], i.e.

$$U(\boldsymbol{q}) = \frac{1}{2} k_n \tilde{\boldsymbol{q}}^T \tilde{\boldsymbol{q}} \,, \tag{4.31}$$

since it satisfies the requirement of having a unique constrained local minimum at $\boldsymbol{q}_d$, if $q_{d1} = 0$ is chosen. In Fig. 4.4 it is possible to check that the system reaches the constraint submanifold and at the same time oscillates as prescribed, with $H \to H_d$. Table 4.1 provides the values of the parameters used in the simulation.

Table 4.1.: Desired energy value and gains used for the simulation results in Fig. 4.4. When only the i - th entry is shown, then the others have the same value.

H_d	K_{y_i}	D_{y_i}	K_α	K_H	k_n
1 J	0.25 Nm/rad	0.75 Nms/rad	0	8 s/rad	5 Nm/rad

4.4.2. Limit cycle in Cartesian space

In this case, the robot is required to produce an oscillation along the vertical direction in the Cartesian space while keeping the horizontal one and the orientation constant. For this case the submanifold is chosen as

$$\begin{bmatrix} y_1(\boldsymbol{q}) - y_{d1} \\ y_3(\boldsymbol{q}) - y_{d3} \end{bmatrix} = \mathbf{0} \,, \tag{4.32}$$

where $y_1(\boldsymbol{q})$ is the horizontal position of the end effector, $y_2(\boldsymbol{q})$ the vertical, $y_3(\boldsymbol{q})$ the orientation and the desired values are $y_{d1} = y_{d2} = 0.85\,\text{m}$, $y_{d3} = \frac{\pi}{2}\,\text{rad}$.

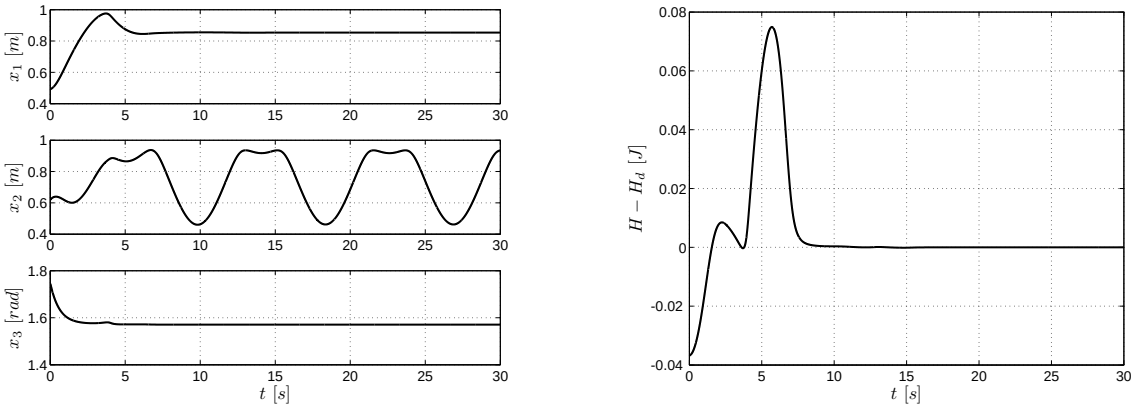

Figure 4.5.: End effector coordinates, with $U(\boldsymbol{q}) = 0.05\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{q}}$.

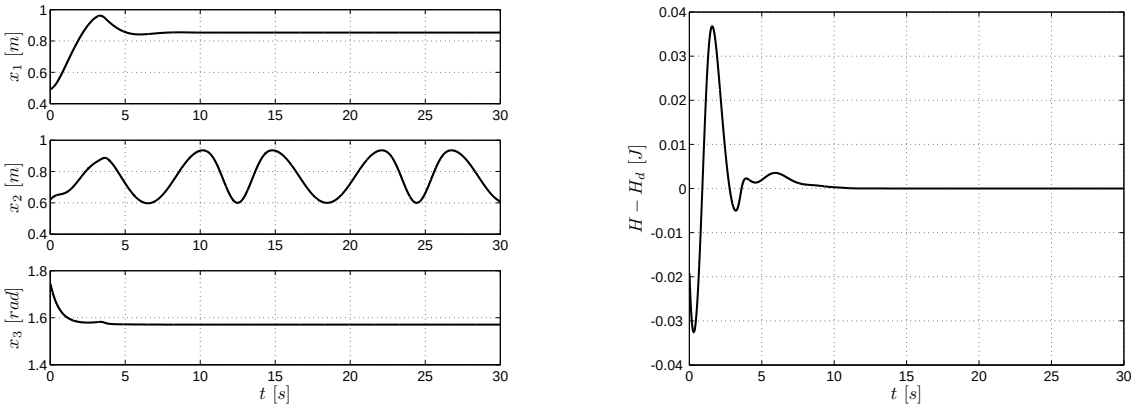

Figure 4.6.: End effector coordinates, with $U(\boldsymbol{q}) = 2.5\tilde{y}_2^2(\boldsymbol{q})$.

In order to show the effect of a different choice for the potential function in the energy controller, first the one in (4.31) is used, where $\boldsymbol{q}_d$ is computed from $\boldsymbol{y}_d$ solving the inverse kinematics problem, and then

$$U(\boldsymbol{q}) = \frac{1}{2}k_n\tilde{y}_2^2(\boldsymbol{q}) \ , \tag{4.33}$$

where $\tilde{y}_2(\boldsymbol{q}) = y_2 - y_{d2}$. The results are shown in Fig. 4.5 and in Fig. 4.6, respectively. Table 4.2 provides the values of the common controller parameters used in the simulation. Obviously, changing the potential function $U(\boldsymbol{q})$ the shape of the limit cycle is changed.

Table 4.2.: Desired energy value and gains used for the simulation results in Fig. 4.5 - 4.6.

H_d	K_{y_1}	D_{y_1}	K_{y_2}	D_{y_2}	K_α	K_H
0.1 J	4 N/m	3 Ns/m	4 Nm/rad	3 Nms/rad	0	8 s/m^2

Finally, the role of the energy shifting term has not been analyzed ($K_\alpha = 0$) in this simulation study since it will be considered already in the next chapter, where the extension of the controller to the flexible case will be presented.

4.5. Summary

In this chapter, the problem of generating asymptotically stable limit cycles for rigidly actuated manipulators was addressed. To this end, it was first shown that for a 1-DoF system this can be realized using a velocity dependent term proportional to the Hamiltonian of the system. The result was then generalized to the multi-body case by forcing the system to evolve on a submanifold of the configuration space. Usually, feedback linearization is used to enforce the virtual constraints, while here a passivity based approach is proposed. Therefore, the dynamics of the system are not completely altered by the control action. Finally, although the limit cycle was designed for this reduced system (once again by energy regulation), it was proven to be an asymptotically stable solution of the whole system. The energy based approach is crucial to exploit the benefits of compliant actuator designs, as it will be shown in the next chapter.

CHAPTER 5

Energy regulation for elastically actuated robots

In this chapter the first simplifying hypothesis is removed. The considered robot has elastic joints, but a fixed base. The idea presented in the previous chapter is extended here in order to deal with elastically actuated robots. Therefore, the contribution of this chapter is the design of a dynamic state feedback to produce asymptotically stable limit cycles for an elastically actuated robot. Following the same steps taken for the rigid case, first a single joint example is considered and then the extension to a robotic manipulator. While, traditionally, the inevitable elasticity of the joint has been seen as a disturbance to minimize both in the mechanical and control design phases, here the goal is to exploit it to perform highly dynamical, explosive and cyclic motions. Although appealing for their capability of storing energy, the mechanical design of such systems possibly introduces additional nonlinearities in the system dynamics which, along with the underactuation problem, make the controller design more challenging. This results in a more complex control law. While for the single joint case a static state feedback still guarantees the regulation of the energy and the existence of an asymptotically stable limit cycle, in order to achieve the same result for an elastically actuated manipulator, a dynamic state feedback is used instead, i.e. the designed controller is itself a dynamical system.

The method presented here solves the problem of orbital stabilization for elastic joint robots without strong simplifying hypothesis, such as single perturbation hypothesis which overcome the problem using one of the motor side states as a virtual input [GPB$^+$11]. In the single joint case, not even computation of higher order derivatives [De 88, NT88] or filtering and state estimation is required, although in the more general case the higher order derivatives are necessary to guarantee that the system reaches the prescribed submanifold. The control laws presented in the chapter can be used also to regulate the robot to a desired configuration, independently on the number of joints. Compared to classic results for elastic joint robots [ASOP12, Ott08] also a scaling of the torque produced by the elastic elements is included.

All the assumption collected in chapter 2 are assumed to hold, so that the reduced model can be used to describe the system, i.e.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}(\boldsymbol{\phi}) \tag{5.1a}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau}(\boldsymbol{\phi}) = \boldsymbol{\tau}_m \,. \tag{5.1b}$$

Finally, also in this chapter, with an abuse of notation, the state of the system will be always indicated with $\boldsymbol{\chi}$ and with $H(\boldsymbol{\chi})$ the "energy-like" function which will be regulated to the desired

value H_d. The same is valid also for the sets $\mathscr{L}_d$, E and the function $V(\chi)$.

5.1. The energy function

Since the goal is to regulate the energy in order to produce a limit cycle, obviously the definition of such an energy function plays a central role in the control design. To exploit the presence of the spring, the potential energy that they store clearly has to be used in the energy function to regulate. The simple substitution of the virtual potential used in chapter 4 with the physical one, will imply that the minimum of the energy function will be attained when the springs are not deflected. This condition is limiting and impossible to achieve at the equilibrium since there will be an inevitable deflection of the springs due to the gravity. Considering the gravity as well in the definition of the energy is indeed the right way to pursue.

In [Ott08] the problem of constructing an online gravity compensation for an elastic joint robot based on the motor position was solved using a static mapping between motor and link positions. The basic idea is to find, for any motor position, the link position which allows the elastic elements in the joints to compensate for the gravity term. Remembering from chapter 2 that $U_g(\boldsymbol{q})$ and $U_k(\boldsymbol{\theta}-\boldsymbol{q})$ are the gravitational and elastic potential respectively, with $U_k(\boldsymbol{\theta}-\boldsymbol{q})$ satisfying assumption 2.4, the total potential function is $U(\boldsymbol{\theta},\boldsymbol{q}) = U_g(\boldsymbol{q}) + U_k(\boldsymbol{\theta}-\boldsymbol{q})$. The latter allows to consider the function $\bar{\boldsymbol{q}} : \mathbb{R}^n \to \mathbb{R}^n$ defined by the implicit equation

$$\nabla_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q})\big|_{(\boldsymbol{\theta},\bar{\boldsymbol{q}}(\boldsymbol{\theta}))} = \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) - \boldsymbol{\tau}(\boldsymbol{\theta}-\bar{\boldsymbol{q}}(\boldsymbol{\theta})) = \boldsymbol{0}\,, \tag{5.2}$$

$\forall \boldsymbol{\theta} \in \mathbb{R}^n$, which exists, is unique and a diffeomorphism [ASOP12]. Therefore, the desired equilibrium configuration of the robot can be indifferently given in terms of motor or link position, since the two are uniquely related through the function $\bar{\boldsymbol{q}}$.

The function $\bar{\boldsymbol{q}}$ not only can be used to construct a gravity compensation term [Ott08], but also the following "energy-like" function

$$H(\boldsymbol{\theta},\boldsymbol{q},\dot{\boldsymbol{\theta}},\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + U(\boldsymbol{\theta},\boldsymbol{q}) - U(\boldsymbol{\theta},\bar{\boldsymbol{q}}(\boldsymbol{\theta}))\,, \tag{5.3}$$

which is a positive semidefinite function of the state and in particular

$$H(\boldsymbol{\theta},\boldsymbol{q},\dot{\boldsymbol{\theta}},\dot{\boldsymbol{q}}) = 0 \Longleftrightarrow \boldsymbol{q} = \bar{\boldsymbol{q}}(\boldsymbol{\theta}), \dot{\boldsymbol{q}} = \boldsymbol{0} \tag{5.4}$$

$\forall \boldsymbol{\theta} \in \mathbb{Q}$. The proof of this property is a direct consequence of the following Lemma 5.1.

Lemma 5.1. *Given assumption 2.4, the total potential function $U(\boldsymbol{\theta},\boldsymbol{q}) = U_g(\boldsymbol{q}) + U_k(\boldsymbol{\theta}-\boldsymbol{q})$, satisfies the conditions*

$$\begin{cases} U(\boldsymbol{\theta},\boldsymbol{q}) \geq U(\boldsymbol{\theta},\bar{\boldsymbol{q}}(\boldsymbol{\theta})) & \forall \boldsymbol{\theta},\boldsymbol{q} \in \mathbb{Q} \\ U(\boldsymbol{\theta},\boldsymbol{q}) = U(\boldsymbol{\theta},\bar{\boldsymbol{q}}(\boldsymbol{\theta})) & \boldsymbol{q} = \bar{\boldsymbol{q}}(\boldsymbol{\theta}), \boldsymbol{\theta} \in \mathbb{Q}\,. \end{cases} \tag{5.5}$$

Proof. Given any potential function $U_0(\chi)$ it is always possible to write

$$U_0(\chi_2) - U_0(\chi_1) - \left.\frac{\partial U_0(\chi)}{\partial \chi}\right|_{\chi_1} \Big(\chi_2 - \chi_1\Big) = i_0(\chi_1,\chi_2)\,, \tag{5.6}$$

where $i_0(\chi_1,\chi_2) = \int_{\chi_1}^{\chi_2} \left(\int_{\chi_1}^{\chi} \frac{\partial^2 U_0(\chi)}{\partial \chi^2}\Big|_{\xi} \mathrm{d}\xi \right)^T \mathrm{d}\chi$.

Moreover, since $i_0(\chi_1,\chi_2)$ is path independent, one can choose in particular the paths $\xi(\lambda):[0,1]\to\mathbb{Q}$ and $\chi(\gamma):[0,1]\to\mathbb{Q}$

$$\xi(\lambda) := \chi_1 + \lambda\Big(\chi - \chi_1\Big)$$
$$\chi(\gamma) := \chi_1 + \gamma\Big(\chi_2 - \chi_1\Big)$$

such that, evaluating the integral along them, it follows

$$\int_0^1\int_0^1 \Big(\chi_2-\chi_1\Big)^T \frac{\partial^2 U_0(\chi)}{\partial\chi^2}\bigg|_{\chi_1+\gamma\lambda(\chi_2-\chi_1)} \Big(\chi_2-\chi_1\Big)\gamma\,\mathrm{d}\lambda\,\mathrm{d}\gamma\,. \tag{5.7}$$

If $U_0(\chi)=U_g(q)$, taking the absolute value and using Proposition 2.1 leads to

$$|i_g(q_1,q_2)| \le \int_0^1\int_0^1 \sup_{\forall q\in\mathbb{Q}}\left\|\frac{\partial^2 U_g(q)}{\partial q^2}\right\| \|q_2-q_1\|^2\,\gamma\,\mathrm{d}\lambda\,\mathrm{d}\gamma < \frac{1}{2}\beta\,\|q_2-q_1\|^2\,, \tag{5.8}$$

while if $U_0(\chi)=U_k(\phi)$, using assumption 2.4 (A2) it follows that

$$\frac{1}{2}\alpha_1\,\|\phi_2-\phi_1\|^2 < i_k(\phi_1,\phi_2) < \frac{1}{2}\alpha_2\,\|\phi_2-\phi_1\|^2\,, \tag{5.9}$$

where i_g and i_k are to be considered as the function i_0 with the correspondent potential function in it. Using $i_g(\bar q(\theta),q)$ and $i_k(\theta-\bar q(\theta),\theta-q)$ the following holds

$$\begin{aligned} U(\theta,q)-U(\theta,\bar q(\theta)) = {} & i_g(\bar q(\theta),q)+i_k(\theta-\bar q(\theta),\theta-q)+ \\ & \Big(g(\bar q(\theta))-\tau(\theta-\bar q(\theta))\Big)^T\Big(q-\bar q(\theta)\Big)\,, \end{aligned} \tag{5.10}$$

which leads to the inequality

$$U(\theta,q)-U(\theta,\bar q(\theta)) < \frac{1}{2}\Big(\alpha_1-\beta\Big)\,\|q-\bar q(\theta)\|^2\,, \tag{5.11}$$

if (5.2), (5.8) and (5.9) are used. From the last inequality and assumption 2.4 (A3) the conditions on the total potential function $U(\theta,q)$ follow directly, since $\alpha_1-\beta>0$. □

5.2. Motivating example for the elastic case

The control law derived in this section is the base for the more general one in section 5.3. When applied to a robot manipulator, it can be used to regulate the robot to a given configuration, while when applied to a single joint setup can also be used to induce a controlled oscillation. The oscillation, in the nominal case (i.e. absence of friction), is achieved since the motor pumps in the required energy and then stops, letting the spring sustain the oscillation.

Theorem 5.1. *Given the system* (5.1) *and the assumption 2.4, let K_H be a positive scalar, K_θ and D_θ two symmetric and positive definite matrices and finally let the tilde denote the difference between a variable and the correspondent desired constant value, then the control input*

$$\tau_m = \tau(\phi) - K_H\tilde H(\theta,q,\dot\theta,\dot q)\Big(\tau(\phi)-g(\bar q(\theta))\Big) - K_\theta\tilde\theta - D_\theta\dot\theta\,, \tag{5.12}$$

will produce a closed loop system that will always reach the set

$$\mathscr{L}_d = \big\{(\theta,q,\dot\theta,\dot q)\mid\ \theta=\theta_d,\dot\theta=0,H(\theta_d,q,0,\dot q)=H_d\big\}$$

and therefore:

(a) for $H_d = 0$ has an asymptotically stable equilibrium point in $\boldsymbol{\chi}_0 = (\boldsymbol{\theta}_d, \dot{\boldsymbol{\theta}} = \mathbf{0}, \bar{\boldsymbol{q}}(\boldsymbol{\theta}_d), \dot{\boldsymbol{q}} = \mathbf{0})$

(b) for $H_d > 0$ and $n = 1$ has an asymptotically stable limit cycle defined by $\mathscr{L}_d$.

Proof. The proof is a bit cumbersome and therefore split in multiple parts. The goal is to show that $H \to H_d$ always. First a Lyapunov function that shows the stability of the invariant set $\mathscr{L}_d$ for the closed loop system is considered. Using LaSalle's invariance principle it is then proven that the equilibrium point $\boldsymbol{\chi}_0$ and the set $\mathscr{L}_d$ are the only positive limit sets for the solutions of (5.1) with the control law (5.12). Finally, depending on the value of H_d, the results (a) and (b) naturally arise.

Stability The C^1 function of the state

$$V(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = \frac{1}{2} K_H \tilde{H}^2(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) + \frac{1}{2} \tilde{\boldsymbol{\theta}}^T \boldsymbol{K}_\theta \tilde{\boldsymbol{\theta}} + \frac{1}{2} \dot{\boldsymbol{\theta}}^T \boldsymbol{B} \dot{\boldsymbol{\theta}} \tag{5.13}$$

is such that

$$\begin{cases} V(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = 0 & \forall (\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) \in \mathscr{L}_d \\ V(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) > 0 & \forall (\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) \notin \mathscr{L}_d \end{cases}$$

and therefore a candidate Lyapunov function. Before computing its derivative along the flow of the closed loop system, the same needs to be done for (5.3). To this end, it is useful to first notice that

$$\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}, \bar{\boldsymbol{q}}(\boldsymbol{\theta})) = \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}, \boldsymbol{q})\big|_{(\boldsymbol{\theta}, \bar{\boldsymbol{q}}(\boldsymbol{\theta}))} = \boldsymbol{\tau}(\boldsymbol{\theta} - \bar{\boldsymbol{q}}(\boldsymbol{\theta})) \,, \tag{5.14}$$

since the term depending on $\nabla_{\boldsymbol{q}} U(\boldsymbol{\theta}, \boldsymbol{q})\big|_{(\boldsymbol{\theta}, \bar{\boldsymbol{q}}(\boldsymbol{\theta}))}$ disappears due to (5.2). From (5.2) it also follows that $\boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) = \boldsymbol{\tau}(\boldsymbol{\theta} - \bar{\boldsymbol{q}}(\boldsymbol{\theta}))$, leading to

$$\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}, \bar{\boldsymbol{q}}(\boldsymbol{\theta})) = \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) \,, \tag{5.15}$$

which finally is used to obtain respectively

$$\dot{H}(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{\theta}}^T \Big(\boldsymbol{\tau}(\phi) - \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) \Big) \tag{5.16}$$

$$\dot{V}(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = -\dot{\boldsymbol{\theta}}^T \boldsymbol{D}_\theta \dot{\boldsymbol{\theta}} \,. \tag{5.17}$$

The latter, being a negative semidefinite function of the state, ensures the stability of $\mathscr{L}_d$.

In order to refine the result (i.e. show that $\mathscr{L}_d$ is asymptotically stable) one must look for the largest invariant set M within the set E defined as

$$E = \left\{ (\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) \mid \dot{V}(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = 0 \right\} \,. \tag{5.18}$$

By virtue of LaSalle's invariance principle, then every solution of the system approaches M as $t \to \infty$. Since from (5.17) $\dot{V}(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = 0 \Leftrightarrow \dot{\boldsymbol{\theta}} = \mathbf{0}$, the next goal is to show that the condition $\dot{\boldsymbol{\theta}} = \mathbf{0}$ implies $M = \{\boldsymbol{\chi}_0, \mathscr{L}_d\}$.

Invariant set Firstly, it can be shown that $\dot{\boldsymbol{\theta}} = \mathbf{0} \Rightarrow \tilde{\boldsymbol{\theta}} = \mathbf{0}$. In fact, $\dot{\boldsymbol{\theta}} = \mathbf{0}$ implies that $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, $\boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) = \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta}_0)) = \bar{\boldsymbol{g}}_0$ and, because of (5.33), also $H(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}}) = H_0$, where $\boldsymbol{\theta}_0$, $\bar{\boldsymbol{g}}_0$ and H_0 are constants. Moreover, from (2.21b) and (5.12) it follows that

$$-K_H\Big(H_0 - H_d\Big)\Big(\boldsymbol{\tau}(\boldsymbol{\phi}) - \bar{\boldsymbol{g}}_0\Big) - \boldsymbol{K}_\theta\Big(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_d\Big) = \mathbf{0}\ , \tag{5.19}$$

from which two cases are possible

$$\begin{cases} H_0 = H_d \\ H_0 \neq H_d \end{cases} .$$

While from the first case it can be directly concluded that $\boldsymbol{\theta} = \boldsymbol{\theta}_d$, in the second case it can be concluded that $\boldsymbol{\tau}(\boldsymbol{\phi}) = \boldsymbol{\tau}_0$, with $\boldsymbol{\tau}_0$ constant. It can be shown that also the latter condition leads to $\boldsymbol{\theta} = \boldsymbol{\theta}_d$. From the definition of $\boldsymbol{\tau}(\boldsymbol{\phi})$ and the fundamental theorem of calculus for line integrals, one gets

$$U_k(\boldsymbol{\phi}_1) - U_k(\boldsymbol{\phi}_2) = \boldsymbol{\tau}_0^T\Big(\boldsymbol{\phi}_1 - \boldsymbol{\phi}_2\Big) \tag{5.20}$$

which holds, without contradicting assumption 2.4 (A1), $\forall \boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \in \mathbb{R}^n$ if and only if $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_2$. This implies $\boldsymbol{\phi} = \boldsymbol{\phi}_0$ and $\boldsymbol{q} = \boldsymbol{q}_0$, where $\boldsymbol{\phi}_0$ and $\boldsymbol{q}_0$ are constants. If $\boldsymbol{q}$ is constant, (2.16) implies that $\boldsymbol{g}(\boldsymbol{q}_0) = \boldsymbol{\tau}_0$ and from the uniqueness of (5.2) that $\boldsymbol{q}_0 = \bar{\boldsymbol{q}}(\boldsymbol{\theta}_0)$. Concluding, if $H_0 \neq H_d$ then $\boldsymbol{\tau}(\boldsymbol{\phi}) = \bar{\boldsymbol{g}}_0$ and therefore also in this case $\boldsymbol{\theta} = \boldsymbol{\theta}_d$.

Now it can be proven that $\dot{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}} = \mathbf{0} \Rightarrow M = \{\boldsymbol{\chi}_0, \mathscr{L}_d\}$. In fact, in addition to the previous conditions also $\boldsymbol{\theta} = \boldsymbol{\theta}_d$ holds, therefore from (2.21b) and (5.12) it follows that

$$-K_H\Big(H_0 - H_d\Big)\Big(\boldsymbol{\tau}(\boldsymbol{\theta}_d - \boldsymbol{q}) - \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta}_d))\Big) = \mathbf{0}\ . \tag{5.21}$$

If $H_0 = H_d$ one gets $\mathscr{L}_d$ by definition, while in case $H_0 \neq H_d$ then $\boldsymbol{\tau}(\boldsymbol{\theta}_d - \boldsymbol{q}) = \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta}_d))$, that is $\boldsymbol{\tau}(\boldsymbol{\phi})$ is constant and, as it has been shown before, this implies that $\boldsymbol{q} = \bar{\boldsymbol{q}}(\boldsymbol{\theta}_d)$, $\dot{\boldsymbol{q}} = \mathbf{0}$ or in other words $(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\chi}_0$.

Asymptotic stability For the case $H_d = 0$, using (5.4) it can be easily verified that the positively invariant set $\mathscr{L}_d$ coincides with $\boldsymbol{\chi}_0$, which becomes the only positive limit set and therefore result (a) is obtained. On the other hand, when $H_d > 0$, starting from a neighborhood of $\mathscr{L}_d$ which does not contain the equilibrium the system will always converge to $\mathscr{L}_d$. In case $n = 1$ the set $\mathscr{L}_d$ is a closed orbit in the state space and therefore a limit cycle, proving result (b). □

5.2.1. Controller discussion

The key feature of the proposed controller is in a torque feedback that, depending on the energy error, can be either positive or negative and therefore intuitively able to either inject or dissipate energy. The gain of the feedback can be additionally tuned if a generalization of the control law is used

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}(\boldsymbol{\phi}) - K_H\tilde{H}^{m-1}(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})\Big(\boldsymbol{\tau}(\boldsymbol{\phi}) - \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta}))\Big) - \boldsymbol{K}_\theta\tilde{\boldsymbol{\theta}} - \boldsymbol{D}_\theta\dot{\boldsymbol{\theta}}\ ,$$

obtained through the more general term $\frac{1}{m}K_H\tilde{H}^m(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})$ in the Lyapunov function, with $m = 2k$, $k \in \mathbb{N}^+$. Obviously, the case examined so far is retrieved when $k = 1$.

When used to produce an oscillation, the proposed controller is expected to be very efficient. As it was shown, after a transient in which the motor brings the energy to the desired level no

more power is provided by it since the velocity goes to zero. Basically, the oscillation will be sustained by the springs indefinitely. Obviously, this can happen only in the nominal case while in reality there will be always friction taking out energy that the motor will have to inject again in the system.

Concerning the computational cost of the proposed control law (5.12) it is clear that the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ and the total potential energy need to be computed in order to evaluate $H(\boldsymbol{\theta},\boldsymbol{q},\dot{\boldsymbol{\theta}},\dot{\boldsymbol{q}})$. In addition, the gradient of the potential functions and the function $\bar{\boldsymbol{q}}$ are required. The latter can be computed using a fixed-point iteration or Newton method to solve (5.2), as shown in [ASOP12] where also comments concerning the efficiency of the approach can be found. Finally, the Coriolis and centrifugal terms (which are the most costly) are not required, as well as no higher order differentiation of any of the involved expressions. This is instead often the case in many nonlinear controllers, e.g. those based on feedback linearization.

5.3. Elastically actuated manipulator

As for the rigidly actuated case, the key point is to introduce enough constraints so that the "remaining dynamics" of the system is equivalent to the single joint case that has been just examined. The designed controller is itself a nonlinear dynamic system. Loosely speaking, the motor dynamics is used to satisfy the virtual constraints, while the additional dynamics of the controller is used to regulate the energy of the system. In chapter 4, the virtual potential energy was chosen to have a constrained local minimum at the submanifold. This suggests that a modification of the energy function defined in section 5.1 is necessary to take into account the presence of the virtual constraints. Before presenting the control law, it is therefore necessary to extend some results to take into account that the system will be forced to evolve on a submanifold.

For the derivation of the control law it will be additionally required:

Assumption 5.1.

- $U_k(\boldsymbol{\theta}-\boldsymbol{q}) = \dfrac{1}{2}\displaystyle\sum_{i=1}^{n} K_i\Big(\boldsymbol{\theta}_i-\boldsymbol{q}_i\Big)^2$ (A'1)
- $\beta < \min\limits_i K_i \qquad\qquad i=1,\ldots,n$ (A'2)

Notice that assumption 5.1 actually replaces assumption 2.4, since any such a $U_k(\boldsymbol{\theta}-\boldsymbol{q})$ always fulfills assumption 2.4. Finally, the stiffness constants can be collected in the diagonal joint stiffness matrix $\boldsymbol{K}\in\mathbb{R}^{n\times n}$.

5.3.1. The link-side equilibrium configurations

The function $\bar{\boldsymbol{q}}$ can be generalized to take into account constraints on the allowed configurations of the robot. This is exactly what happens when the system is forced to evolve on a submanifold. To this end, firstly an alternative definition of the function $\bar{\boldsymbol{q}}$ is provided using Lemma 5.1

$$\begin{aligned}\bar{\boldsymbol{q}}(\boldsymbol{\theta}) &= \arg\min_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q}) - U(\boldsymbol{\theta},\bar{\boldsymbol{q}}(\boldsymbol{\theta}))\\ &= \arg\min_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q})\ ,\end{aligned} \tag{5.22}$$

$\forall\boldsymbol{\theta}\in\mathbb{Q}$. Notice that the necessary condition for optimality of the minimization problem coincides with the definition of $\bar{\boldsymbol{q}}(\boldsymbol{\theta})$ in (5.2). At this point the constraint is taken into account by simply

modifying the problem as

$$\begin{aligned} \bar{\boldsymbol{q}}_y(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{q}} \; & U(\boldsymbol{\theta},\boldsymbol{q}) \\ \text{s.t. } & \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}\,, \end{aligned} \tag{5.23}$$

$\forall \boldsymbol{\theta} \in \mathbb{Q}$. The necessary condition for optimality obtained using the Lagrange multipliers $\boldsymbol{\lambda}$ is

$$\nabla_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\lambda} = 0\,, \tag{5.24}$$

so that pre-multiplying by the nullspace base matrix $\boldsymbol{Z}(\boldsymbol{q})$, the condition

$$n(\boldsymbol{q}) := \boldsymbol{Z}(\boldsymbol{q})\Big[\boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}\Big(\boldsymbol{\theta} - \boldsymbol{q}\Big)\Big] = 0 \tag{5.25}$$

is obtained, where $n(\boldsymbol{q})$ can be seen as a local nullspace coordinate [Bai85]. Concluding, for a given $\boldsymbol{\theta} \in \mathbb{Q}$, $\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})$ is the unique configuration with coordinates $\boldsymbol{y}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = \boldsymbol{0}$, $n(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = 0$. This suggests that, from a computational point of view, the optimization problem can be solved setting up an inverse kinematic problem and finding $\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})$ with a standard closed loop inverse kinematic scheme [SSVO08]. As it can be noticed, in this way, the definition of the function $\bar{\boldsymbol{q}}_y$ is very similar to the desired configuration in chapter 4.

Proposition 5.1. *The function $\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})$, in a similar manner to $\bar{\boldsymbol{q}}(\boldsymbol{\theta})$, satisfies the properties*

- $U(\boldsymbol{\theta},\boldsymbol{q}) \geq U(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))$ $\qquad \forall \boldsymbol{\theta},\boldsymbol{q} \in \mathbb{Q} : \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}$
- $U(\boldsymbol{\theta},\boldsymbol{q}) = U(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) \Leftrightarrow \boldsymbol{q} = \bar{\boldsymbol{q}}_y(\boldsymbol{\theta})$ $\qquad \forall \boldsymbol{\theta},\boldsymbol{q} \in \mathbb{Q} : \boldsymbol{y}(\boldsymbol{q}) = \boldsymbol{0}$
- $\dot{U}(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = \dot{\boldsymbol{\theta}}^T \boldsymbol{K}\Big(\boldsymbol{\theta} - \bar{\boldsymbol{q}}_y(\boldsymbol{\theta})\Big)$

Proof. While the first two properties are a direct consequence of how the function is defined, for the third the following must be taken into account. By definition $\boldsymbol{y}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = \boldsymbol{0}$ and therefore $\boldsymbol{J}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))\dot{\bar{\boldsymbol{q}}}_y(\boldsymbol{\theta}) = \boldsymbol{0}$, $\forall \boldsymbol{\theta}(t) \in \mathbb{Q}$. Using the chain rule it is possible to write

$$\dot{U}(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = \dot{\boldsymbol{\theta}}^T \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta},\boldsymbol{q})\big|_{(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))} + \dot{\bar{\boldsymbol{q}}}_y^T(\boldsymbol{\theta}) \nabla_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q})\big|_{(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))}\,.$$

The second term can be shown to be always zero rewriting it as

$$\begin{aligned} &\dot{\bar{\boldsymbol{q}}}_y^T(\boldsymbol{\theta})\, \boldsymbol{J}_N^T(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))\boldsymbol{J}_N^{-T}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))\nabla_{\boldsymbol{q}} U(\boldsymbol{\theta},\boldsymbol{q})\big|_{(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))} = \\ &\begin{bmatrix} \boldsymbol{0} & \dot{\bar{\boldsymbol{q}}}_y^T(\boldsymbol{\theta})\, \boldsymbol{N}^T(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) \end{bmatrix} \begin{bmatrix} \nabla_{\boldsymbol{q}}^T U(\boldsymbol{\theta},\boldsymbol{q})\big|_{(\boldsymbol{\theta},\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))}\, \boldsymbol{J}^{+_M}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) & \boldsymbol{0} \end{bmatrix}^T\,, \end{aligned}$$

where $\boldsymbol{J}(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta}))\dot{\bar{\boldsymbol{q}}}_y(\boldsymbol{\theta}) = \boldsymbol{0}$ and $n(\bar{\boldsymbol{q}}_y(\boldsymbol{\theta})) = 0$ have been used. The proof is concluded considering that only the potential energy due to the springs depends on $\boldsymbol{\theta}$ in the expression of the total potential $U(\boldsymbol{\theta},\boldsymbol{q})$. □

5.3.2. Controller design

This section provides an intuitive description of the controller design, while the next ones provide a rigorous analysis of the closed loop system. The torque produced by the springs has to enforce the virtual constraints and produce, at the same time, the desired oscillation. To this end, it is split in two contributions as:

$$\boldsymbol{K}\Big(\boldsymbol{\theta} - \boldsymbol{q}\Big) = \boldsymbol{K}\Big(\boldsymbol{\theta} - \boldsymbol{\eta}\Big) + \boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big)\,, \tag{5.26}$$

where $\boldsymbol{\eta}$ is part of the controller state. The latter will have to online adjust this value in order to distribute the torques between the two tasks. To fulfill the constraints, the torques produced by the springs should converge to the torques $\boldsymbol{\tau}_x$ in section 4.3.1. The input to the motors $\boldsymbol{\tau}_m$ have to be chosen in such a way the the motors are driven in a position at which the desired torque is produced, i.e. $\boldsymbol{K}\Big(\boldsymbol{\theta}-\boldsymbol{\eta}\Big) \to \boldsymbol{\tau}_d$. The contribution of the spring torque which does not interfere with the constraints will be instead used to sustain the oscillation. Nevertheless, since the input to the system is somehow already used to fulfill the constraint, the dynamics of the controller has to be chosen to realize the regulation of the energy to the desired value. To this end, a dynamics that mimics the one of the motors is assigned to the controller.

5.3.3. Closed loop system

Given the positive scalars $K_H, K_\alpha \in \mathbb{R}$ and the symmetric, positive definite matrices $\boldsymbol{K}_\tau$, $\boldsymbol{D}_\tau$, $\boldsymbol{K}_\eta$, $\boldsymbol{D}_\eta \in \mathbb{R}^{n\times n}$, $\boldsymbol{K}_y, \boldsymbol{D}_y \in \mathbb{R}^{(n-1)\times(n-1)}$, the proposed nonlinear dynamic state feedback controller is

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + \boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big) = \boldsymbol{\rho} \tag{5.27a}$$

$$\boldsymbol{\tau}_m = \boldsymbol{\rho} + \boldsymbol{\tau}_d + \boldsymbol{B}\boldsymbol{K}^{-1}\Big(\ddot{\boldsymbol{\tau}}_d - \boldsymbol{D}_\tau\dot{\tilde{\boldsymbol{\tau}}} - \boldsymbol{K}_\tau\tilde{\boldsymbol{\tau}}\Big)\,, \tag{5.27b}$$

where $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}} \in \mathbb{R}^n$ is the state of the controller, $\boldsymbol{\tau}_m \in \mathbb{R}^n$ is the controller output function and $\tilde{\boldsymbol{\tau}} = \boldsymbol{K}\Big(\boldsymbol{\theta}-\boldsymbol{\eta}\Big) - \boldsymbol{\tau}_d$ being both $\boldsymbol{\tau}_d, \boldsymbol{\rho} \in \mathbb{R}^n$ the controller input functions. The latter are computed based on the state of the system and of the controller itself and are given by

$$\boldsymbol{\tau}_d = \boldsymbol{J}_N^T\left(\begin{bmatrix} -\boldsymbol{D}_y & \boldsymbol{\Gamma}_{yn} - K_\alpha\dot{\boldsymbol{y}}v \\ -\boldsymbol{\Gamma}_{yn}^T + K_\alpha v\dot{\boldsymbol{y}}^T & \boldsymbol{O} \end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{y}} \\ v\end{bmatrix} + \begin{bmatrix}\boldsymbol{J}^{+_M T}\Big[\boldsymbol{g} - \boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big)\Big] - \boldsymbol{K}_y\boldsymbol{y} \\ \boldsymbol{0}\end{bmatrix}\right) \tag{5.28a}$$

$$\boldsymbol{\rho} = \boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big) - K_H\tilde{H}(\boldsymbol{\chi})\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) - \boldsymbol{D}_\eta\dot{\boldsymbol{\eta}} - \boldsymbol{K}_\eta\tilde{\boldsymbol{\eta}}\,, \tag{5.28b}$$

where in this case

$$H(\boldsymbol{\chi}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}\dot{\boldsymbol{q}} + U(\boldsymbol{\eta},\boldsymbol{q}) - U(\boldsymbol{\eta},\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}))\,. \tag{5.29}$$

Interconnecting the system (5.1) with the controller (5.27) as in Fig. 5.1, produces the closed loop system:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big) + \boldsymbol{\tau}_d + \tilde{\boldsymbol{\tau}} \tag{5.30a}$$

$$\ddot{\tilde{\boldsymbol{\tau}}} + \boldsymbol{D}_\tau\dot{\tilde{\boldsymbol{\tau}}} + \Big(\boldsymbol{K}_\tau + \boldsymbol{K}\boldsymbol{B}^{-1}\Big)\tilde{\boldsymbol{\tau}} = \boldsymbol{0} \tag{5.30b}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + \boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big) = \boldsymbol{\rho}\,, \tag{5.30c}$$

which, using the state vector $\boldsymbol{\chi} = (\boldsymbol{q}, \dot{\boldsymbol{y}}, v, \boldsymbol{\tau}, \tilde{\boldsymbol{\tau}}, \boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ and the input functions (5.28), can be rewritten as (omitting the dependencies):

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M}\dot{\boldsymbol{y}} + \boldsymbol{Z}^T v \tag{5.31a}$$

$$\boldsymbol{\Lambda}_y\ddot{\boldsymbol{y}} + \big(\boldsymbol{\Gamma}_y + \boldsymbol{D}_y + K_\alpha v^2\boldsymbol{E}\big)\dot{\boldsymbol{y}} + \boldsymbol{K}_y\boldsymbol{y} = \boldsymbol{J}^{+_M T}\tilde{\boldsymbol{\tau}} \tag{5.31b}$$

$$\Lambda_n\dot{v} + \big(\Gamma_n - K_\alpha\dot{\boldsymbol{y}}^T\dot{\boldsymbol{y}}\big)v + \boldsymbol{Z}\boldsymbol{g} = \boldsymbol{Z}\Big[\boldsymbol{K}\Big(\boldsymbol{\eta}-\boldsymbol{q}\Big) + \tilde{\boldsymbol{\tau}}\Big] \tag{5.31c}$$

$$\ddot{\tilde{\boldsymbol{\tau}}} + \boldsymbol{D}_\tau\dot{\tilde{\boldsymbol{\tau}}} + \Big(\boldsymbol{K}_\tau + \boldsymbol{K}\boldsymbol{B}^{-1}\Big)\tilde{\boldsymbol{\tau}} = \boldsymbol{0} \tag{5.31d}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + K_H\tilde{H}\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) + \boldsymbol{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta\tilde{\boldsymbol{\eta}} = \boldsymbol{0}\,. \tag{5.31e}$$

The stability properties of the system are summarized in the following theorem.

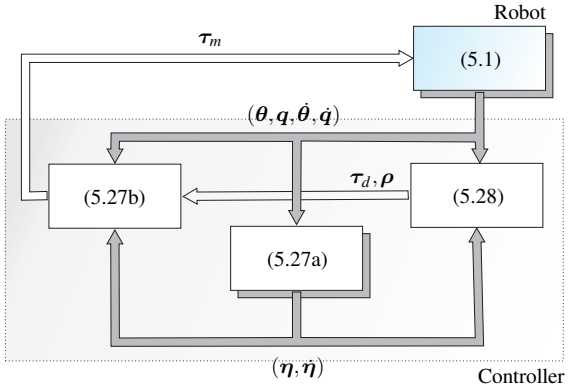

Figure 5.1.: Closed loop system.

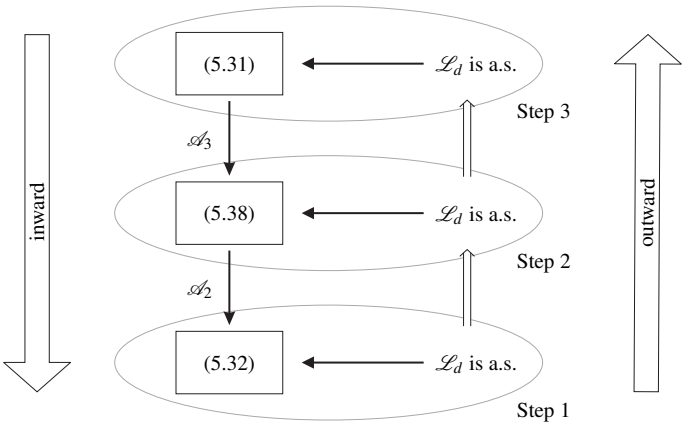

Figure 5.2.: Flow of the argumentation used in the proof.

Theorem 5.2 (Elastic case)**.** *Given assumption 5.1, the nonlinear autonomous system* (5.31)*, depending on the value of H_d, has an asymptotically stable solution consisting of*

(a) the equilibrium point $\boldsymbol{\chi}^* = (\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_d), \dot{\boldsymbol{q}} = \mathbf{0}, v = 0, \tilde{\boldsymbol{\tau}} = \mathbf{0}, \dot{\tilde{\boldsymbol{\tau}}} = \mathbf{0}, \boldsymbol{\eta}_d, \dot{\boldsymbol{\eta}} = \mathbf{0})$ *for* $H_d = 0$

(b) the limit cycle defined by $\mathscr{L}_d = \left\{\boldsymbol{\chi} \mid \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \mathbf{0},\ \tilde{\boldsymbol{\eta}} = \dot{\boldsymbol{\eta}} = \tilde{\boldsymbol{\tau}} = \dot{\tilde{\boldsymbol{\tau}}} = \mathbf{0},\ \tilde{H}(\boldsymbol{\chi}) = 0\right\}$ *for* $H_d > 0$.

Proof. The proof is based on the theorems and lemmas collected in Appendix A and it is split in three parts, since the conditional stability theorem will be used twice. The flow of the argumentation used in the proof is shown in Fig. 5.2. A semidefinite Lyapunov function $V_3(\boldsymbol{\chi})$ allows to carry on the stability analysis of (5.31). Therefore, it is considered the system conditionally to the largest invariant set within the set where $\dot{V}_3 = 0$, i.e. (5.38). Once again, a semidefinite Lyapunov function V_2 allows to carry on the stability analysis of (5.38). Therefore, it is considered the system conditionally to the largest invariant set within the set where $\dot{V}_2 = 0$, i.e. (5.32). Starting from the most inner system the proof proceeds outwards till (5.31) after two steps.

Step 1 The nonlinear autonomous system

$$\dot{\boldsymbol{q}} = \boldsymbol{Z}^T v \tag{5.32a}$$

$$\Lambda_n \dot{v} + \Gamma_n v + \boldsymbol{Z}\boldsymbol{g} = \boldsymbol{Z}\boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big) \tag{5.32b}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + K_H \tilde{H}\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) + \boldsymbol{D}_\eta \dot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta \tilde{\boldsymbol{\eta}} = \boldsymbol{0} \tag{5.32c}$$

with $\boldsymbol{\chi} \in \mathscr{A}_2 = \big\{\boldsymbol{\chi} \mid \tilde{\boldsymbol{\tau}} = \dot{\tilde{\boldsymbol{\tau}}} = \boldsymbol{0}, \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}\big\}$, will always reach the set $\mathscr{L}_d$ and therefore

- for $H_d = 0$ has an asymptotically stable equilibrium point in $\boldsymbol{\chi}^*$
- for $H_d > 0$ has asymptotically stable limit cycle defined by $\mathscr{L}_d$.

This is proven showing invariance, stability and finally attractiveness of $\mathscr{L}_d$, similarly to what has been done in section 5.2.

Invariance Given the properties of $\bar{\boldsymbol{q}}_y$, $H(\boldsymbol{\chi})$ is an "energy-like" function for (5.32). Additionally, computing the time derivative of $\tilde{H}$ along the flow of (5.32), results in

$$\dot{\tilde{H}}(\boldsymbol{\chi}) = \dot{H}(\boldsymbol{\chi}) = \dot{\boldsymbol{\eta}}^T \Big[\boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big) - \boldsymbol{K}\Big(\boldsymbol{\eta} - \bar{\boldsymbol{q}}_y(\boldsymbol{\eta})\Big)\Big] . \tag{5.33}$$

Since $\dot{\boldsymbol{\eta}} = \boldsymbol{0} \Longrightarrow \dot{\tilde{H}} = 0$, it can be concluded that $\mathscr{L}_d$ is an invariant set for (5.32), because $\tilde{H}$ will not change when starting from $\mathscr{L}_d$.

Stability The C^1 function of the state

$$V_1(\boldsymbol{\chi}) = \frac{1}{2}\left(K_H \tilde{H}^2(\boldsymbol{\chi}) + \dot{\boldsymbol{\eta}}^T \boldsymbol{B}\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \boldsymbol{K}_\eta \tilde{\boldsymbol{\eta}}\right) , \tag{5.34}$$

is such that

$$\begin{cases} V_1(\boldsymbol{\chi}) = 0 & \forall \boldsymbol{\chi} \in \mathscr{L}_d \\ V_1(\boldsymbol{\chi}) > 0 & \forall \boldsymbol{\chi} \notin \mathscr{L}_d \end{cases}$$

and therefore it is a candidate Lyapunov function to prove the stability of $\mathscr{L}_d$ for (5.32). Computing its derivative along the flow of the system (5.32), leads to $\dot{V}_1(\boldsymbol{\chi}) = -\dot{\boldsymbol{\eta}}^T \boldsymbol{D}_\eta \dot{\boldsymbol{\eta}} \leq 0$, which ensures the stability of $\mathscr{L}_d$ for the system (5.32).

LaSalle's invariance principle allows to conclude that the positive limit set for the solutions of (5.32) is given by the largest positive invariant set M_1 within the set $E_1 = \{\boldsymbol{\chi} \mid \dot{\boldsymbol{\eta}} = \boldsymbol{0}\}$, i.e. the set where $\dot{V}_1(\boldsymbol{\chi}) = 0$. Starting from the condition $\dot{\boldsymbol{\eta}} = \boldsymbol{0}$, it will be shown in two subsequent steps that $M_1 = \{\boldsymbol{\chi}^*, \mathscr{L}_d\}$.

Invariant set Firstly, it can be shown that $\dot{\boldsymbol{\eta}} = \boldsymbol{0} \Rightarrow \tilde{\boldsymbol{\eta}} = \boldsymbol{0}$. In fact, $\dot{\boldsymbol{\eta}} = \boldsymbol{0}$ implies that $\boldsymbol{\eta} = \boldsymbol{\eta}_0$ and, because of (5.33), also $H(\boldsymbol{\chi}) = H_0$, where $\boldsymbol{\eta}_0$ and H_0 are constants. Moreover, from (5.32c) it follows that

$$-K_H\Big(H_0 - H_d\Big)\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_0) - \boldsymbol{q}\Big) - \boldsymbol{K}_\eta\Big(\boldsymbol{\eta}_0 - \boldsymbol{\eta}_d\Big) = \boldsymbol{0} , \tag{5.35}$$

from which two cases are possible

$$\begin{cases} H_0 = H_d \\ H_0 \neq H_d . \end{cases}$$

While from the first case it can be directly concluded that $\boldsymbol{\eta}_0 = \boldsymbol{\eta}_d$, in the second case it can be concluded that $\boldsymbol{q} = \boldsymbol{q}_0$, where $\boldsymbol{q}_0$ is constant, since all the quantities in (5.35) are constants. This implies $v = 0$ and therefore from (5.32b) one gets

$$\boldsymbol{Z}(\boldsymbol{q}_0)\Big[\boldsymbol{g}(\boldsymbol{q}_0) - \boldsymbol{K}\Big(\boldsymbol{\eta}_0 - \boldsymbol{q}_0\Big)\Big] = 0\,.$$

In addition, $\boldsymbol{y}(\boldsymbol{q}_0) = \boldsymbol{0}$ holds, or in other words $\boldsymbol{q}_0 = \bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_0)$ and therefore from (5.35) also in this case $\boldsymbol{\eta}_0 = \boldsymbol{\eta}_d$.

Now it can be proven that $\dot{\boldsymbol{\eta}} = \tilde{\boldsymbol{\eta}} = \boldsymbol{0} \Rightarrow M_1 = \{\boldsymbol{\chi}^*, \mathscr{L}_d\}$. In fact, in addition to the previous conditions also $\boldsymbol{\eta} = \boldsymbol{\eta}_d$ holds, therefore from (5.32c) it follows that

$$-K_H\Big(H_0 - H_d\Big)\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_d) - \boldsymbol{q}\Big) = \boldsymbol{0}\,. \tag{5.36}$$

If $H_0 = H_d$ one gets $\mathscr{L}_d$ by definition, while in case $H_0 \neq H_d$ then $\boldsymbol{q} = \bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_d)$ and consequently $\dot{\boldsymbol{y}} = \boldsymbol{0}$, $v = 0$ or in other words $\boldsymbol{\chi} = \boldsymbol{\chi}^*$.

Asymptotic stability For the case $H_d = 0$, since

$$H(\boldsymbol{\chi}) = 0 \Longleftrightarrow \boldsymbol{q} = \bar{\boldsymbol{q}}_y(\boldsymbol{\eta}), v = 0 \tag{5.37}$$

$\forall \boldsymbol{\chi} \in \mathscr{A}_2$, it can be easily verified that $\mathscr{L}_d$ coincides with $\boldsymbol{\chi}^*$, which becomes the only positive limit set and therefore result (a) is obtained. On the other hand, when $H_d > 0$, starting from a neighborhood of $\mathscr{L}_d$ which does not contain the equilibrium point the system will always converge to $\mathscr{L}_d$. The set $\mathscr{L}_d$ is uniquely determined by one parameter, e.g. the value of $n(\boldsymbol{q})$, and therefore it is a closed[1] orbit in the state space, i.e. a limit cycle, proving result (b).

Given this result, in the remainder of the proof it will be simply referred to the stability property of the set $\mathscr{L}_d$, with the conditions (a) and (b) naturally arising depending on the value of H_d.

Step 2 Given the nonlinear autonomous system

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M}\dot{\boldsymbol{y}} + \boldsymbol{Z}^T v \tag{5.38a}$$

$$\boldsymbol{\Lambda}_y\ddot{\boldsymbol{y}} + \big(\boldsymbol{\Gamma}_y + \boldsymbol{D}_y + K_\alpha v^2\boldsymbol{E}\big)\dot{\boldsymbol{y}} + \boldsymbol{K}_y\boldsymbol{y} = \boldsymbol{0} \tag{5.38b}$$

$$\Lambda_n\dot{v} + \big(\Gamma_n - K_\alpha\dot{\boldsymbol{y}}^T\dot{\boldsymbol{y}}\big)v + \boldsymbol{Z}\boldsymbol{g} = \boldsymbol{Z}\boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big) \tag{5.38c}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + K_H\tilde{H}\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) + \boldsymbol{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta\tilde{\boldsymbol{\eta}} = \boldsymbol{0}\,, \tag{5.38d}$$

with $\boldsymbol{\chi} \in \mathscr{A}_3 = \big\{\boldsymbol{\chi} \mid \tilde{\boldsymbol{\tau}} = \dot{\tilde{\boldsymbol{\tau}}} = \boldsymbol{0}\big\}$, consider the C^1 function of the state

$$V_2(\boldsymbol{\chi}) = \frac{1}{2}\left(\dot{\boldsymbol{y}}^T\boldsymbol{\Lambda}_y(\boldsymbol{q})\dot{\boldsymbol{y}} + \boldsymbol{y}^T(\boldsymbol{q})\boldsymbol{K}_y\boldsymbol{y}(\boldsymbol{q})\right)\,, \tag{5.39}$$

such that

$$\begin{cases} V_2(\boldsymbol{\chi}) = 0 & \forall \boldsymbol{\chi} \in \mathscr{A}_2 \\ V_2(\boldsymbol{\chi}) > 0 & \forall \boldsymbol{\chi} \notin \mathscr{A}_2\,. \end{cases}$$

Since $\mathscr{L}_d \subset \mathscr{A}_2$, $V_2(\boldsymbol{\chi})$ is a candidate semidefinite Lyapunov function to prove the stability of $\mathscr{L}_d$ for (5.38). Computing its derivative along the flow of the system, it can be shown that $\dot{V}_2(\boldsymbol{\chi}) = -\dot{\boldsymbol{y}}^T\big(\boldsymbol{D}_y + K_\alpha v^2\boldsymbol{E}\big)\dot{\boldsymbol{y}} \leq 0$. The set $\mathscr{A}_2$ is the largest positively invariant set within $E_2 = \{\boldsymbol{\chi} \mid \dot{\boldsymbol{y}} = \boldsymbol{0}\}$, i.e. the set where $\dot{V}_2(\boldsymbol{\chi}) = 0$. In fact, if $\boldsymbol{y}(\boldsymbol{q}) \neq \boldsymbol{0}$ the system will leave E_2 as it is clear from (5.38b). Finally, since $\mathscr{L}_d$ is asymptotically stable conditionally to $\mathscr{A}_2$ (which is exactly what has been proven in the previous step), then by Theorem A.3 $\mathscr{L}_d$ is asymptotically stable for (5.38).

[1] It is the level set of a positive definite and radially unbounded function.

Step 3 Consider for the system (5.31) the C^1 function of the state

$$V_3(\boldsymbol{\chi}) = \frac{1}{2}\dot{\tilde{\boldsymbol{\tau}}}^T\dot{\tilde{\boldsymbol{\tau}}} + \frac{1}{2}\tilde{\boldsymbol{\tau}}^T\left(\boldsymbol{K}_\tau + \boldsymbol{K}\boldsymbol{B}^{-1}\right)\tilde{\boldsymbol{\tau}}\,, \tag{5.40}$$

such that

$$\begin{cases} V_3(\boldsymbol{\chi}) = 0 & \forall \boldsymbol{\chi} \in \mathscr{A}_3 \\ V_3(\boldsymbol{\chi}) > 0 & \forall \boldsymbol{\chi} \notin \mathscr{A}_3\,. \end{cases}$$

Since $\mathscr{L}_d \subset \mathscr{A}_3$, $V_3(\boldsymbol{\chi})$ is a candidate semidefinite Lyapunov function to prove the stability of $\mathscr{L}_d$ for (5.31). Computing its derivative along the flow of the system, leads to $\dot{V}_3(\boldsymbol{\chi}) = -\dot{\tilde{\boldsymbol{\tau}}}^T\boldsymbol{D}_\tau\dot{\tilde{\boldsymbol{\tau}}} \leq 0$. The set $\mathscr{A}_3$ is the largest positively invariant set within $E_3 = \left\{\boldsymbol{\chi} \mid \dot{\tilde{\boldsymbol{\tau}}} = \boldsymbol{0}\right\}$, i.e. the set where $\dot{V}_3(\boldsymbol{\chi}) = 0$. In fact, if $\tilde{\boldsymbol{\tau}} \neq \boldsymbol{0}$ then the system will leave E_3 as it is clear from (5.31d). Finally, since $\mathscr{L}_d$ is asymptotically stable conditionally to $\mathscr{A}_3$ (which is exactly what has been proven in the previous step), then by Theorem A.3 $\mathscr{L}_d$ is asymptotically stable for (5.31). □

A modification of the control law can be obtained by computing $\bar{\boldsymbol{q}}_y(\boldsymbol{\eta})$ using the current value of $\boldsymbol{y}$, instead of zero (i.e. instead of the desired one). In the proof, in fact, $\bar{\boldsymbol{q}}_y(\boldsymbol{\eta})$ is consider for the analysis of the system upon convergence of the virtual constraints, so that the current value of $\boldsymbol{y}$ coincides with the desired one. This modification results useful in case of interactions, as the set-point is only reached in case of no external forces, otherwise a virtual equilibrium point is reached.

5.3.4. Comparison to the rigid case

The input function $\boldsymbol{\tau}_d$ has the meaning of a desired input to the link side equation. For this reason, it is very similar to the control torque (4.24). Also for $\boldsymbol{\tau}_d$ different contributions can be recognized: the torque responsible for forcing the system to evolve on the constraint submanifold, the one compensating for the coupling terms in the Coriolis matrix and the one shifting energy from the constraint space to the nullspace. Only the part of the gravitational torque which cause the system to go off the submanifold are compensated this time, since the gravitational potential energy itself is used to produce the oscillation. The way the limit cycle is produced is also different. The springs in the joints are directly involved in this task. The torque that they produce is split in the one necessary to keep the system to evolve on the submanifold and the one that can be used to produce the limit cycle. This is achieved via the control state, which by mimicking the motor dynamics, is acting as a rest length adjustment.

The output function (5.27b) of the controller requires up to the second derivative of the input function $\boldsymbol{\tau}_d$. Since the latter is a function of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\boldsymbol{\eta}$, in addition to the state, it is required the knowledge of the link acceleration $\ddot{\boldsymbol{q}}$, the jerk $\boldsymbol{q}^{(3)}$ and $\ddot{\boldsymbol{\eta}}$. While the signal $\ddot{\boldsymbol{\eta}}$ is easily computed, i.e. through (5.31e), since the model of the controller can be reasonably assumed to be very well known, the same is not true for the link acceleration and jerk. One can compute these signals based on the model equations or alternatively, directly through acceleration sensors and appropriate filtering techniques. From an implementation prospective, this is a weak point, which is although shared by all the control laws that use the motors to make the torque produced by the springs track a desired one (i.e. using the rigid case as an intermediate design step [Ott08, Chapter 6]). This aspect is a bit less critical for the control law proposed in this section since the torque $\boldsymbol{\tau}_d$ that needs to be tracked is responsible only to guarantee the convergence to the invariant set of the state space and not for the regulation of the energy, as it will be clear with the results shown in section 5.4.3.

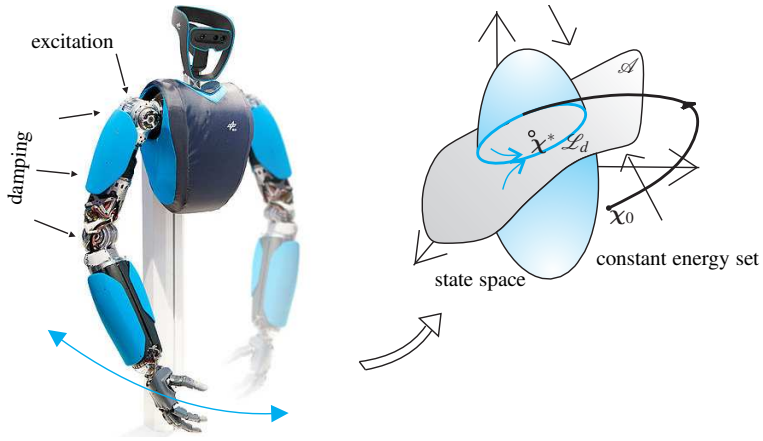

Figure 5.3.: The DLR Hand Arm System (HASy), used to validate the theoretical results.

5.4. Validation

In this section, the controllers (5.12) and (5.27) will be tested. The first one will be apllied to the robotic system in Fig. 5.3, both in simulation and experiments. The DLR Hand Arm System (HASy) [GASB+11] is a highly integrated robot, which aims to imitate the whole human upper limb in terms of size, weight and range of motion. Variable stiffness actuators (VSA), whose intrinsic nonlinear compliance can be adjusted by an additional motor, are used to replace the typically impedance controlled, mechanically stiff joints. As a result the robot can, for example, better handle collisions with stiff objects since part of the impact energy, which changes too quickly for an active controller, is temporarily stored in the elastic components. The system has more than 25 degrees of freedom, 50 motors and 100 position sensors. In the considered setup the stiffness will not be adjusted on line, so that the system dynamics can be described as in (5.1). Moreover, only the upper arm is used to focus the attention on the first four joints (which share the same mechanical design), as indicated in Fig. 5.3. Since the HASy does not satisfy assumpiotn 5.1, the second control law will be instead only tested in simulation.

In order to demonstrate the two possible application of the control law (5.12) as a way to excite or damp out the oscillations, two tests will be conducted. In the first one, an oscillation is produced in the first joint while the motion is damped out with a classic PD controller in the remaining ones. In particular the axis are oriented in such a way that the effect of the gravity are negligible and therefore the dynamic coupling between the joints can be approximately compensated with a high gain motor PD controller. In the other, the control law is used to regulate the robot to a desired configuration, which corresponds to ask for zero energy in the system. The two tests will always be labeled as respectively (a) and (b) in all the figures. Fig. 5.3 sums up the first test, highlighting the joints involved and their role. In addition, a conceptual interpretation of what happens in the state space is provided. Imagining to have a three dimensional state space, the set of constant desired energy can, for example, be thought as a sphere where the close loop system is bound to evolve. The additional constraints introduced through the PD controller, render one of the infinite many possible iso-energy orbits the one actually followed by the system. Basically the system can be thought as a one joint robot for which the existence of an asymptotically stable limit cycle has been proved. Fig. 5.4 supports this explanation. There the link positions of two trials obtained applying the control law (5.12) to all the joints of the robot with a strictly positive desired energy is shown. The only difference between the two trials is in the initial value of the link velocities. Even though in both cases the same desired energy is reached, as it can be seen the orbit followed

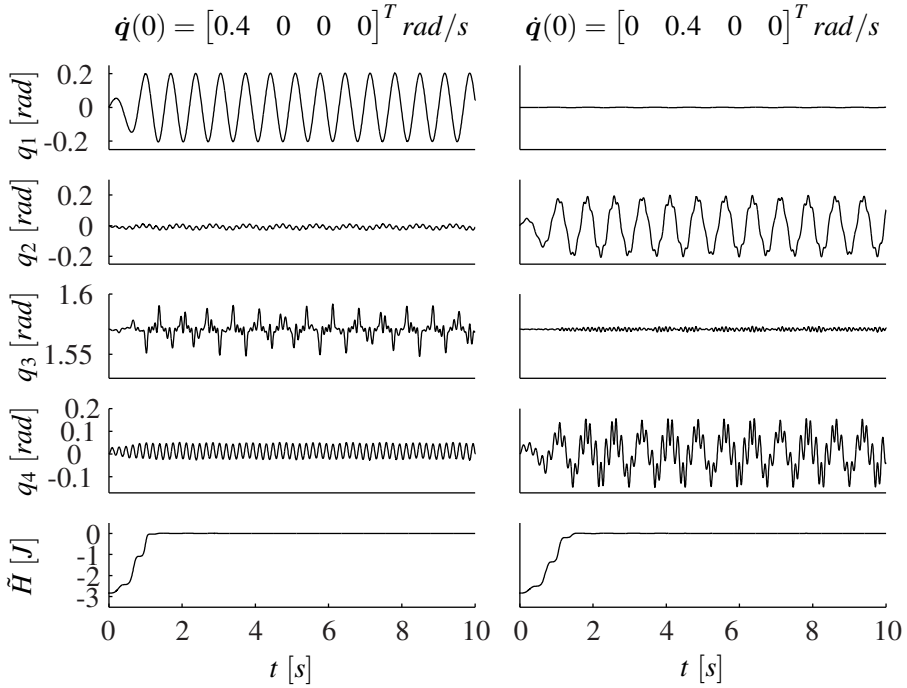

Figure 5.4.: Link positions obtained applying the controller (5.12) to all the joints with $H_d = 3\,\mathrm{J}$.

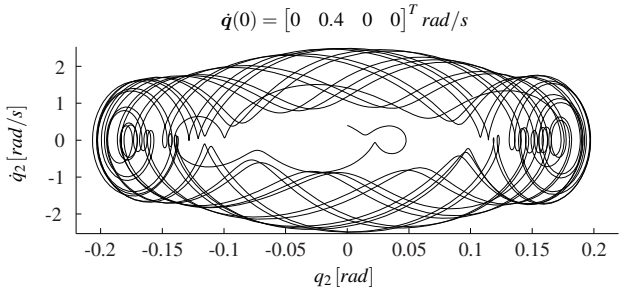

Figure 5.5.: Visualization of the chaotic behavior through the phase plot of the second joint.

by the system is different. The oscillation produced in this case is the consequence of a chaotic behavior and not a periodic one, as the phase plot of the link variables of the second joint shows in Fig. 5.5.

5.4.1. Simulations of the first controller

The simulations are carried out using MATLAB/Simulink ®, where the algorithm and formulas from chapter 2 are implemented to compute all the quantities necessary for the control law and the robot direct dynamics. In this section, all the plots will show with a blue line the performances of the controller in presence of a non modeled link side friction $-d\,\dot{\boldsymbol{q}}$, with $d = 0.5\,\mathrm{Nms/rad}$. While this term is of course not harmful when the system is required to damp out the oscillations, it forces the controller to re-inject the lost energy due to the unforeseen energy leak. This is especially clear in Fig. 5.6 where the convergence of the error $\tilde{H}(\boldsymbol{\theta},\boldsymbol{q},\dot{\boldsymbol{\theta}},\dot{\boldsymbol{q}}) = H(\boldsymbol{\theta},\boldsymbol{q},\dot{\boldsymbol{\theta}},\dot{\boldsymbol{q}}) - H_d$ is shown, both for damping of the oscillations and limit cycle generation. The different behavior of the system is simply due to a different value of H_d, which is $H_d = 3\,\mathrm{J}$ for limit cycle generation and $H_d = -1\,\mathrm{J}$

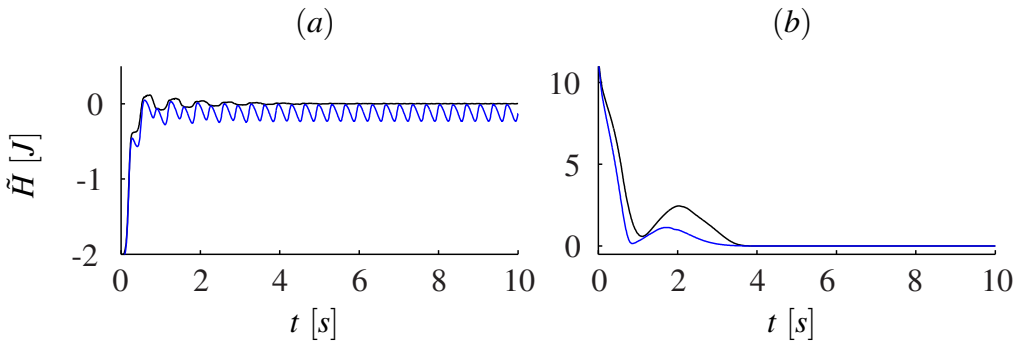

Figure 5.6.: Convergence of the energy error for limit cycle generation, case (a), and damping of the oscillations, case (b).

for regulation to the desired configuration[2].

The value of the gains for the simulations are listed in Table 5.1 and Table 5.2 respectively. The desired value for the motor position is $\boldsymbol{\theta}_d = \begin{bmatrix} 0 & 0 & \pi/2 & 0 \end{bmatrix}^T$ rad, as it can be seen in Fig. 5.7 for the two tests. It is worth to mention that the robot is always initialized in the equilibrium position, but with a nonzero initial link side velocity. The latter is always chosen to be $2\,\mathrm{rad/s}$ for the joints where the control law (5.12) is active, i.e. only the first joint in case of excitation and all the joints in case of damping of the oscillations. This explains why in Fig. 5.7, after a transient, the motors recover their initial positions.

Table 5.1.: Gains used for the simulation in Fig. 5.6, case (a). When only the i - th entry is shown, then the others have the same value.

K_H	K_{θ_i}	D_{θ_i}
$5\,\mathrm{J}^{-1}$	$20\,\mathrm{Nm/rad}$	$20\,\mathrm{Nms/rad}$

Table 5.2.: Gains used for the simulation in Fig. 5.6, case (b).

K_H	K_θ	D_θ
$0.5\,\mathrm{J}^{-1}$	$20\,\mathrm{Nm/rad}$	$20\,\mathrm{Nms/rad}$

In Fig. 5.7, for the case $H_d > 0$, a very small oscillation is present in the joints controlled with the simple PD law. This is not surprising due to the dynamic effect of one link to the other. For the same reason, also the link side positions will have a coupled oscillation and will qualitatively look like the ones in Fig. 5.4, case $\dot{\boldsymbol{q}}(0) = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix}^T$ rad/s. To conclude, in Fig. 5.8 the link positions, as well as the velocities, are shown from which it can be seen that the initial oscillations are damped out using a non positive desired value for $H(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})$.

5.4.2. Experiments

Using the same values of the parameters as in simulation, two tests are conducted with the real system. In the first one, the robot is initialized at the equilibrium point and the desired value

[2]Asking for a negative desired value for $H(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})$ has the only effect to increase the convergence speed and reduce residual small oscillations around the desired value due to numeric errors. The smallest possible value for $H(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})$ is zero, which is the one that will be reached. In Fig. 5.6 (b) the plot is shifted of 1 J to take this effect into account.

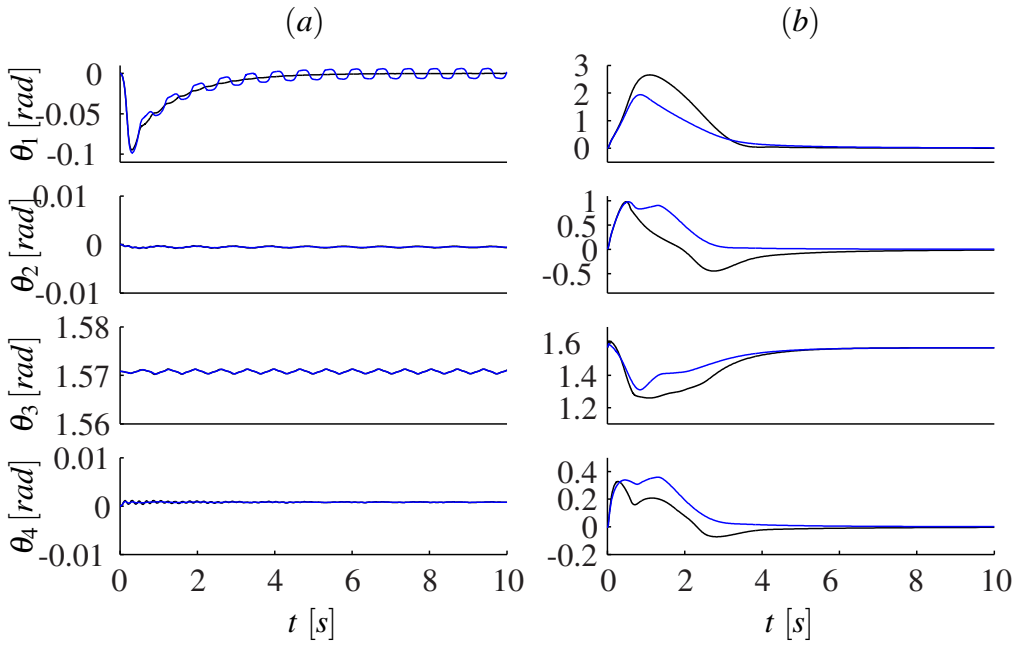

Figure 5.7.: Motor positions in case of limit cycle generation (left) and regulation (right).

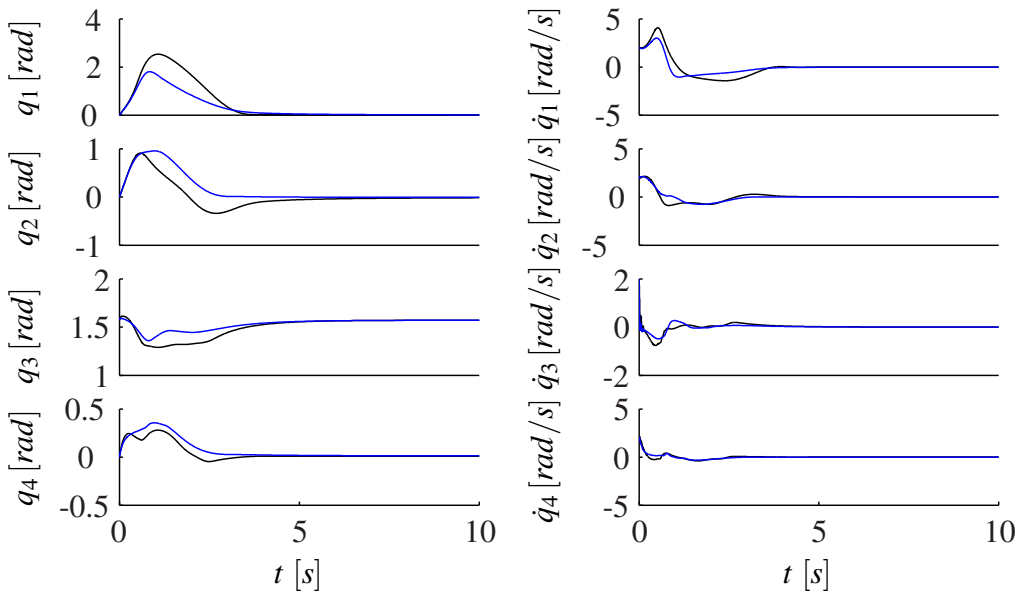

Figure 5.8.: Link position and velocities when an initial velocity perturbs the robot from the equilibrium point.

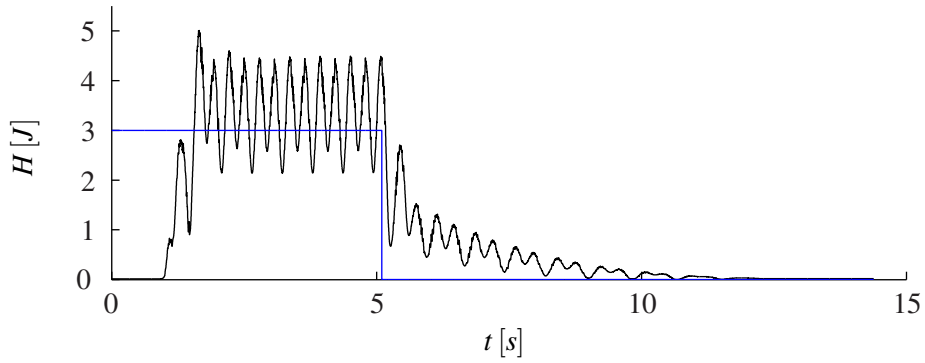

Figure 5.9.: Energy function (5.3) (black line) and its desired value (blue line).

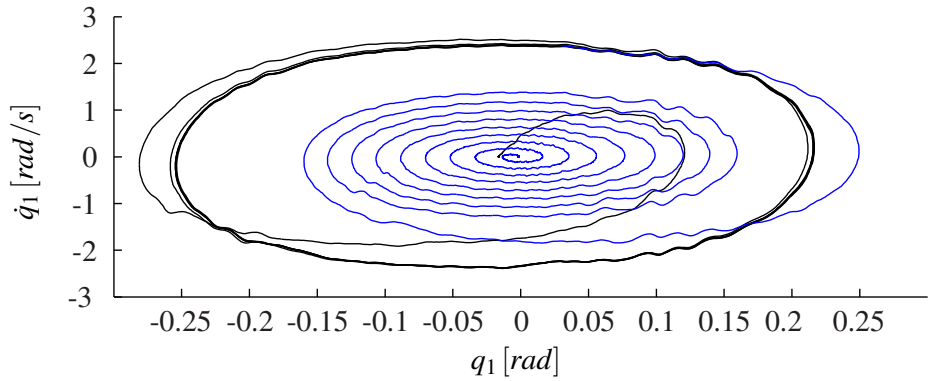

Figure 5.10.: Phase plot of the link side position and velocity of the first joint.

$H_d = 3\,\mathrm{J}$ is chosen for the energy function (5.3). The latter is shown in Fig. 5.9 together with its desired value. When the operator slightly pushes the robot, it starts moving away from the equilibrium point and oscillating around the desired value of the energy. As expected, the system does not exactly converge to the desired value, but it exhibits a behavior similar to the one observed in simulation in case of unmodeled friction. Of course, due to noise and bigger model uncertainty than the simulated case, a bigger deviation from the desired value is observed. When the desired value is switched to zero, the energy converges to it without any residual oscillation.

Particularly interesting is the phase plot of the link variables involved in the generation of the oscillation, shown in Fig. 5.10. There with a black line is shown the behavior when $H_d = 3\,\mathrm{J}$ and with a blue line when $H_d = 0$. As it can be seen, despite the presence of noise and model uncertainty, the system does reach a limit cycle. Basically, what will be perturbed is the shape of the limit cycle itself. Adaptive techniques and tuning of the parameters can of course reduce these effects. The spiraling convergence towards the asymptotically stable equilibrium point obtained when the desired energy is set to zero is also shown in the same plot. It is worth to mention that the damping behavior of the system can be tuned through H_d and K_H, since they will influence how quickly $H(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{q}})$ will converge to zero and therefore how quickly the link side oscillations will be damped out.

In Fig. 5.11 both the motor and link positions of the first joint are shown. It is interesting to notice that, although the motor is not fixed because of friction, the oscillations obtained on the link side are still amplified compared to those required from the motor.

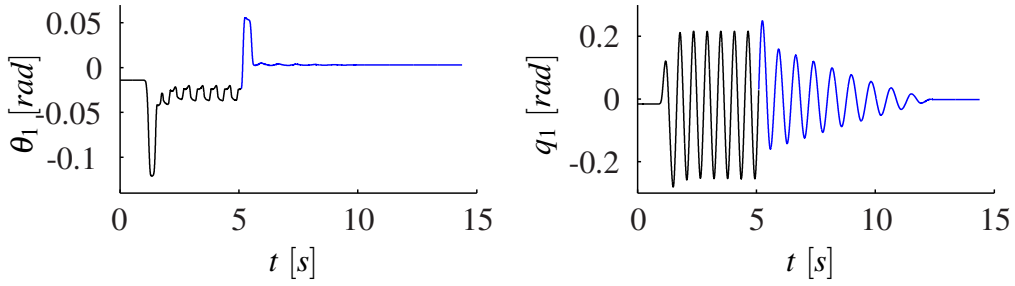

Figure 5.11.: Motor and link positions of the first joint when H_d is switched from $H_d = 3\,\mathrm{J}$ (black line) to $H_d = 0$ (blue line).

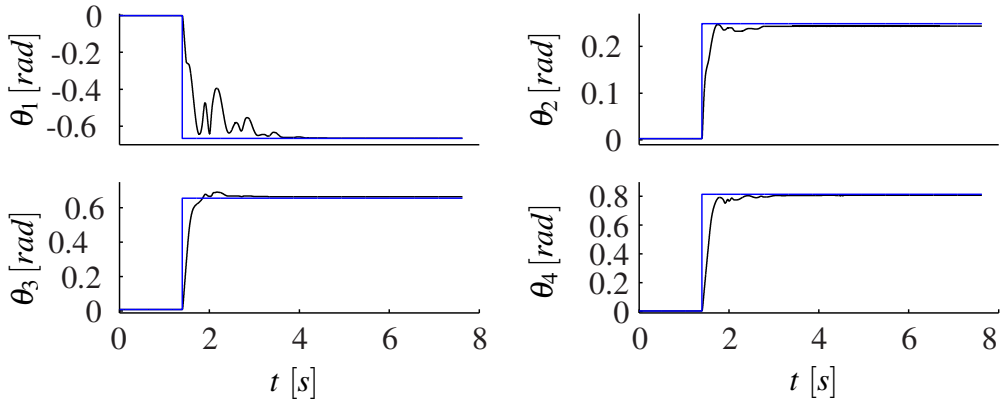

Figure 5.12.: Motor positions (black lines) and desired values (blue lines).

In the second test[3] all the joints are involved and the response of the system to a step in the desired value of the motor positions is considered. The result is reported in Fig. 5.12, where it can be noticed that the systems shows a good convergence behavior.

Concerning the link side, in Fig. 5.13 the link velocities are shown which prove that the robot can reach the new desired equilibrium point and no unwanted residual oscillations of the links are present.

5.4.3. Simulation results for the dynamic state feedback controller

As case study a 3 - link robotic arm is considered, with each link having the properties reported in Table 5.3. The robot starts from the configuration shown in Fig 5.14.

Table 5.3.: Properties of one of the three modules of the manipulator in Fig 5.14.

Link			Spring	Motor
Length	Mass	Inertia	Stiffness	Inertia
0.4 m	5 kg	0.2 kgm^2	200 Nm/rad	0.6325 kgm^2

The function $\boldsymbol{y}(\boldsymbol{q})$ is chosen as

$$y_1(\boldsymbol{q}) = z(\boldsymbol{q}) - 0.8$$
$$y_2(\boldsymbol{q}) = \psi(\boldsymbol{q}) \,,$$

[3] In this experiment, to show the high achievable performances of the system, $K_{\theta_i} = 400\,\mathrm{Nm/rad}$.

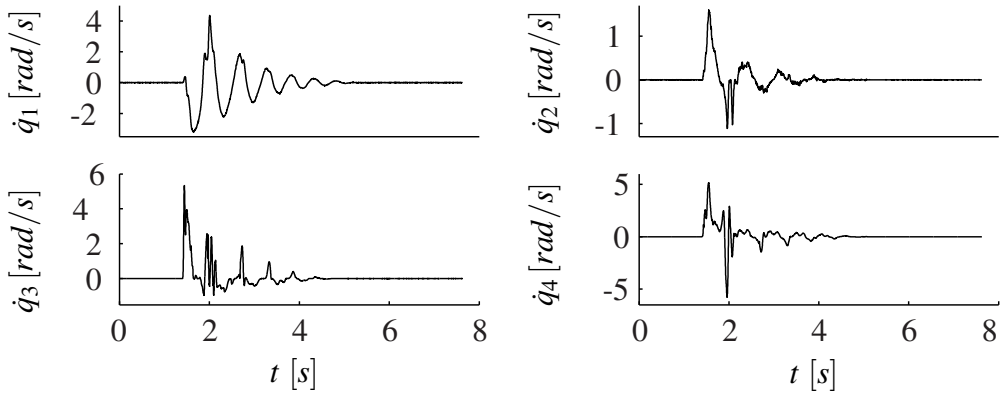

Figure 5.13.: Link velocities obtained after a step in the desired motor position, with $H_d = 0$.

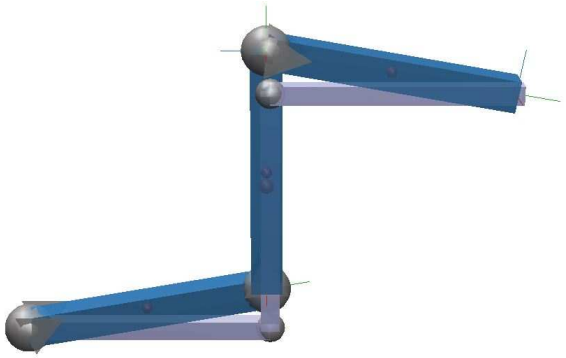

Figure 5.14.: Starting configuration of the robot ed equilibrium configuration (lighter color).

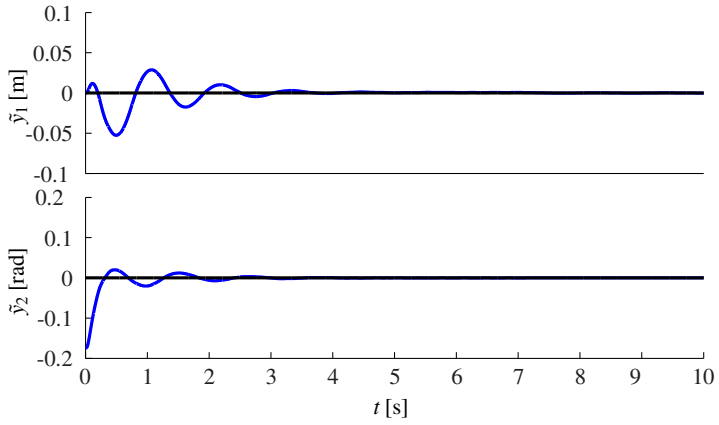

Figure 5.15.: Convergence to zero of the two components of the constraint function $\boldsymbol{y}(\boldsymbol{q})$.

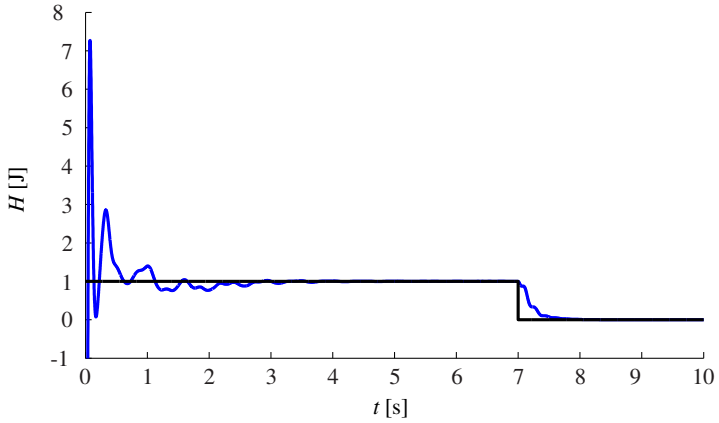

Figure 5.16.: Energy function (5.29) (blue line) and desired value (black line).

where $z(\boldsymbol{q})$ is the vertical position and $\psi(\boldsymbol{q})$ the orientation of the end - effector which, given the definition of $\boldsymbol{y}(\boldsymbol{q})$, is required to stay at a height of 0.8 m from the floor and to keep the orientation parallel to it. The convergence of $\boldsymbol{y}(\boldsymbol{q})$ to zero is shown in Fig. 5.15.

The desired value of the energy is switched from $H_d = 1\,\mathrm{J}$ to zero in order to show how the oscillation can be produced and then damped out. Given the definition of $\boldsymbol{y}(\boldsymbol{q})$, the resulting motion will be an horizontal oscillation when $H_d > 0$ while the robot will stop at the equilibrium when $H_d = 0$. The evolution of the energy function is shown in Fig. 5.16. As it can be seen, the energy is effectively regulated to the desired value, whose sudden variation has no influence on $\boldsymbol{y}(\boldsymbol{q})$.

In order to highlight the role of K_α, in Fig. 5.17 and Fig. 5.18 the total kinetic energy and its two contributions (i.e. the kinetic energy in the constraint space and nullspace) are shown for $K_\alpha = 0\,\mathrm{Nms^3/rad^3}$ and $K_\alpha = 30\,\mathrm{Nms^3/rad^3}$ respectively. As it can be noticed, in the second case the kinetic energy in the constraint space converges more rapidly to zero, while the one in the nullspace increases faster.

Table 8.1 collects the values of all the gains used in the simulation.

Before concluding the section, the control law proposed in section 5.3 is compared to a cascaded design [Ott08, Chapter 6], in which the desired torque is chosen as in section 4.3. The goal is to show that, in the latter case, similar performances can be achieved at the price of using considerably larger gains $\boldsymbol{K}_\tau$ and $\boldsymbol{D}_\tau$. Obviously, this can compromise the implementation on a

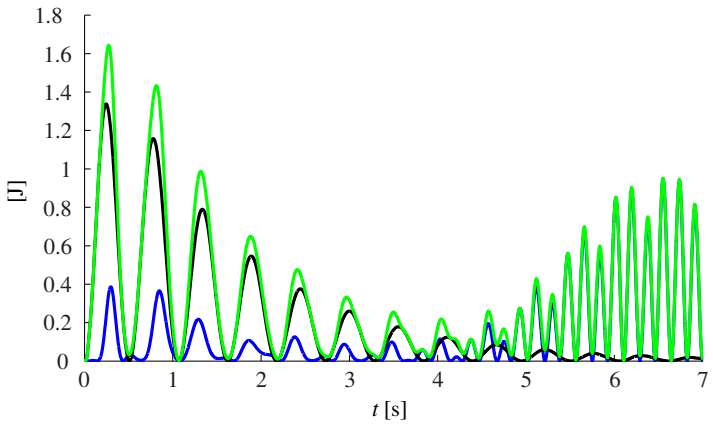

Figure 5.17.: Total kinetic energy (green line) and its two components, i.e. kinetic energy in the constraint space (black line) an in the nullspace (blue line), in case $K_\alpha = 0$.

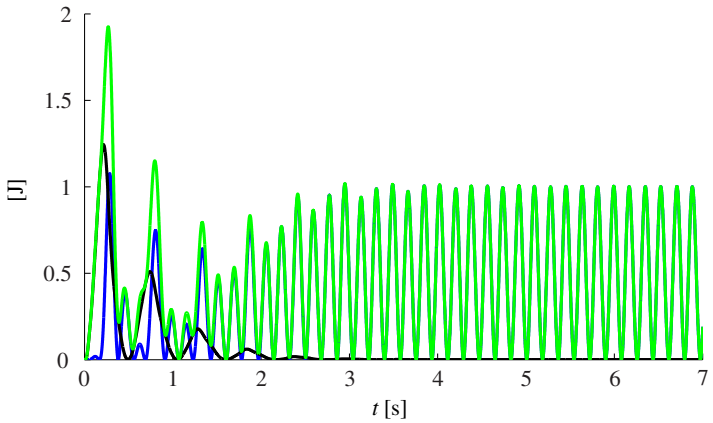

Figure 5.18.: Total kinetic energy (green line) and its two components, i.e. kinetic energy in the constraint space (black line) an in the nullspace (blue line), in case $K_\alpha = 30\,{}^{\mathrm{Nms}^3}/_{\mathrm{rad}^3}$.

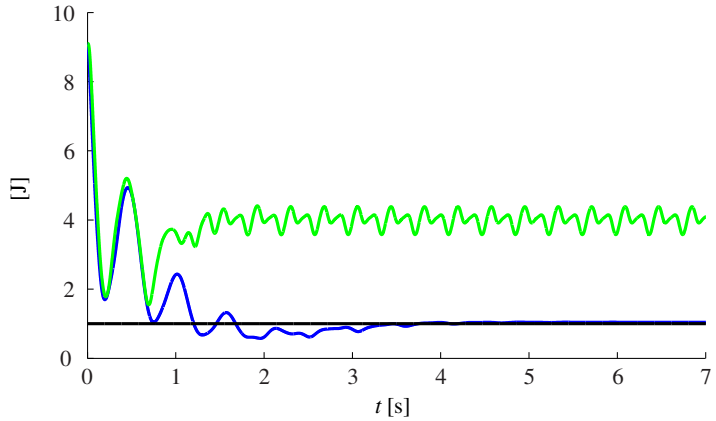

Figure 5.19.: Energy function (4.16) for $K_{\tau_i} = 10^4\,{}^1/_{s^2}, D_{\tau_i} = 160\,{}^1/_{s}$ (green line), $K_{\tau_i} = 10^8\,{}^1/_{s^2}, D_{\tau_i} = 16^3\,{}^1/_{s}$ (blue line) and desired value (black line).

real system. In order to obtain a fair comparison the virtual potential $U(\boldsymbol{q})$ for the control law in section 4.3 is chosen as

$$U(\boldsymbol{q}) = U_g(\boldsymbol{q}) + U_k(\boldsymbol{\eta}_d - \boldsymbol{q}) - U_g(\boldsymbol{q}_d) - U_k(\boldsymbol{\eta}_d - \boldsymbol{q}_d) \tag{5.41}$$

where $\boldsymbol{q}_d = \bar{\boldsymbol{q}}_y(\boldsymbol{\eta}_d)$. With this choice, $U(\boldsymbol{q})$ has a constrained minimum at $\boldsymbol{q}_d$ as it is required in section 4.3 and additionally the real potential of the system is used. The results, shown in Fig. 5.19, can be interpreted as follows: in the control law proposed in section 5.3, it is not necessary to track a desired torque to generate the limit cycle, but only to force the system to evolve on a submanifold. The latter is not varying as rapidly as the one necessary to produce the limit cycle, therefore smaller gains are sufficient. Additionally, the tracking of the desired torque in the cascaded design has a more direct effect on the regulation of the energy, since an imperfect tracking will directly result in the incapability for the system to reach the desired value of the energy.

Table 5.4.: Values of the gains. When only the i - th entry is shown, then the others have the same value.

K_{y_1}	D_{y_1}	K_{y_2}	D_{y_2}	
$400\,{}^{\mathrm{N}}/_{\mathrm{m}}$	$36\,{}^{\mathrm{N}}/_{\mathrm{ms}}$	$400\,{}^{\mathrm{Nm}}/_{\mathrm{rad}}$	$36\,{}^{\mathrm{Nms}}/_{\mathrm{rad}}$	
K_{τ_i}	D_{τ_i}	K_{η_i}	D_{η_i}	K_H
$1600\,{}^1/_{s^2}$	$64\,{}^1/_{s}$	$100\,{}^{\mathrm{Nm}}/_{\mathrm{rad}}$	$16\,{}^{\mathrm{Nms}}/_{\mathrm{rad}}$	$0.3\,{}^{s^2}/_{\mathrm{kgm}^2}$

5.5. Summary

In this chapter, new control laws for elastically actuated manipulators were proposed. First, a single joint system was examined. In this case, a static state feedback was designed to allow (by changing the desired value of an energy function) both regulation to a desired configuration of the robot and generation of an asymptotically stable limit cycle. When this control law is used for a multi-joints manipulator, only regulation to a desired configuration is possible. Therefore, a dynamic state feedback was designed, which allows to produce an asymptotically stable limit cycle also for a multi-joints manipulator. As the limit cycle is sustained, in both case, by the elastic elements present in the joints themselves, efficient performances are expected by both control laws. In the nominal case of frictionless systems, the motors stop once the energy reaches the desired value, so no power is provided by the motors. On the contrary, classic approaches, being designed to minimize the effects of the elasticity, aim at producing via the springs the torque disgned for the rigid case. Finally, since elastically actuated manipulators are underactuated mechanical systems, intuitively there are not enough control inputs to force the system to evolve on the submanifold and regulate at the same time the energy to the desired value, as it was done in the previous chapter for a rigidly actuated robot. Loosely speaking, introducing additional dynamics through the controller allowed to solve this problem. Finally, the stability analysis made extensive use of the semidefinite Lyapunov theory in order to reduce the system in subsystems easier to analyze.

CHAPTER 6

Jumping control

In this chapter, the complete underactuated problem is considered. The robot is neither fixed to the floor, nor rigidly actuated. A bipedal robot is used as case study (see Fig. 6.1). The only simplifying hypothesis that will be used throughout the chapter is to assume that the torque control loop is reacting fast enough so to not compromise the computations of the balancing part of the controller. In other words, the torques provided by the balancer guarantee that the robot does not fall, but the deviation of the torques produced by the springs from the desired ones computed by the balancing controller must be small enough during the transient not to cause the robot to lose balance. The assumption is reasonable since the spring torque is part of the state and cannot be instantaneously changed by any controller. As it has been mentioned, the control law will be composed of a "balancing part", whose goal is to guarantee the feasibility of the control law. Loosely speaking, in a first stage the input to the system will be designed assuming to have full actuation, i.e. including the wrench at the base. Next, the control forces computed in the previous stage are considered as desired values and the new control forces are obtained through an optimization problem aiming at keeping them as close as possible to the desired ones, while satisfying some opportune constraints. In this second stage, the requirement of having no actuation at the base is included through one of the constraints.

Although jumping robots are present on the robotic scene since a while (consider for example ASIMO [SWA+02]), during the years this task has been mainly achieved by quadrupedal robots for which, obviously, the balancing problem is simplified. Both bipedal and quadrupedal robots usually use simplified models or offline optimization in order to design force profiles and reference trajectories [SY05, PPK15]. Few of them include elastic elements in the legs [GCH+13], but it is not clear how the controller is actively using the springs beside benefit from the improved shock absorption at the impacts. These are exactly the points on which this chapter focuses.

The contribution presented here is a control law that directly uses the elasticity in the joints in order to produce a sequence of jumps, while considering the whole dynamic model of the robot. The resulting jumps are not pre-planned, but a dynamical result of the actions of the springs, whose torques are shaped trough the control action. To this end, the energy controller for elastically actuated robots presented in chapter 5 is combined with the balancing controller [HDO16]. As the latter makes use of the task hierarchy controller [DOAS13, ODAS15], both of them are shortly presented in the following.

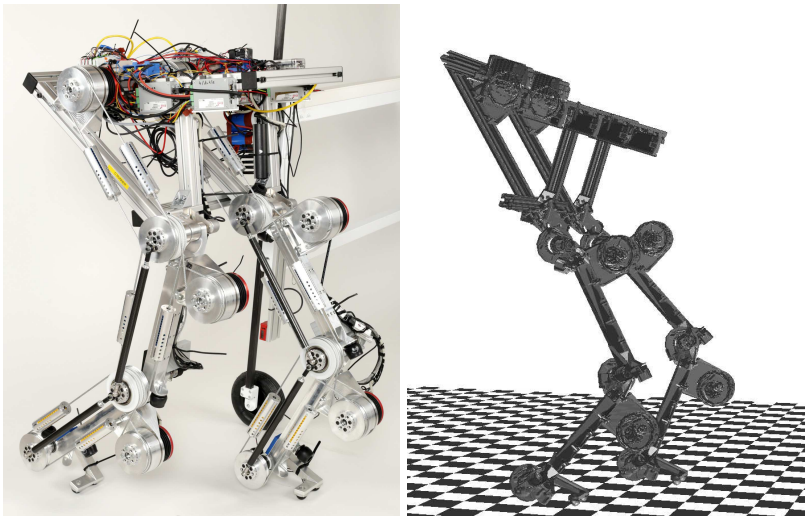

Figure 6.1.: C-Runner: test-bed (left) and simulated model in the initial configuration (right).

6.1. The task hierarchy controller in a nutshell

The task hierarchy controller introduced in [DOAS13, ODAS15] extends the results in [OKN08] in order to cope with the different priorities that the tasks of a fixed base manipulator with rigid joints can have. With an abuse of notation, let $\boldsymbol{y}(\boldsymbol{q})$ indicate the whole task coordinates, so that $\boldsymbol{y}(\boldsymbol{q}) \in \mathbb{R}^n$, and let $\boldsymbol{y}_i(\boldsymbol{q})$ be one of the subtasks. Given the mapping

$$\dot{\boldsymbol{y}}_i = \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad\qquad 1 \leq i \leq r\,, \tag{6.1}$$

assume additionally that each of the Jacobian matrix $\boldsymbol{J}_i(\boldsymbol{q}) \in \mathbb{R}^{m_i \times n}$ is full rank, as well as each Jacobian matrix obtained stacking any $\boldsymbol{J}_i(\boldsymbol{q})$, then a coordinate transformation similar to the one in section 4.2 can be considered. Indicating with $\bar{\boldsymbol{J}}_N(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ the new extended Jacobian matrix, then

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_1 \\ \vdots \\ \boldsymbol{\zeta}_r \end{bmatrix} = \bar{\boldsymbol{J}}_N(\boldsymbol{q})\dot{\boldsymbol{q}}\,, \tag{6.2}$$

where $\boldsymbol{\zeta}_1 = \dot{\boldsymbol{y}}_1$, while the others are nullspace velocities $\boldsymbol{\zeta}_i \in \mathbb{R}^{m_i}$. The latter are designed such that they represent the original task as closely as possible, but are inertially decoupled from the tasks with a higher priority level. This new extended Jacobian matrix is invertible, so that

$$\dot{\boldsymbol{q}} = \bar{\boldsymbol{J}}_N^{-1}(\boldsymbol{q})\boldsymbol{\zeta} = \sum_{i=1}^{r} \boldsymbol{Z}_i(\boldsymbol{q})\boldsymbol{\zeta}_i\,, \tag{6.3}$$

where $\boldsymbol{Z}_1(\boldsymbol{q}) = \boldsymbol{J}^{+_M}(\boldsymbol{q})$, while for $2 \leq i \leq r$

$$\boldsymbol{Z}_i(\boldsymbol{q}) = \boldsymbol{J}_i(\boldsymbol{q})\boldsymbol{M}^{-1}\boldsymbol{\Upsilon}_i(\boldsymbol{q})\,, \tag{6.4}$$

being $\boldsymbol{\Upsilon}_i(\boldsymbol{q})$ the classical nullspace projector [SSVO08] defined as

$$\boldsymbol{\Upsilon}_i(\boldsymbol{q}) = \boldsymbol{E}_n - \begin{bmatrix} \boldsymbol{J}_1 \\ \vdots \\ \boldsymbol{J}_{i-1} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{J}_1 \\ \vdots \\ \boldsymbol{J}_{i-1} \end{bmatrix}^{+_M T}\,, \tag{6.5}$$

where $\boldsymbol{E}_n$ is the identity matrix.

Notice that the case examined in section 4.2 can be retrieved setting $\boldsymbol{y}(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{y}_1^T & \cdots & \boldsymbol{y}_{r-1}^T\end{bmatrix}^T$ and given that $y_r \in \mathbb{R}$.

Using the new coordinate transformation the system can be written, similarly to (4.8), as

$$\boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\zeta} = \bar{\boldsymbol{J}}_N^{-T}(\boldsymbol{q})\Big(\boldsymbol{\tau} - \boldsymbol{g}(\boldsymbol{q})\Big)\,, \tag{6.6}$$

where this time the matrices $\boldsymbol{\Lambda}(\boldsymbol{q})$ and $\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})$ are still block diagonal, but with r entries on the main diagonal. The same stability property as in [OKN08] holds for the closed loop system obtained through the feedback control law

$$\boldsymbol{\tau} = \boldsymbol{g} + \bar{\boldsymbol{J}}_N^T \left(\bar{\boldsymbol{\Gamma}} \begin{bmatrix}\boldsymbol{\zeta}_1 \\ \vdots \\ \boldsymbol{\zeta}_r\end{bmatrix} + \begin{bmatrix}\boldsymbol{Z}_1 \boldsymbol{J}_1^T \boldsymbol{\phi}_1 \\ \vdots \\ \boldsymbol{Z}_r \boldsymbol{J}_r^T \boldsymbol{\phi}_r\end{bmatrix} \right)\,, \tag{6.7}$$

where $\bar{\boldsymbol{\Gamma}}$ is obtained from $\boldsymbol{\Gamma}$ setting the blocks on the main diagonal to zero and $\boldsymbol{\phi}_i$ is a PD term regulating the task coordinates $\boldsymbol{y}_i$ to their desired value $\boldsymbol{y}_{d_i}$. In addition to [OKN08], here the transient response is guaranteed to have no inertial coupling between the different tasks.

6.2. The balancing controller in a nutshell

In [HDO16] a balancing controller based on the task hierarchy controller was proposed for a humanoid robot. The goal is to have the CoM of the robot and the hip orientation (CoM task) in a given configuration in space. The joint configuration should also be as close as possible to a desired one (posture task). The CoM task together with the posture task completely define the configuration of the floating base robot. Due to the restrictions imposed to the system by the contact state, the posture task might not be fully fulfilled. Assuming to have some of the end effectors of the robot in contact with the environment and some free to move in space, the requirements are to produce balancing wrenches with the first (balancing task) and an impedance behavior with the latter (interaction task). The balancing wrenches have to counteract the effects of all the other task and compensate the gravity, in order to ensure that the base does not need to be actuated. The four tasks are organized in the task hierarchy as

- balancing task ($\boldsymbol{\phi}_1$)
- interaction task ($\boldsymbol{\phi}_2$)
- CoM task ($\boldsymbol{\phi}_3$)
- posture task ($\boldsymbol{\phi}_4$)

where all the $\boldsymbol{\phi}_i$ are chosen as PD terms, except for $\boldsymbol{\phi}_1$. The latter is determined through an optimization problem, which guarantees the feasibility of the control law, i.e. it makes sure that each balancing end effector does not lift off, slip or tilt, as well as ensuring that no forces at the floating base are commanded. In fact, the main difference compared to the task hierarchy controller in [DOAS13] is that the input $\boldsymbol{u}$ to the system cannot be fully chosen, i.e. $\boldsymbol{u} = \begin{bmatrix}\boldsymbol{0}^T & \boldsymbol{\tau}^T\end{bmatrix}^T$. The balancing controller in [HDO16] is designed using a frame attached to the CoM and orientation given by the hip frame, so that with this coordinates the gravity torque simplifies to $\begin{bmatrix}m\boldsymbol{g}_0 \\ \boldsymbol{0}\end{bmatrix}$. Using

a quasi-static argument the controller (6.7) becomes

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{g}_0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{\Upsilon}_2 \boldsymbol{J}_2^T & \boldsymbol{\Upsilon}_3 \boldsymbol{J}_3^T & \boldsymbol{\Upsilon}_4 \boldsymbol{J}_4^T \end{bmatrix} \boldsymbol{\phi} \,, \tag{6.8}$$

where the property $\bar{\boldsymbol{J}}_{N_i} \boldsymbol{Z}_i \boldsymbol{J}_i = \boldsymbol{\Upsilon}_i \boldsymbol{J}_i$ [DOAS13] has been used and all the forces have been stacked in $\boldsymbol{\phi}$. The optimization problem can then be formulated as

$$\min_{\boldsymbol{\phi}_1} \left(\boldsymbol{\phi}_1 - \boldsymbol{\phi}_{d_1} \right)^T \boldsymbol{W} \left(\boldsymbol{\phi}_1 - \boldsymbol{\phi}_{d_1} \right) \tag{6.9a}$$

$$\text{s.t. } m\boldsymbol{g}_0 + \boldsymbol{\Xi}_u \boldsymbol{\phi} = \mathbf{0} \tag{6.9b}$$

$$f_{min,i} \leq f_{i,\perp} \tag{6.9c}$$

$$\left\| \boldsymbol{f}_{i,\parallel} \right\| \leq \mu_i f_{i,\perp} \tag{6.9d}$$

$$CoP_i(\boldsymbol{f}_i) \in \mathscr{S}_i \tag{6.9e}$$

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\Xi}_a \boldsymbol{\phi} \leq \boldsymbol{\tau}_{max} \,, \tag{6.9f}$$

where the cost function minimizes the deviation from a default wrench distribution $\boldsymbol{\phi}_{d_i}$ with $\boldsymbol{W}$ a symmetric positive definite weighting matrix. The equality constraint is derived from the first equation in (6.8), where

$$\begin{bmatrix} \boldsymbol{\Xi}_u \\ \boldsymbol{\Xi}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_1^T & \boldsymbol{\Upsilon}_2 \boldsymbol{J}_2^T & \boldsymbol{\Upsilon}_3 \boldsymbol{J}_3^T & \boldsymbol{\Upsilon}_4 \boldsymbol{J}_4^T \end{bmatrix} \,, \tag{6.10}$$

from which it also follows

$$\boldsymbol{\tau} = \boldsymbol{\Xi}_a \boldsymbol{\phi} \,, \tag{6.11}$$

that is the control law applied to the robot. The last inequality constraint is, therefore, ensuring that the resulting joint torques stay within the limitations of the hardware. Finally, the other constraints take into account the contact model. There, $f_{i,\perp}$ and $\boldsymbol{f}_{i,\parallel}$ denote the components of the contact force $\boldsymbol{f}_i$ perpendicular and parallel to the contact surface $\mathscr{S}_i$. The unilaterality of the contact is taken into account by limiting the normal component to $f_{min,i} \geq 0$. To prevent the end effector from slipping, $\boldsymbol{f}_{i,\parallel}$ is limited via the friction coefficient μ_i, while tilting is avoided restricting the center of pressure (CoP) of each end effector to $\mathscr{S}_i$.

6.3. A toy model for jumping

The controllers designed in chapter 4 and chapter 5 have been proved to produce an asymptotically stable limit cycle for a robotic manipulator. The results have been validated in simulations and experiments (see also chapter 8). Although not formally proven in the stability analysis, the experiments show that periodicity is achieved even in case of oscillations around the desired value of the energy, due to unmodeled disturbances. On the other hand, in chapter 3, it was shown that a form of energy regulation was able to produce a periodic walking pattern for a simulated biped robot with impacts. The idea behind the use of energy regulation for a jumping robot can be exemplified using a modified 1 - dimensional SLIP model, i.e. the simplest scenario in which the elasticity and the energy regulation can be combined to obtain a jumping behavior. The system is represented by a single massless spring connected to a mass constrained to move along the vertical direction. The model of the system is:

$$\begin{cases} \ddot{y} + g = -K_h \tilde{H} \dot{y} - k\left(y - l_0\right) & y \leq l_0 \\ \ddot{y} + g = 0 & y > l_0 \end{cases} \,, \tag{6.12}$$

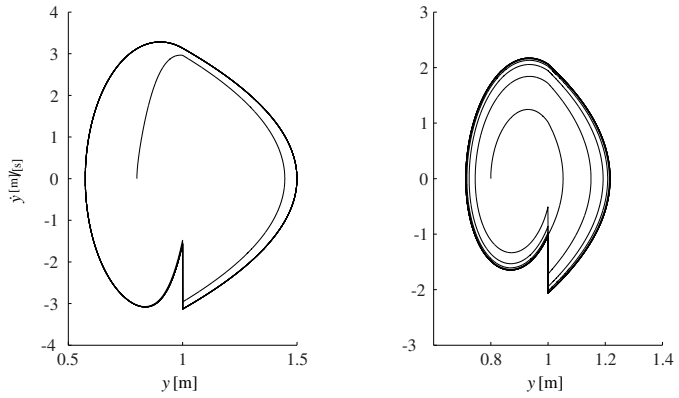

Figure 6.2.: Phase portrait of the toy model for $K_h = 5\,\mathrm{s/kgm^2}$ (left) and $K_h = 0.5\,\mathrm{s/kgm^2}$ (right).

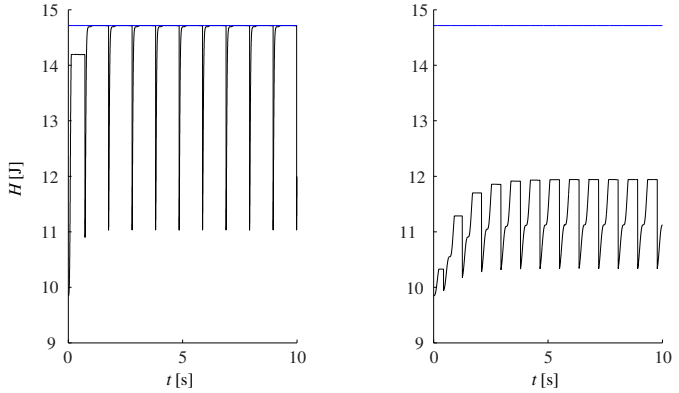

Figure 6.3.: Total energy and desired value for $K_h = 5\,\mathrm{s/kgm^2}$ (left) and $K_h = 0.5\,\mathrm{s/kgm^2}$ (right).

where H is the total energy of the system, l_0 the rest length of the spring, k the stiffness constant and g the gravitational acceleration. In addition, it is assumed that the velocity before and after the impact are related by a certain coefficient of restitution α, i.e. $\dot{y}^+ = \alpha \dot{y}^-$.

Depending on the values of α and K_h, the different phase portraits are reported in Fig. 6.2. Firstly, in both cases, the strong asymmetry with respect to x is due to the switching of the dynamics. The discontinuity, instead, is due to the impact. Finally, while with a high K_h (left) the system is able to reach the desired value of energy before the next take off, this is not the case with a small value of K_h (right). Nevertheless, a limit cycle is reached for which the energy lost at the impact balances the one injected before the take off. The energy for the two cases is plotted in Fig. 6.3.

Controlling the energy is therefore not only beneficial to exploit the presence of the elastic joints, but it also provides some kind of robustness against the interaction with the floor. In this simple example, when the energy is quickly regulated to the desired value, the latter is also directly responsible for the jumping height. In Fig. 6.4, the actual and desired height are plotted, with the latter obtained from the desired energy value. For a given energy loss, the maximum reachable height is limited by the speed at which the energy can be reinjected in the system.

6.4. The jumping controller

The main idea behind the jumping controller is to use the regulation of the energy to produce the desired oscillatory behavior of a jumping motion, but without taking into account any specific

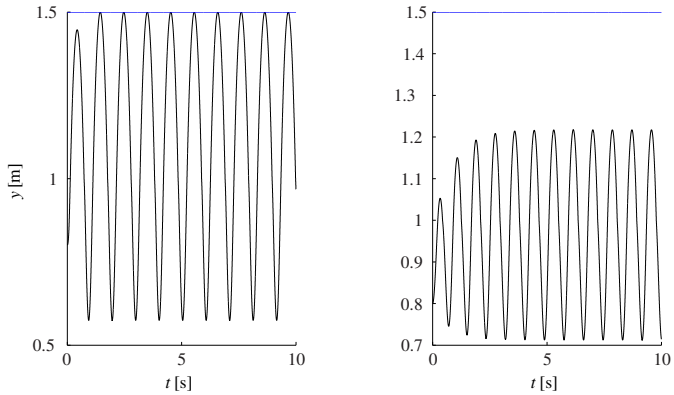

Figure 6.4.: Jumping height for $K_h = 5\,\mathrm{s/kgm^2}$ (left) and $K_h = 0.5\,\mathrm{s/kgm^2}$ (right).

limitations due to the contact model, similarly to what was done in the previous section. The latter will be ensured by an optimization problem as in the balancing controller, which redistributes ϕ in order to have feasible contact forces.

As a first step, a dynamic state feedback controller as in section 5.3 is used for the robot, leading to

$$\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{Q}^T\boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big) + \boldsymbol{Q}^T\boldsymbol{\tau}_d + \boldsymbol{Q}^T\tilde{\boldsymbol{\tau}} + \boldsymbol{J}_f^T(\boldsymbol{x})\boldsymbol{w}_f \tag{6.13a}$$

$$\ddot{\tilde{\boldsymbol{\tau}}} + \boldsymbol{D}_\tau\dot{\tilde{\boldsymbol{\tau}}} + \Big(\boldsymbol{K}_\tau + \boldsymbol{K}\boldsymbol{B}^{-1}\Big)\tilde{\boldsymbol{\tau}} = \boldsymbol{0} \tag{6.13b}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + K_H\tilde{H}\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) + \boldsymbol{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta\tilde{\boldsymbol{\eta}} = \boldsymbol{0}\,, \tag{6.13c}$$

where, the input function $\boldsymbol{\tau}_d$ will be designed through an optimization based approach similar to the one used in the balancing controller [HDO16]. The details of this optimization problem will be presented later on. For now assume that $\boldsymbol{\tau}_d$ has been computed and consider the system conditionally to the convergence of the torque error. Additionally, during the design phase external wrenches are not taken into account. Therefore, the system to consider for the design of $\boldsymbol{\tau}_d$ is

$$\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{x},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}'(\boldsymbol{x}) = \boldsymbol{u}' \tag{6.14a}$$

$$\boldsymbol{B}\ddot{\boldsymbol{\eta}} + K_H\tilde{H}\boldsymbol{K}\Big(\bar{\boldsymbol{q}}_y(\boldsymbol{\eta}) - \boldsymbol{q}\Big) + \boldsymbol{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta\tilde{\boldsymbol{\eta}} = \boldsymbol{0}\,, \tag{6.14b}$$

where $\boldsymbol{u}' = \begin{bmatrix}\boldsymbol{0}^T & \boldsymbol{\tau}_d^T\end{bmatrix}^T$ and

$$\boldsymbol{g}'(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) - \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{K}\Big(\boldsymbol{\eta} - \boldsymbol{q}\Big)\end{bmatrix}\,. \tag{6.15}$$

Similarly to (5.28), $\boldsymbol{u}'$ is chosen as

$$\boldsymbol{u}' = \bar{\boldsymbol{J}}_N^T\left(\bar{\boldsymbol{\Gamma}}\boldsymbol{\zeta} + \begin{bmatrix}\boldsymbol{Z}_1\Big(\boldsymbol{g}' + \boldsymbol{J}_1^T\boldsymbol{\phi}_{d_1}\Big)\\ \vdots \\ \boldsymbol{Z}_{r-1}\Big(\boldsymbol{g}' + \boldsymbol{J}_{r-1}^T\boldsymbol{\phi}_{d_{r-1}}\Big)\\ 0\end{bmatrix}\right)\,, \tag{6.16}$$

where $\boldsymbol{\phi}_{d_i}$ is a PD term regulating the task coordinates $\boldsymbol{y}_i$ to their desired value $\boldsymbol{y}_{d_i}$, both still to be defined in order to make the robot jump. For the last task, $\boldsymbol{\phi}_{d_r} = 0$ since the projection

of the spring and gravity torque (which have not been compensated) will be responsible for the oscillatory behavior. In order to facilitate the derivation of the controller input function $\boldsymbol{\tau}_d$, it is possible to rewrite $\boldsymbol{u}'$ as

$$\boldsymbol{u}' = \begin{bmatrix} \boldsymbol{\Xi}_u \\ \boldsymbol{\Xi}_a \end{bmatrix} \boldsymbol{\phi}_d + \begin{bmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_a \end{bmatrix} , \tag{6.17}$$

with

$$\begin{bmatrix} \boldsymbol{\Xi}_u \\ \boldsymbol{\Xi}_a \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{J}}_{N_1}^T \boldsymbol{Z}_1 \boldsymbol{J}_1^T & \cdots & \bar{\boldsymbol{J}}_{N_r}^T \boldsymbol{Z}_r \boldsymbol{J}_r^T \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_a \end{bmatrix} = \bar{\boldsymbol{J}}_N^T \left(\bar{\boldsymbol{\Gamma}} \boldsymbol{\zeta} + \begin{bmatrix} \boldsymbol{Z}_1 \boldsymbol{g}' \\ \vdots \\ \boldsymbol{Z}_{r-1} \boldsymbol{g}' \\ 0 \end{bmatrix} \right) . \tag{6.18}$$

Finally, $\boldsymbol{\tau}_d$ can be obtained solving the following optimization problem

$$\min_{\boldsymbol{\phi}} \left(\boldsymbol{\phi} - \boldsymbol{\phi}_d \right)^T \boldsymbol{W} \left(\boldsymbol{\phi} - \boldsymbol{\phi}_d \right) \tag{6.19a}$$

$$\text{s.t. } \boldsymbol{\Xi}_u \boldsymbol{\phi} + \boldsymbol{b}_u = \boldsymbol{0} \tag{6.19b}$$

$$f_{min,i} \leq f_{i,\perp} \tag{6.19c}$$

$$\left\| \boldsymbol{f}_{i,\parallel} \right\| \leq \mu_i f_{i,\perp} \tag{6.19d}$$

$$CoP_i(\boldsymbol{f}_i) \in \mathscr{S}_i \tag{6.19e}$$

$$\boldsymbol{\tau}_{min} - \boldsymbol{b}_a \leq \boldsymbol{\Xi}_a \boldsymbol{\phi} \leq \boldsymbol{\tau}_{max} - \boldsymbol{b}_a \tag{6.19f}$$

and setting $\boldsymbol{\tau}_d = \boldsymbol{\Xi}_a \boldsymbol{\phi}^* + \boldsymbol{b}_a$, where $\boldsymbol{\phi}^*$ is the optimal solution.

Two aspects still need to be considered: the definition of the constraints and the treatment of the flight phase. The exact definition of the constraints is strongly related to the robotic system under consideration and therefore in the next subsection the case of the robot in Fig. 6.1 is considered. Nevertheless, the choice is always based on similar tasks as in section 6.2, i.e. a task for the end effectors in contact with the environment, one for those free to move, one for the CoM and a posture task. Concerning the flight phase, it is important to remember that the CoM dynamics cannot be influenced by the motors until the next touch down. Similarly, not all the velocity coordinates can be freely influenced since they have to satisfy the conservation of the angular momentum. The strategy consists, therefore, in switching the weights of the different tasks used in the cost function. Depending on the contact situation, a bigger deviation from the desired values is allowed for one or more tasks in order to satisfy the constraints of the optimization problem.

6.4.1. The C-Runner

The C-Runner, Fig. 6.1, is a two-legged planar robot with linear series elastic actuation. Both legs are attached to a common trunk and all the segments are connected by revolute joints. The trunk, thighs and shanks carry the electric servomotors which actuate through a cable drive transmission system the correspondent joint. The drive axis of the motor and the one of the joint are, therefore, not coincident. The system is scaled to approximately match the dimensions of a 1.65 m tall human. Table 6.1 shows the segment lengths and masses.

The robot is additionally equipped with torque and position sensors to provide a full state measurement. Also, a ground force sensor is integrated into each of the two silicone half domes representing the contact points of the foot. They give the possibility to roll over during heel strike or ball push off. The silicone domes provide elasticity, damping and a high coefficient of friction. This results in shock absorption and improved traction.

The trunk is connected to a boom, which constrains the robot to move in the sagittal plane. Therefore, the link-side configuration of the robot is defined by nine parameters, three for the trunk and one for each joint. The first task defines the pose of the feet (6 parameters), the second the horizontal position of the CoM and the trunk orientation (2 parameters), the third and last one the vertical position of the CoM (1 parameter). A unique link-side configuration exists, which fulfills all the tasks.

Task one: feet pose The feet pose task is responsible for keeping the feet on the floor during the contact phase and to reach the landing pose during the flight phase. Inspired by the results of chapter 3, a constant offset from the CoM can be used for the desired feet position, while the desired orientation is parallel to the floor. The weights of this task are the lowest during the contact phase and increase during the flight phase. In this way, during the contact phase, the feet are mainly used to provide the necessary reaction forces by pushing in the floor. Moreover, note that since the unilateral constraints in the optimization only allow the feet to press against the floor, the feet will not lift even if the desired position is above the ground. This task is equivalent to the balancing task in [HDO16] when there is a contact, or the interaction task when the foot is above the floor.

Task two: CoM horizontal position and trunk orientation While in [HDO16] the CoM task was taking into account also the vertical position of the CoM, in this case the latter is used to produce the oscillatory behavior and therefore removed from the CoM task. The horizontal position of the CoM is chosen to be in the middle of the support polygon, in order to maximize the stability margins during static balancing. The weights of this tasks are the highest during the contact phase, as it is fundamental to prevent the robot from falling. During the flight phase, instead, the weights are lowered since only the relative position between the CoM and the feet can be influenced, which is already tackled by the previous task. Finally, the trunk orientation is equivalent to the posture task in [HDO16], since it is the only part of the link-side configuration space left free by the other tasks.

Task three: CoM vertical position As in the last task the oscillatory behavior will be produced, clearly using the CoM height is a good choice for achieving a jump. Assuming that all the other tasks are perfectly satisfied, it means that the floor will exert a net force on the robot which is in line with the weight. This is easily justified by the fact that a movement of the CoM along the horizontal position would be otherwise produced. Therefore, the vertical oscillation will continue until the energy is high enough to produce the lift off.

A graphical representation of how the weights are updated can be found in Fig. 6.5. Notice that while the number associated to the task defines how they are organized in the task hierarchy, i.e. defines the inertial decoupling of the tasks, the weights define which forces the optimization is more likely to modify from the desired values in order to satisfy the constraints of the problem.

Table 6.1.: Segment lengths and masses.

Segment	Trunk	Thigh	Shank	Foot	Total
Length [m]	0.276	0.40	0.40	0.07	1.15
Mass [kg]	24.96	5.46	5.42	1.31	49.34

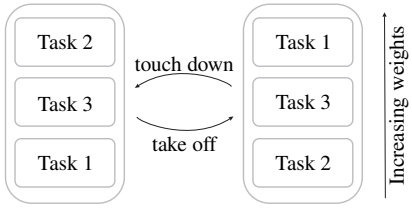

Figure 6.5.: The weights in the cost function are switched depending on the contact phase.

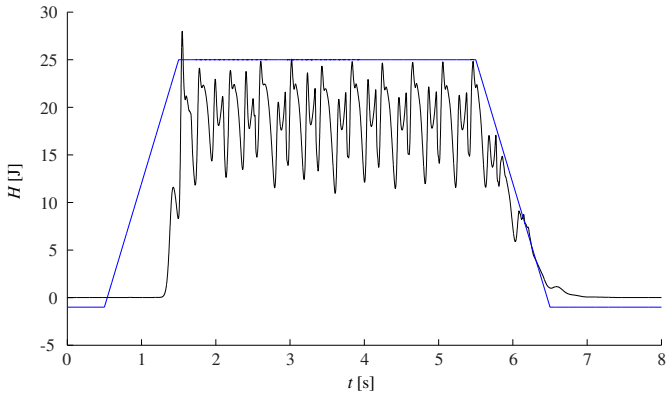

Figure 6.6.: Desired energy (blue) and actual value (black).

6.4.2. Evaluation

The proposed control law is evaluated in simulation. An elastic contact is used to model the interaction with the floor. Notice that, as a small non constant deflection of the floor will always be present, this is a challenging condition for the controller. The contact is detected by the controller when the vertical component of the reaction force is bigger than 4 N. Moreover, in order to match the implementation of the real system, the torque control loop is replaced by a motor position loop with high gains ($\boldsymbol{K}_P = 8000\,\mathrm{Nm/rad}$, $\boldsymbol{K}_D = 250\,\mathrm{Nms/rad}$), as at the current stage the drive unit only allows to control the motor in position. This means that the torque $\boldsymbol{\tau}_d$, is translated into a desired position $\boldsymbol{\theta}_d$ by inverting the joint stiffness (set to $\boldsymbol{K} = 260\,\mathrm{Nm/rad}$), i.e. $\boldsymbol{\theta}_d = \boldsymbol{q} + \boldsymbol{K}^{-1}\boldsymbol{\tau}_d$. The values $\boldsymbol{K}_x$ for each PD term are listed in Table 6.2, while the correspondent coefficient of the derivative action was always chosen using a damping ratio of 0.7.

Table 6.2.: Coefficients for the proportional action of the PD terms.

	Task one	Task two
Linear	$625\,\mathrm{N/m}$	$2025\,\mathrm{N/m}$
Angular	$4\,\mathrm{Nm/rad}$	$100\,\mathrm{Nm/rad}$

Starting from the initial configuration shown in Fig. 6.1, corresponding to the equilibrium, the desired energy is ramped up and then down as shown in Fig. 6.6. The plot also shows the real value of the energy. Notice how, although a negative value is asked for the energy, the minimum reached value is zero. The computation of $\bar{\boldsymbol{q}}_y(\boldsymbol{\eta})$, as mentioned in section 5.3.3, is performed using the current value of $\boldsymbol{y}(\boldsymbol{q})$, which ensures that there are no offsets in the energy due to the interaction with the floor.

Concerning the evolution of the energy, an imperfect convergence to the desired value was expected if for no other than the presence of the impacts, as seen in section 6.3. It should be

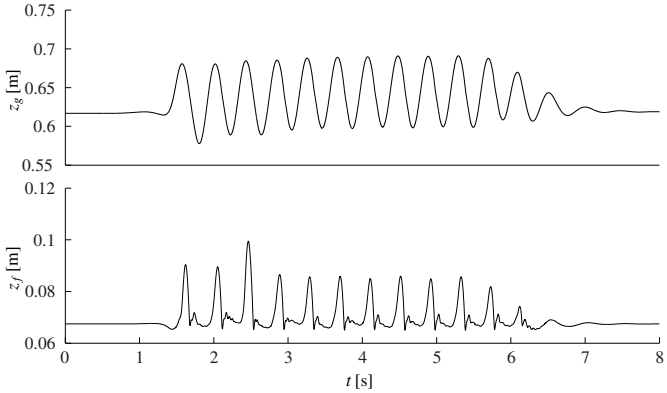

Figure 6.7.: Vertical position z_g of the CoM and height z_f of one of the feet.

mentioned that there is no special requirement of reaching the exact desired energy, as long as the desired jumping behavior is obtained. If a higher jumping height is requested, one might increase the desired value of energy and/or the gain K_h, within the limits set by the hardware.

The resulting jumping height of the CoM is reported together with the height of one of the two feet in Fig. 6.7. As it can be noticed, the robot starts from the equilibrium configuration, then the jumps are performed according to the increased desired energy and finally stops again when the energy goes back to zero.

Finally, in Fig. 6.8 one can see three entries of the input and output of the optimization problem. All the desired values of the PD terms are well tracked when the robot is at a rest or making small oscillations, except for the vertical force of the foot. The mismatch is actually the sign that the optimization is working properly, although one might think exactly the opposite. When the robot is at rest on the floor, the desired PD term is zero since the foot is exactly stopped at the required height. Nevertheless, the optimization "knows" that the gravity needs to be compensated and, as the foot task has a low weight in the cost function, commands the robot to push into the floor. On the other hand, when the robot starts jumping, in all the plots there are phases with big deviations. This is again a consequence of the constraints; in particular the PD terms for the trunk orientation and the CoM horizontal position (bottom two plots in the figure) are well tracked only during the contact phase, while the one for the height of the foot (top plot in the figure) is tracked better during the flight phase. This is in accordance with how the weights are chosen in Fig. 6.5. The touch down events can be easily recognized by the jumps in the plots.

6.4.3. Discussion and limitations

The main differences between the control law presented in this chapter and the ones in chapter 5 consist in the presence of the reaction wrench due to the interaction with the floor and the use of the optimization problem to guarantee the feasibility of the control action. The latter, although necessary from a practical point of view, is clearly limiting the possibility of considering a stability analysis. A possibility to still ensure the stability of the system might be to only scale the desired forces without changing their direction and orientation, although this increases the complexity of the optimization problem itself. Nevertheless, the use of optimization has been proven to be effective in practise and it is the state of the art for legged robotics (see section 1.1). If the result of the optimization is close enough to the desired values and at least for one foot the contact with the floor can be considered rigid with good approximation, similar passivity arguments as in [DOAS13] hold.

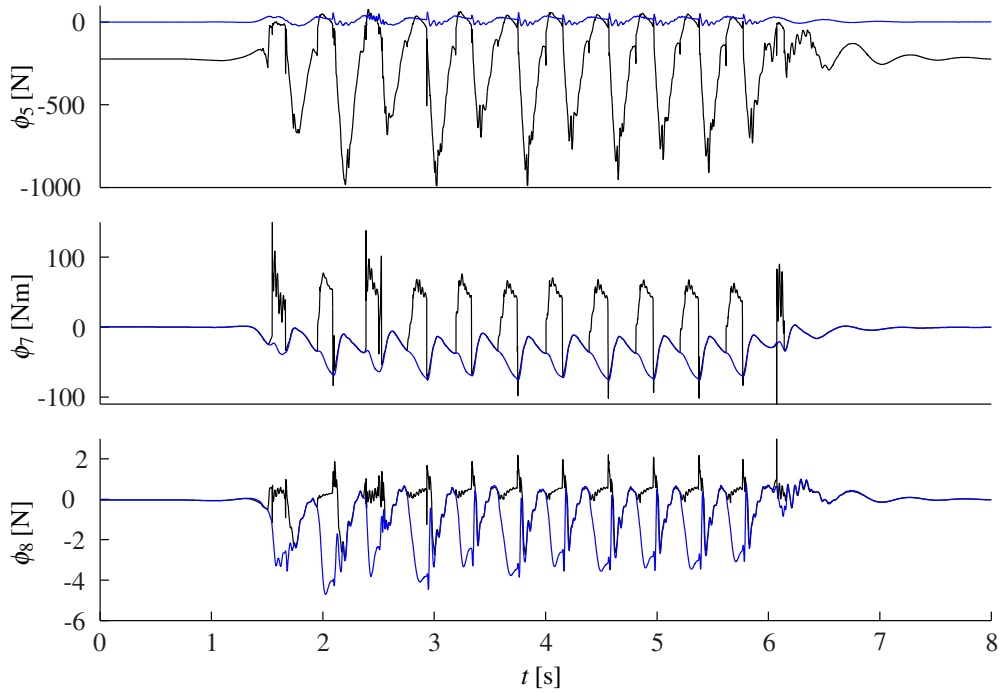

Figure 6.8.: Values of ϕ for the height of one foot, the orientation of the trunk and the horizontal position of the CoM (from top to bottom). The desired values are in blue and the real in black.

Finally, the jumping controller can be used as core for developing additional locomotion patterns. Crucial in these cases becomes the role of planning and the presence of a higher level supervisor. The position of the feet not in contact with the floor is, in fact, paramount for a successful stride. Chapter 7 provides some preliminary results for this aspect. At the same time, a state machine has to orchestrate the sequence of different phases, i.e. double support, single support or flight phase. This topic has not been covered in this work. In the current state, only the horizontal position of the CoM relative to the feet is controlled and, therefore, the robot can change position with respect to the inertial frame.

6.5. Summary

In this chapter, the energy controller for elastically actuated manipulators presented in chapter 5 was incorporated in an optimization framework (originally thought for balancing porpoises), in order to realize a jumping pattern. A toy model was used to motivate the use of energy regulation to achieved the desired jumps, showing that energy regulation is not beneficial just to exploit the presence of the elastic joints. The controller was evaluated in simulation using a planar biped robot, which shows the effectiveness of the controller and a behavior similar to the toy example.

CHAPTER 7

Improving the flight control

The content of this chapter is devoted to provide directions for designing ad hoc controllers to be used during the flight phase. As a consequence the considered robot model is the one of a rigidly actuated robot with a free floting base and no contacts. This kind of scenario is typical in space robotics. In that field, the results presented in [UY87, UY89], where the generalized Jacobian matrix was introduced for the first time, are often used. The idea is to replace the floating base velocity in the computation of the end-effector Jacobian matrix with the generalized momentum of the system which, in case of space operations, is often zero or at least constant. This allows to obtain a simpler expression, i.e. the generalized Jacobian matrix. Therefore, controller based on the generalized Jacobian matrix rely on the assumption of no generalized momentum in the system, which is easily fulfilled in space operation but unrealistic for humanoid robots due to the gravity. On the other hand, even during space operations momentum is accumulated in on-orbit servicing scenarios where contacts are involved. Additionally, typically humanoid robots are also redundant for the prescribed task. Strategies to cope with the presence of both a varying momentum and a nullspace need, therefore, to be considered. Although the proposed controller takes explicitly into account the dynamics of the generalized momentum, inevitable limitations are present due to the impossibility to influence such dynamics. The presence of dynamic singularities is another limiting factor. Unlike for the fixed based case, these singularities are path dependent [PD93] and further complicate the trajectory planning phase.

The controller can be applied, for example, to guarantee the convergence of the feet to the desired poses before the next impact with the ground or in a space mission where a robot manipulator is attached to a satellite in order to perform some manipulation task. The latter case is considered in a simulation study in section 7.3 and depicted in Fig 7.1. The contribution of this chapter is the design of a control law for regulating the end effector of a floating based robot in presence of nonzero generalized momentum. Although the controller is considered for a satellite with a robotic arm, it can be seen as a first result towards improved control of the feet of a humanoid robot during the flight phase. A satellite, in fact, shares the same structure of the dynamic model as a humanoid robot, but the absence of gravity simplifies the problem.

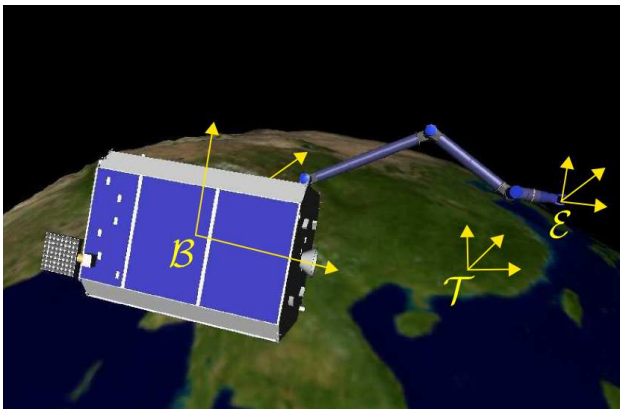

Figure 7.1.: Scenario and reference frames.

7.1. The generalized Jacobian matrix

Since the generalized Jacobian matrix relies on the conservation of the momentum for its derivation, it is clear that using the transformation presented in section 2.2.2 can easily lead to its expression. That formulation, in fact, implicitly contains the concept of generalized Jacobian matrix. Let $\boldsymbol{\nu}_e$ be the end-effector body twist and $\boldsymbol{J}_e(\boldsymbol{x})$ the associated Jacobian matrix, so that using (2.33) it follows

$$\boldsymbol{\nu}_e = \boldsymbol{J}_e(\boldsymbol{x})\boldsymbol{v} = \boldsymbol{J}_e(\boldsymbol{x})\boldsymbol{A}_Q^{-1}(\boldsymbol{x})\boldsymbol{\xi} \,,$$

expressed in terms of both the old and new coordinates. If $\boldsymbol{h} = \boldsymbol{0}$, taking into account the expression of $\boldsymbol{A}_Q^{-1}(\boldsymbol{x})$, the previous expression simplifies to

$$\boldsymbol{\nu}_e = \breve{\boldsymbol{J}}_e(\boldsymbol{x})\dot{\boldsymbol{q}} \qquad\qquad \breve{\boldsymbol{J}}_e(\boldsymbol{x}) = \boldsymbol{J}_e(\boldsymbol{x})\boldsymbol{Q}^{+_M}(\boldsymbol{x}) \,.$$

Partitioning the inertia matrix and the end-effector Jacobian matrix to separate the role of the floating base from the joints, i.e.

$$\boldsymbol{M}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{M}_b(\boldsymbol{x}) & \boldsymbol{M}_c(\boldsymbol{x}) \\ \boldsymbol{M}_c^T(\boldsymbol{x}) & \boldsymbol{M}_q(\boldsymbol{x}) \end{bmatrix} \qquad\qquad \boldsymbol{J}_e(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{J}_{eb}(\boldsymbol{x}) & \boldsymbol{J}_{eq}(\boldsymbol{x}) \end{bmatrix} \,,$$

the following expressions are obtained by direct computation

$$\begin{aligned} \boldsymbol{Q}^{+_M}(\boldsymbol{x}) &= \begin{bmatrix} -\boldsymbol{M}_b^{-1}(\boldsymbol{x})\boldsymbol{M}_c(\boldsymbol{x}) \\ \boldsymbol{E}_{n_q} \end{bmatrix} \\ \breve{\boldsymbol{J}}_e(\boldsymbol{x}) &= \boldsymbol{J}_{eq}(\boldsymbol{x}) - \boldsymbol{J}_{eb}(\boldsymbol{x})\boldsymbol{M}_b^{-1}(\boldsymbol{x})\boldsymbol{M}_c(\boldsymbol{x}) \,, \end{aligned} \tag{7.1}$$

where $\boldsymbol{E}_{n_q}$ is the $n_q \times n_q$ identity matrix and (7.1) is the well know definition of generalized Jacobian matrix introduced in [UY87, UY89].

7.2. Momentum decoupling control

The regulation problem of the end - effector of a free floating base robot is addressed, although the control law is not aiming at providing a solution for an indefinitely long flight phase, as it will be made clearer in section 7.2.4. Throughout the derivation of the controller it will be assumed that:

Assumption 7.1. *The generalized Jacobian matrix $\breve{\boldsymbol{J}}_e$ has no singularities for $\boldsymbol{q} \in \mathbb{Q}$.*

The design of the controller also relies on the coordinate transformation presented in the following.

7.2.1. Coordinates transformation

The generalized momentum $\boldsymbol{h}$, as it was introduced in section 2.2.2, can be expressed in any frame once it has been transformed by the appropriate adjoint matrix. In particular, using the partitioning of the matrices in section 7.1, the generalized momentum can be factorized as

$$\boldsymbol{h} = \boldsymbol{A}(\tilde{\boldsymbol{x}}, \boldsymbol{q})\boldsymbol{v} = \mathrm{Ad}_{c,e}^{-T}(\tilde{\boldsymbol{x}})\boldsymbol{J}_{eb}^{-T}\begin{bmatrix}\boldsymbol{M}_b & \boldsymbol{M}_c\end{bmatrix}\boldsymbol{v}\,, \tag{7.2}$$

where $\tilde{\boldsymbol{x}}$ is a local set of coordinates [Ott08] for the position $\boldsymbol{p}_{c,e}$ of the origin of the end effector frame e expressed in the CoM frame and the rotation matrix $\boldsymbol{R}_{c,e}$ of the end effector frame w.r.t. the CoM frame (i.e. a frame attached at the CoM with orientation given by the spatial frame).

Consider first the case of a not redundant manipulator attached to a floating base. The matrices $\boldsymbol{A}$ and $\boldsymbol{J}_e$ can be used to perform a change of coordinates that, as in section 2.2.2, highlights the dynamics of the generalized momentum. Assuming to have no singular configurations and using

$$\begin{bmatrix}\boldsymbol{h}\\ \boldsymbol{\nu}_e\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{J}_e\end{bmatrix}\boldsymbol{v} = \boldsymbol{\Psi}\boldsymbol{v}\,, \tag{7.3}$$

the coupled end effector/momentum dynamics is

$$\begin{bmatrix}\boldsymbol{\Lambda}_h & \boldsymbol{\Lambda}_{he}\\ \boldsymbol{\Lambda}_{he}^T & \boldsymbol{\Lambda}_e\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{h}}\\ \dot{\boldsymbol{\nu}}_e\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Gamma}_h & \boldsymbol{\Gamma}_{he}\\ \boldsymbol{\Gamma}_{eh} & \boldsymbol{\Gamma}_e\end{bmatrix}\begin{bmatrix}\boldsymbol{h}\\ \boldsymbol{\nu}_e\end{bmatrix} + \begin{bmatrix}\boldsymbol{\gamma}_h\\ \boldsymbol{\gamma}_e\end{bmatrix} = \boldsymbol{\Psi}^{-T}\boldsymbol{Q}^T\boldsymbol{\tau}\,. \tag{7.4}$$

In section 2.2.2 the joint dynamics was shown to be inertially decoupled from the generalized momentum, while in (7.4) the end effector/momentum dynamics is fully coupled. However, it can be proven that inserting the second equation of (7.4) into the first, the momentum equation results in $\dot{\boldsymbol{h}} = \boldsymbol{0}$. This is in agreement with the physical fact that any internal forces $\boldsymbol{\tau}$ cannot cause an increase in the generalized momentum of the total system.

In the more general case of a redundant manipulator, a nullspace velocity $\boldsymbol{v}_n$ can be introduced in order to get a square transformation matrix $\boldsymbol{\Psi}_N$:

$$\begin{bmatrix}\boldsymbol{h}\\ \boldsymbol{\nu}_e\\ \boldsymbol{v}_n\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Psi}\\ \boldsymbol{N}\end{bmatrix}\boldsymbol{v} = \boldsymbol{\Psi}_N\boldsymbol{v}\,, \tag{7.5}$$

where $\boldsymbol{N}$ is a dynamically consistent nullspace matrix defined using a nullspace base matrix $\boldsymbol{Z}$. Partitioning the latter as $\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{Z}_b & \boldsymbol{Z}_q\end{bmatrix}$, from the condition $\boldsymbol{\Psi}\boldsymbol{Z}^T = \boldsymbol{0}$ follows that

$$\begin{bmatrix}\mathrm{Ad}_{c,e}^{-T}\boldsymbol{J}_{eb}^{-T}\boldsymbol{M}_b & \mathrm{Ad}_{c,e}^{-T}\boldsymbol{J}_{eb}^{-T}\boldsymbol{M}_c\\ \boldsymbol{J}_{eb} & \boldsymbol{J}_{eq}\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}_b^T\\ \boldsymbol{Z}_q^T\end{bmatrix} = \boldsymbol{0} \qquad \Longrightarrow \qquad \breve{\boldsymbol{J}}_e\boldsymbol{Z}_q^T = \boldsymbol{0}\,, \tag{7.6}$$

i.e. $\boldsymbol{Z}_q$ is a nullspace base matrix for the generalized Jacobian matrix $\breve{\boldsymbol{J}}_e$. The inverse transformation $\boldsymbol{\Psi}_N^{-1} = \begin{bmatrix}\boldsymbol{\Psi}^{+_M} & \boldsymbol{Z}^T\end{bmatrix}$ can be obtained by direct computation of the weighted pseudoinverse as

$$\boldsymbol{\Psi}_N^{-1} = \begin{bmatrix}\boldsymbol{J}_{eb}^{-1}\boldsymbol{J}_{eq}\breve{\boldsymbol{J}}_e^{+_{\breve{M}}}\boldsymbol{J}_{eb}\boldsymbol{M}_b^{-1}\boldsymbol{J}_{eb}^T\mathrm{Ad}_{c,e}^T & -\boldsymbol{M}_b^{-1}\boldsymbol{M}_c\breve{\boldsymbol{J}}_e^{+_{\breve{M}}} & \boldsymbol{Z}_b^T\\ -\breve{\boldsymbol{J}}_e^{+_{\breve{M}}}\boldsymbol{J}_{eb}\boldsymbol{M}_b^{-1}\boldsymbol{J}_{eb}^T\mathrm{Ad}_{c,e}^T & \breve{\boldsymbol{J}}_e^{+_{\breve{M}}} & \boldsymbol{Z}_q^T\end{bmatrix}\,, \tag{7.7}$$

being $\breve{\boldsymbol{J}}_e^{+_{\breve{M}}} = \breve{\boldsymbol{M}}^{-1}\breve{\boldsymbol{J}}_e^T\left(\breve{\boldsymbol{J}}_e\breve{\boldsymbol{M}}^{-1}\breve{\boldsymbol{J}}_e^T\right)^{-1}$ the weighted pseudo-inverse[1] of the generalized Jacobian matrix $\breve{\boldsymbol{J}}_e$ with a weight matrix $\breve{\boldsymbol{M}} = \boldsymbol{M}_q - \boldsymbol{M}_c^T\boldsymbol{M}_b\boldsymbol{M}_c$. This inertia matrix is commonly indicated as *generalized* inertia matrix of the free-floating arm [YSKU92] and, as the generalized Jacobian matrix, can be obtained with the coordinates transformation used in section 2.2.2. Indeed, it coincides with the matrix $\boldsymbol{\Lambda}_q$ defined there, as it can be shown by direct computation using the block partitioned matrices.

[1] Notice that for not redundant robot the pseudo-inverse matrices reduce to the inverse matrices.

7.2.2. Design of the momentum decoupling control

Given the expression of $\boldsymbol{\Psi}_N^{-1}$ in (7.7), one can introduce the *generalized* nullspace matrix $\check{\boldsymbol{N}}$ defined as $\check{\boldsymbol{N}} = \left(\boldsymbol{Z}_q \check{\boldsymbol{M}} \boldsymbol{Z}_q^T\right)^{-1} \boldsymbol{Z}_q \check{\boldsymbol{M}}$, so that the choice

$$\boldsymbol{\tau} = \check{\boldsymbol{J}}_e^T \boldsymbol{f}_e + \check{\boldsymbol{N}}^T \boldsymbol{f}_n \,, \tag{7.8}$$

leads to a closed loop system that after the coordinate transformation, can be written as

$$\boldsymbol{\Lambda} \begin{bmatrix} \dot{\boldsymbol{h}} \\ \dot{\boldsymbol{\nu}}_e \\ \dot{\boldsymbol{v}}_n \end{bmatrix} + \boldsymbol{\Gamma} \begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{\nu}_e \\ \boldsymbol{v}_n \end{bmatrix} + \boldsymbol{\gamma} = \begin{bmatrix} -\mathrm{Ad}_{c,e}\, \boldsymbol{J}_{eb} \boldsymbol{M}_b^{-1} \boldsymbol{J}_{eb}^T \boldsymbol{f}_e \\ \boldsymbol{f}_e \\ \boldsymbol{f}_n \end{bmatrix} \,, \tag{7.9}$$

where

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_h & \boldsymbol{\Lambda}_{he} & \boldsymbol{0} \\ \boldsymbol{\Lambda}_{he}^T & \boldsymbol{\Lambda}_e & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Lambda}_n \end{bmatrix} \qquad \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_h & \boldsymbol{\Gamma}_{he} & \boldsymbol{\Gamma}_{hn} \\ \boldsymbol{\Gamma}_{eh} & \boldsymbol{\Gamma}_e & \boldsymbol{\Gamma}_{en} \\ -\boldsymbol{\Gamma}_{hn}^T & -\boldsymbol{\Gamma}_{en}^T & \boldsymbol{\Gamma}_n \end{bmatrix} \qquad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\gamma}_h \\ \boldsymbol{\gamma}_e \\ \boldsymbol{\gamma}_n \end{bmatrix} \,.$$

For a free-floating robot the transposed of the generalized Jacobian matrix and the transposed of the generalized nullspace matrix are used for the end effector force and the nullspace force respectively. In (7.9) the nullspace dynamics is inertially decoupled from the rest, which is a consequence of choosing a dynamically consistent nullspace matrix. However, the system is still fully coupled in the Coriolis/centrifugal matrix. Finally, similarly to the not redundant case, by rearranging the equations it is possible to show that the generalized momentum changes only due to the gravity.

The control objective is to define the new control inputs $\boldsymbol{f}_e$ and $\boldsymbol{f}_n$ in order to cancel the effects of the generalized momentum $\boldsymbol{h}$ and further decouple the end effector and nullspace dynamics in order to show convergence of the end effector to the target pose and damping of the internal motion. Given the hypothesis of considering the robot only during the flight phase, the expression of $\dot{\boldsymbol{h}}$ can be considered known without requiring measurements of the external forces and torques. It is enough to know the total mass of the robot and the direction of gravity. The generalized momentum can also be assumed to be known once the complete velocity coordinates of the robot are measured. A momentum decoupling law is given by

$$\boldsymbol{f}_e = \boldsymbol{\Gamma}_{eh} \boldsymbol{h} + \boldsymbol{\Gamma}_{en} \boldsymbol{v}_n - \boldsymbol{J}_{\tilde{x}\omega}^T \boldsymbol{K}_x \tilde{\boldsymbol{x}} - \boldsymbol{D}_x \boldsymbol{\nu}_e \tag{7.10a}$$

$$\boldsymbol{f}_n = \boldsymbol{\Gamma}_{nh} \boldsymbol{h} + \boldsymbol{\Gamma}_{ne} \boldsymbol{\nu}_e - \boldsymbol{D}_n \boldsymbol{v}_n \,, \tag{7.10b}$$

where $\boldsymbol{K}_x, \boldsymbol{D}_x$ are symmetric positive definite matrices and $\boldsymbol{J}_{\tilde{x}\omega}(\tilde{\boldsymbol{x}})$ is the representation Jacobian [Ott08] which correlates the error derivative $\dot{\tilde{\boldsymbol{x}}}$ and the end effector twist $\boldsymbol{\nu}_e$, i.e. $\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{J}_{\tilde{x}\omega}(\tilde{\boldsymbol{x}}) \boldsymbol{\nu}_e$.

7.2.3. Analysis

The closed loop end-effector and nullspace equations are

$$\boldsymbol{\Lambda}_e(\boldsymbol{q}) \dot{\boldsymbol{\nu}}_e + \Big(\boldsymbol{\Gamma}_e(\boldsymbol{q}, \boldsymbol{\xi}) + \boldsymbol{D}\Big) \boldsymbol{\nu}_e + \boldsymbol{J}_{\tilde{x}\omega}^T(\tilde{\boldsymbol{x}}) \boldsymbol{K} \tilde{\boldsymbol{x}} = \boldsymbol{0} \tag{7.11a}$$

$$\boldsymbol{\Lambda}_n(\boldsymbol{q}) \dot{\boldsymbol{v}}_n + \Big(\boldsymbol{\Gamma}_n(\boldsymbol{q}, \boldsymbol{\xi}) + \boldsymbol{D}_n\Big) \boldsymbol{v}_n = \boldsymbol{0} \,, \tag{7.11b}$$

where for brevity $\boldsymbol{\xi} = [\boldsymbol{h}^T \boldsymbol{\nu}_e^T \boldsymbol{v}_n^T]^T$.

With the aid of the results in [MM07] and given the expression of the dynamic matrices in (2.17), the following proposition is given

Proposition 7.1. *There exist positive constants $\lambda_{\Lambda,min}$, $\lambda_{\Lambda,max}$, γ_Γ, $\gamma_{\dot\Gamma_1}$, $\gamma_{\dot\Gamma_2}$ such that for all $\boldsymbol{q} \in \mathbb{Q}$, $\boldsymbol{\xi} \in \mathbb{R}^{n_v}$*

$$\lambda_{\Lambda,min} \leq \|\boldsymbol{\Lambda}(\boldsymbol{q})\| \leq \lambda_{\Lambda,max} \tag{7.12a}$$

$$\|\boldsymbol{\Gamma}(\boldsymbol{q},\boldsymbol{\xi})\| \leq \gamma_\Gamma \|\boldsymbol{\xi}\| \tag{7.12b}$$

$$\|\dot{\boldsymbol{\Gamma}}(\boldsymbol{q},\boldsymbol{\xi})\| \leq \gamma_{\dot\Gamma_1} \|\boldsymbol{\xi}\| + \gamma_{\dot\Gamma_2} \|\dot{\boldsymbol{\xi}}\| \ . \tag{7.12c}$$

In order to analyze the asymptotic stability of the system (7.11), it is useful to rewrite it as the time-varying system:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{J}_{\tilde{x}\omega}(\tilde{\boldsymbol{x}})\boldsymbol{\nu}_e \tag{7.13a}$$

$$\boldsymbol{\Lambda}_e(t)\dot{\boldsymbol{\nu}}_e + \Big(\boldsymbol{\Gamma}_e(t) + \boldsymbol{D}_x\Big)\boldsymbol{\nu}_e + \boldsymbol{J}_{\tilde{x}\omega}^T(\tilde{\boldsymbol{x}})\boldsymbol{K}_x\tilde{\boldsymbol{x}} = \boldsymbol{0} \tag{7.13b}$$

$$\boldsymbol{\Lambda}_n(t)\dot{\boldsymbol{v}}_n + \Big(\boldsymbol{\Gamma}_n(t) + \boldsymbol{D}_n\Big)\boldsymbol{v}_n = \boldsymbol{0} \ , \tag{7.13c}$$

where the dependency on $\boldsymbol{q}$ has been replaced in favor of an explicit time dependency, with state variable $\boldsymbol{\chi} = \begin{bmatrix}\tilde{\boldsymbol{x}}^T & \boldsymbol{\nu}_e^T & \boldsymbol{v}_n^T\end{bmatrix}^T$.

The analysis will be conducted using the function

$$V(t,\boldsymbol{\chi}) = \frac{1}{2}\Big(\boldsymbol{\nu}_e^T\boldsymbol{\Lambda}_e(t)\boldsymbol{\nu}_e + \boldsymbol{v}_n^T\boldsymbol{\Lambda}_n(t)\boldsymbol{v}_n + \tilde{\boldsymbol{x}}^T\boldsymbol{K}_x\tilde{\boldsymbol{x}}\Big) \ , \tag{7.14}$$

which is positive definite and decrescent, since it is lower and upper bounded respectively by two positive definite functions $V_{min}(\boldsymbol{\chi})$ and $V_{max}(\boldsymbol{\chi})$. Denoting by $\lambda_{A,min}$ and $\lambda_{A,max}$ the minimum and maximum eigenvalue of a matrix $\boldsymbol{A}$, the latter functions are given by:

$$V_{min}(\boldsymbol{\chi}) = \frac{1}{2}\Big(\lambda_{\Lambda_e,min}\|\boldsymbol{\nu}_e\|^2 + \lambda_{\Lambda_n,min}\|\boldsymbol{v}_n\|^2 + \lambda_{K_x,min}\|\tilde{\boldsymbol{x}}\|^2\Big) \tag{7.15}$$

$$V_{max}(\boldsymbol{\chi}) = \frac{1}{2}\Big(\lambda_{\Lambda_e,max}\|\boldsymbol{\nu}_e\|^2 + \lambda_{\Lambda_n,max}\|\boldsymbol{v}_n\|^2 + \lambda_{K_x,min}\|\tilde{\boldsymbol{x}}\|^2\Big) \ . \tag{7.16}$$

Differentiating $V(t,\boldsymbol{\chi})$ w.r.t. time, one gets $\dot V(t,\boldsymbol{\chi}) = -\boldsymbol{\nu}_e^T\boldsymbol{D}\boldsymbol{\nu}_e - \boldsymbol{v}_n^T\boldsymbol{D}_n\boldsymbol{v}_n \leq 0$, i.e. it is a negative semidefinite function. The function $V(t,\boldsymbol{\chi})$ is used to show the stability property and convergence; furthermore allows to define a region in which quantities of interest are bounded. This is formalized in the following lemma.

Lemma 7.1. *If $\boldsymbol{q} \in \mathbb{Q}$, then there exists a set Ω such that the state $\boldsymbol{\chi}$, the accelerations $\dot{\boldsymbol{\nu}}_e$, $\dot{\boldsymbol{v}}_n$ and the matrices $\boldsymbol{\Lambda}(t)$, $\boldsymbol{\Gamma}(t)$, $\boldsymbol{J}_{\tilde{x}\omega}$, $\dot{\boldsymbol{J}}_{\tilde{x}\omega}$ are all bounded.*

Proof. Given the positive definite function $V(t,\boldsymbol{\chi})$ in (7.14) and a constant $V_0 > 0$, define the set $\Omega = \{\boldsymbol{\chi} : V(\boldsymbol{\chi}) \leq V_0\}$. Considering that $\dot V(t,\boldsymbol{\chi}) \leq 0$, it follows that there exist $c_1, c_2, c_3 > 0$ such that $\forall \boldsymbol{\chi} \in \Omega$

$$\|\tilde{\boldsymbol{x}}\| \leq c_1 \qquad \|\boldsymbol{\nu}_e\| \leq c_2 \qquad \|\boldsymbol{v}_n\| \leq c_3 \ . \tag{7.17}$$

Without loss of generality, assume to use a quaternion representation for the rotational part of the error $\tilde{\boldsymbol{x}}$, then

$$\|\boldsymbol{J}_{\tilde{x}\omega}(\tilde{\boldsymbol{x}})\| \leq \mu_1 \qquad \|\dot{\boldsymbol{J}}_{\tilde{x}\omega}(\tilde{\boldsymbol{x}})\| \leq \mu_2\|\boldsymbol{\nu}_e\| \ , \tag{7.18}$$

with $\mu_1, \mu_2 > 0$.

Using (7.18) and (7.17), the bounds for the end-effector and nullspace accelerations $\dot{\boldsymbol{\nu}}_e$ and $\dot{\boldsymbol{v}}_n$ are:

$$\|\dot{\boldsymbol{\nu}}_e\| = \| - \boldsymbol{\Lambda}_e^{-1}\Big(\boldsymbol{\Gamma}_e + \boldsymbol{D}_x\Big)\boldsymbol{\nu}_e - \boldsymbol{\Lambda}_e^{-1}\boldsymbol{J}_{\tilde{x}\omega}^T\boldsymbol{K}_x\tilde{\boldsymbol{x}}\| \leq c_4 \tag{7.19a}$$

$$\|\dot{\boldsymbol{v}}_n\| = \| - \boldsymbol{\Lambda}_n^{-1}\boldsymbol{\Gamma}_n\boldsymbol{v}_n - \boldsymbol{\Lambda}_n^{-1}\boldsymbol{D}_n\boldsymbol{v}_n\| \leq c_5, \tag{7.19b}$$

with $c_4, c_5 > 0$.

The bounds for $\dot{\boldsymbol{\Lambda}}_e$ and $\dot{\boldsymbol{\Gamma}}_e$ are obtained considering (7.17), (7.19)

$$\|\dot{\boldsymbol{\Lambda}}_e\| = \|\boldsymbol{\Gamma}_e + \boldsymbol{\Gamma}_e^T\| \leq c_6 \qquad\qquad \|\dot{\boldsymbol{\Gamma}}_e\| \leq c_7 \ .$$

□

To show the convergence, Barbalat's Lemma is used and it is reported here for convenience [SL91].

Lemma 7.2 (Barbalat's Lemma)**.** *If a differentiable function $f(t)$ has a finite limit as $t \to \infty$, and is such that $\ddot{f}$ exists and is bounded, then $\dot{f}(t) \to 0$ as $t \to 0$.*

Proposition 7.2. *If $\boldsymbol{\chi} \in \Omega$ and Assumption 7.1 holds, then $\boldsymbol{\chi} = \mathbf{0}$ is an asymptotically stable equilibrium point for the system* (7.13).

Proof. Using the time-varying Lyapunov function $V(t, \boldsymbol{\chi})$, with $V(t, \mathbf{0}) = 0$, one concludes that $\boldsymbol{\chi} = \mathbf{0}$ is (uniformly) stable. Convergence can be proved by means of a double application of Barbalat's lemma. The first time, Barbalat's lemma is applied on $V(t, \boldsymbol{z})$ to show the convergence of the velocity. To this end, it is necessary to show that $\dot{V}(t, \boldsymbol{\chi})$ is bounded along the flow of the system. Applying Lemma 7.1, it exists a set Ω where $\boldsymbol{\nu}_e$, $\dot{\boldsymbol{\nu}}_e$, $\boldsymbol{v}_n, \dot{\boldsymbol{v}}_n$ are bounded and, accordingly:

$$|\ddot{V}(t, \boldsymbol{z})| = | - 2\boldsymbol{\nu}_e^T\boldsymbol{D}_x\dot{\boldsymbol{\nu}}_e - 2\boldsymbol{v}_n^T\boldsymbol{D}_n\dot{\boldsymbol{v}}_n| \leq 2c_2c_4\lambda_{D_x,max} + 2c_3c_5\lambda_{D_n,max} \ , \tag{7.20}$$

where $c_2, c_3, c_4, c_5 > 0$ are respectively the bounds of $\boldsymbol{\nu}_e, \boldsymbol{v}_n, \dot{\boldsymbol{\nu}}_e, \dot{\boldsymbol{v}}_n$, as derived in the proof of Lemma 7.1. Applying Barbalat's lemma on $V(t, \boldsymbol{z})$ it results $\dot{V}(t, \boldsymbol{z}) \to 0$ and accordingly $\boldsymbol{\nu}_e \to \mathbf{0}$ and $\boldsymbol{v}_n \to \mathbf{0}$, i.e. the end effector and nullspace velocities converge to zero.

The second time, Barbalat's lemma is applied on $\boldsymbol{\nu}_e$ to show the convergence of $\dot{\boldsymbol{\nu}}_e$, which is then used to show the convergence of the error $\tilde{\boldsymbol{x}}$. To this end, consider $\ddot{\boldsymbol{\nu}}_e$, which can be obtained differentiating (7.13b):

$$\boldsymbol{\Lambda}_e\ddot{\boldsymbol{\nu}}_e + \Big(\dot{\boldsymbol{\Lambda}}_e + \boldsymbol{\Gamma}_e + \boldsymbol{D}_x\Big)\dot{\boldsymbol{\nu}}_e + \Big(\dot{\boldsymbol{\Gamma}}_e + \boldsymbol{J}_{\tilde{x}\omega}^T\boldsymbol{K}_x\boldsymbol{J}_{\tilde{x}\omega}\Big)\boldsymbol{\nu}_e + \dot{\boldsymbol{J}}_{\tilde{x}\omega}^T\boldsymbol{K}_x\tilde{\boldsymbol{x}} = \mathbf{0} \ . \tag{7.21}$$

Considering Lemma 7.1, also $\ddot{\boldsymbol{\nu}}_e$ is shown to be bounded

$$\|\ddot{\boldsymbol{\nu}}_e\| \leq \|\boldsymbol{\Lambda}_e^{-1}\| \left(\|\dot{\boldsymbol{\Lambda}}_e + \boldsymbol{\Gamma}_e + \boldsymbol{D}_x\| \, \|\dot{\boldsymbol{\nu}}_e\| + \|\dot{\boldsymbol{\Gamma}}_e + \boldsymbol{J}_{\tilde{x}\omega}^T\boldsymbol{K}_x\boldsymbol{J}_{\tilde{x}\omega}\| \, \|\boldsymbol{\nu}_e\| + \|\dot{\boldsymbol{J}}_{\tilde{x}\omega}^T\boldsymbol{K}_x\| \, \|\tilde{\boldsymbol{x}}\|\right) \leq c_8 \ , \tag{7.22}$$

with $c_8 > 0$. Applying Barbalat's Lemma on $\boldsymbol{\nu}_e$, it follows that $\dot{\boldsymbol{\nu}}_e \to \mathbf{0}$. The convergence of the end-effector error $\tilde{\boldsymbol{x}}$ can be finally shown considering the limit of (7.13b)

$$\lim_{t\to\infty}\Big[\boldsymbol{\Lambda}_e(t)\dot{\boldsymbol{\nu}}_e + \Big(\boldsymbol{\Gamma}_e + \boldsymbol{D}_x\Big)\boldsymbol{\nu}_e + \boldsymbol{J}_{\tilde{x}\omega}^T(\tilde{\boldsymbol{x}})\boldsymbol{K}_x\tilde{\boldsymbol{x}}\Big] = \mathbf{0} \ , \tag{7.23}$$

from which it follows that $\tilde{\boldsymbol{x}} \to \mathbf{0}$. Therefore, after using Barbalat's Lemma twice, the whole state $\boldsymbol{\chi}$ converges to zero. □

Table 7.1.: Parameters of the free-floating robot.

	l [m]	m [kg]	I_x [kgm^2]	I_y [kgm^2]	I_z [kgm^2]
Base	1.17	375	165	280	250
Link 1	0.17	5.1	0.03	0.03	0.03
Link 2	1.3	18.8	1.65	0.64	1.65
Link 3	0.17	8.9	0.15	0.03	0.14
Link 4	1.3	12.0	0.25	0.03	0.25
Link 5	0.17	11.7	0.26	0.03	0.26
Link 6	0.1	5.5	0.02	0.02	0.03
Link 7	0	4.7	0.02	0.01	0.01

7.2.4. Limitation of the control law

From the inverse of the momentum transformation (7.5) it is possible to see that, while the end effector and nullspace velocities go to zero, the coupled base-joints motion continues to evolve as a consequence of the nonzero generalized momentum $\boldsymbol{h}$:

$$\boldsymbol{\nu}_b \to \boldsymbol{J}_{eb}^{-1}\boldsymbol{J}_{eq}\check{\boldsymbol{J}}_e^{+_{\check{M}}}\boldsymbol{J}_{eb}\boldsymbol{M}_b^{-1}\boldsymbol{J}_{eb}^T\,\mathrm{Ad}_{c,e}^T\,\boldsymbol{h} \tag{7.24a}$$

$$\dot{\boldsymbol{q}} \to -\check{\boldsymbol{J}}_e^{+_{\check{M}}}\boldsymbol{J}_{eb}\boldsymbol{M}_b^{-1}\boldsymbol{J}_{eb}^T\,\mathrm{Ad}_{c,e}^T\,\boldsymbol{h}\ . \tag{7.24b}$$

Additionally, the convergence of the end effector is guaranteed only when the singularities of $\check{\boldsymbol{J}}_e(\boldsymbol{q})$ are avoided. However, in presence of initial momentum, the singularities could be encountered also when the end effector has converged, due to the highlighted persistent joint motion. Considering, for example, the case when the system has a linear momentum, the joints will necessarily reach after a certain time a singular configuration. This result, observed also in [NP11], is a natural consequence of the fact that it is not possible to control the end-effector in the inertial frame for an indefinite time if the whole system is moving in space with a linear velocity. On the other hand, in the case of only angular momentum, singularities could be less problematic, however the robot will likely converge to joint limits, due to the persistent joint motion induced by the angular momentum.

In conclusion, until a singularity is reached, the error will continuously decrease due to the proven convergence properties. In this way, in a space mission, the required end-effector precision could be ensured during the time of the gripper closure. Additional measures need to be taken to avoid entering the singularities, for example by activating the satellite thrusts. For a humanoid robot, instead, the next contact phase will be used to readjust the uncontrolled variables. From a theoretical point of view, the controller has no assumption on the maximum amount of momentum that can be handled, however the higher the momenta, the shorter the time to singularity will be.

7.3. Simulation results

In order to verify the proposed control method, a seven degrees-of-freedom free-floating robot in presence of linear and angular momenta is simulated. The kinematics and dynamics parameters of the robot are defined in Tab. 7.1.

The initial joints configuration is $\boldsymbol{q}(0) = \begin{bmatrix}0 & 1.047 & 0 & -1.047 & 0 & 0.52 & 0\end{bmatrix}^T$ rad, as schematically depicted in Fig. 7.1. The initial linear and angular momenta are due to the initial linear velocity $\boldsymbol{v}_b = \begin{bmatrix}-2 & 1 & 53\end{bmatrix}^T 10^{-3}\,\mathrm{m/s}$ and angular velocity $\boldsymbol{\omega}_b = \begin{bmatrix}-3.5 & -57.6 & 3.5\end{bmatrix}^T 10^{-3}\,\mathrm{rad/s}$ of the base. The desired pose of the end-effector differs from the initial one of $\Delta\boldsymbol{p} = [-0.035, 0, -0.03]\,\mathrm{m}$

for the position and of $\Delta\phi = [0, 0.087, 0]$ rad for the orientation. The proposed controller is compared against [MMA89]:

$$\boldsymbol{\tau} = -\check{\boldsymbol{J}}_e^T \left(\boldsymbol{J}_{\tilde{x}\omega}^T \boldsymbol{K}_x \tilde{\boldsymbol{x}} + \boldsymbol{D}_x \boldsymbol{\nu}_e \right) , \tag{7.25}$$

where no compensation of the generalized momentum is used. The stiffness and damping values are respectively $K_t = 165\,\mathrm{N/m}$, $D_t = 175\,\mathrm{Ns/m}$ for the translation, $K_r = 20\,\mathrm{Nm/rad}$ and $D_r = 20\,\mathrm{Nms/rad}$ for the rotation. The norm of the end-effector position error is shown in Fig. 7.2 for both controllers. The results show that for the proposed control the end-effector converges to the desired position. On the other hand, a deviation on the end-effector is induced in case that the momentum compensation is not used. In general, this deviation is time-varying and its magnitude depends on both the initial momentum and on the particular maneuver commanded to the robot, since $\boldsymbol{\Gamma}_{eh} = \boldsymbol{\Gamma}_{eh}(\boldsymbol{q}(t))$.

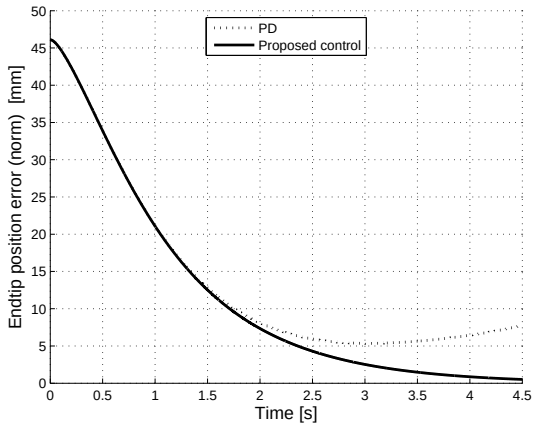

Figure 7.2.: Norm of the end effector position error for PD control and for proposed control.

The joint angles and base attitude are shown for the proposed control respectively in Fig. 7.3 and Fig. 7.4, where it is possible to notice that the joints and base motion continues to evolve while the end-effector remains fixed, as a consequence of the presence of momentum.

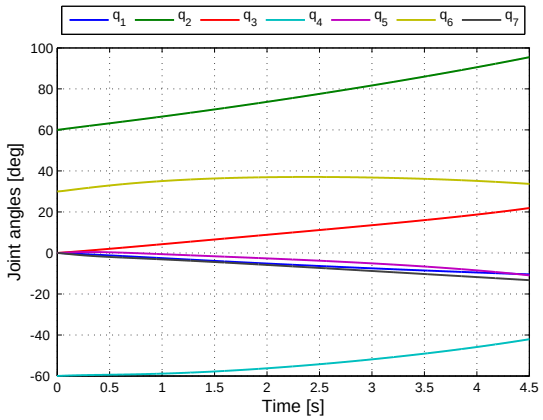

Figure 7.3.: Joint angles with the proposed control.

The convergence of the nullspace velocity v_n can be observed in Fig. 7.5, showing the effectiveness of the nullspace control.

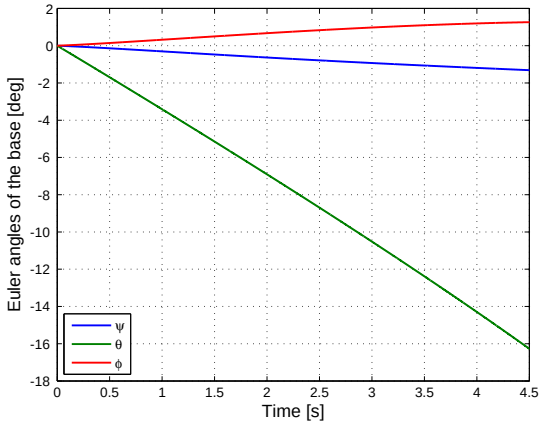

Figure 7.4.: Attitude of the base.

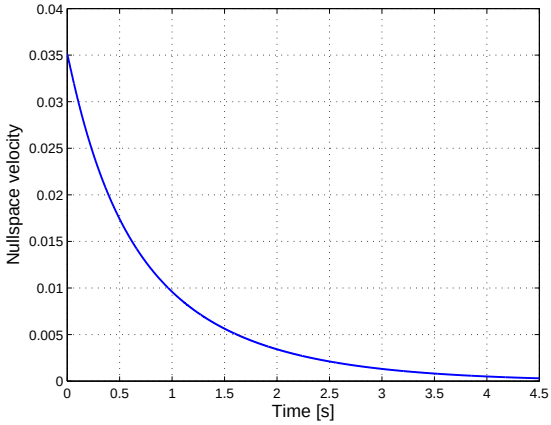

Figure 7.5.: Nullspace velocity with the proposed control.

Since the proposed control is based on the online dynamic model of the space robot, uncertainties in the dynamics parameters are introduced, in order to test the robustness. The proposed controller has been tested with 20 randomly generated models, considering for each body 10% uncertainty in the mass and 40% uncertainty in the inertias and in the CoMs. Fig. 7.6 shows an envelope of norm of the step response of the proposed controller with the uncertain models. The results show that the controller is robust considering dynamics parameters uncertainties and a worst-case deviation of 3.5 mm from the nominal behavior is obtained.

7.4. Summary

In this chapter, the dynamics of a free-floating system in presence of linear and angular momenta was investigated. It was shown that the presence of the momentum introduces a drift on the end effector with standard methods, therefore a torque control law has been proposed in order to compensate this effect without feedback linearizing the system. For the proposed controller, a the stability proof was given under the hypothesis of non-singularity of the generalized Jacobian matrix. The effectiveness of the control has been tested in simulation considering model uncertainties.

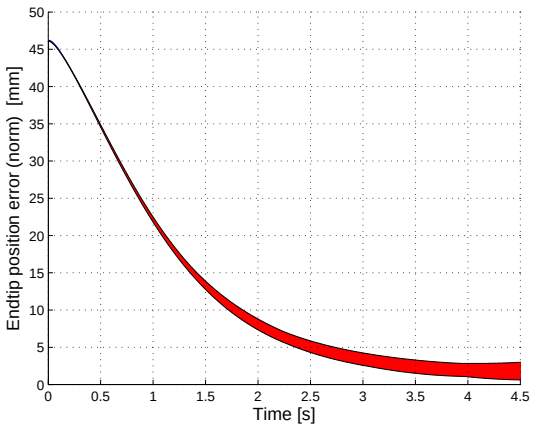

Figure 7.6.: Envelope of step response (norm) of the end effector for the proposed controller for 20 uncertain models.

CHAPTER 8

Improving the robustness

The problem of coping with uncertain models has been widely studied in robotics both from a robust control point of view [SLT92, CX99, PF99, PVAL+03] and from an adaptive control point of view [SL87, OS88, SH90]. The contribution of this chapter is the design of a feedback control law to produce closed attractive orbits for fully actuated robots with unknown disturbances, which can be modeled using a regressors matrix, e.g. torques due to friction. This extends the results of chapter 4 where the ideal frictionless case was considered. The idea is once again to impose $n-1$ constraints to the system and generate an attractive path for the "remaining dynamics". Since the design of the controller is carried on in two independent steps, the friction compensation cannot be achieved with standard techniques. As for the control design, also the friction compensation will be taken into account in two different steps. To this end, the part of the controller responsible for forcing the system to evolve on a 1 - dimensional submanifold of the configuration space is modified using a sliding mode approach as in [PVAL+03], which is robust to model uncertainty and guarantees asymptotic convergence. For the "remaining dynamics", on the other hand, it is not as straightforward to apply a sliding mode approach. For this reason, in the nullspace an adaptive approach is used to compensate for the disturbance as in [SL87]. Crucial for the analysis is the concept of conditional attractiveness with semidefinite Lyapunov functions to show the attractiveness of the closed orbit of the whole system.

8.1. The adaptive control in a nutshell

The adaptive tracking control law presented in [SL87] is given by a feedback term proportional to the sliding variable $\boldsymbol{s}$ (which is a linear combination of the velocity and position tracking errors) and a feedforward term which uses an estimation of the unknown parameters. Given a positive definite matrix $\boldsymbol{P}$, the estimation is computed using an adaptive law which renders the positive definite function

$$V_\pi = \frac{1}{2}\left(\boldsymbol{s}^T \boldsymbol{M}(\boldsymbol{q})\boldsymbol{s} + \tilde{\boldsymbol{\pi}}^T \boldsymbol{P}^{-1}\tilde{\boldsymbol{\pi}}\right) , \tag{8.1}$$

a Lyapunov function, where $\tilde{\boldsymbol{\pi}} = \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}$ is the error between the estimated and real parameters. In particular, the estimation is updated as

$$\dot{\hat{\boldsymbol{\pi}}} = -\boldsymbol{P}\boldsymbol{Y}_r^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{s} , \tag{8.2}$$

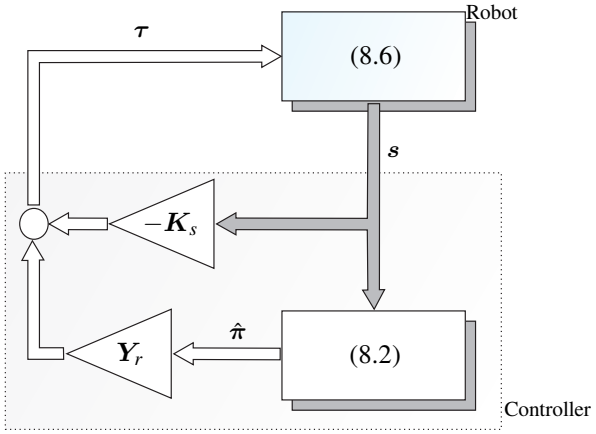

Figure 8.1.: The adaptive Slotine - Li control. The matrix $\boldsymbol{K}_s$ is positive definite and diagonal.

where $\boldsymbol{Y}_r$ is the Slotine - Li regressor and $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}} - \boldsymbol{s}$. The control scheme is as in Fig. 8.1.

8.2. The dynamic sliding PID control in a nutshell

The dynamic sliding PID control [PVAL$^+$03] uses, as the control law in [SL87], a sliding variable $\boldsymbol{s}$. Nevertheless, unlike in [SL87], the control input is not chosen in order to obtain an asymptotically stable dynamics for the sliding variable, but to obtain a stable dynamics in terms of the output $\boldsymbol{s}_r$ of the dynamic system

$$\dot{\boldsymbol{s}}_d = -\kappa \boldsymbol{s}_d \tag{8.3a}$$
$$\dot{\boldsymbol{\sigma}} = \operatorname{sgn}(\tilde{\boldsymbol{s}}) \tag{8.3b}$$
$$\boldsymbol{s}_r = \boldsymbol{Q}_s \boldsymbol{\sigma} + \tilde{\boldsymbol{s}}\,, \tag{8.3c}$$

with $\tilde{\boldsymbol{s}} = \boldsymbol{s} - \boldsymbol{s}_d$, being $\boldsymbol{s}$ the input and $\boldsymbol{s}_d, \boldsymbol{\sigma}$ the state. The latter is initialized as $\boldsymbol{s}_d(t_0) = \boldsymbol{s}(t_0)$, i.e. the value of $\boldsymbol{s}$ at the initial time t_0, and $\boldsymbol{\sigma}(t_0) = \boldsymbol{0}$. Finally, $\kappa > 0$, $\boldsymbol{Q}_s$ is a positive definite diagonal matrix and $\operatorname{sgn}(\boldsymbol{a}) = \left[\operatorname{sgn}(a_1) \ldots \operatorname{sgn}(a_n)\right]^T$, for $\boldsymbol{a} \in \mathbb{R}^n$. Using a control input proportional to $\boldsymbol{s}_r$, as it is shown in Fig. 8.2, and given some bounds for the terms present in the dynamic model of the robot, it is possible to prove the boundedness of $\boldsymbol{s}_r$ and $\dot{\boldsymbol{s}}_r$. The latter allows to conclude that a sliding mode is established for all time on the error variable $\tilde{\boldsymbol{s}}$. To this end, the following second order dynamical system defined by (8.3)

$$\dot{\tilde{\boldsymbol{s}}} = -\boldsymbol{Q}_s \operatorname{sgn}(\tilde{\boldsymbol{s}}) + \dot{\boldsymbol{s}}_r\,, \tag{8.4}$$

together with the Lyapunov function

$$V_s = \frac{1}{2} \tilde{\boldsymbol{s}}^T \tilde{\boldsymbol{s}}\,, \tag{8.5}$$

are used, since they imply the sliding mode condition $\tilde{\boldsymbol{s}}^T \dot{\tilde{\boldsymbol{s}}} \leq -\mu |\tilde{\boldsymbol{s}}|$, $\mu > 0$ [Utk92]. In this way, the tracking errors are constrained to evolve on a submanifold that has an exponential solution toward zero [PVAL$^+$03].

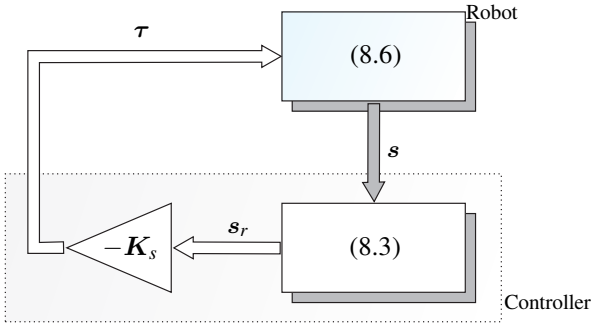

Figure 8.2.: Dynamic sliding PID control. The matrix $\boldsymbol{K}_s$ is positive definite and diagonal.

8.3. Energy regulation with friction compensation

As in chapter 4, a function $\boldsymbol{y}(\boldsymbol{q})$ is introduced that defines a 1 - dimensional submanifold. Additionally, a virtual potential function $U(\boldsymbol{q})$ is chosen, such that it posses a constrained local minimum at $\boldsymbol{q}_d$ on the submanifold. In section 4.3.1, it has been shown that in this case $\boldsymbol{y}(\boldsymbol{q}_d) = \boldsymbol{0}$ and $n(\boldsymbol{q}_d) = 0$, where $n(\boldsymbol{q}) := \boldsymbol{Z}(\boldsymbol{q})\nabla_{\boldsymbol{q}} U(\boldsymbol{q})$. The control law derived here relies on the coordinate transformation presented in section 4.2. As a consequence, a perfect knowledge of the system matrices is required. The main reason for this is that the inertia matrix of the robot is required in order to compute the nullspace projector $\boldsymbol{N}(\boldsymbol{q})$. Therefore, the considered uncertain model is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_f \,, \tag{8.6}$$

where $\boldsymbol{\tau}_f \in \mathbb{R}^n$ are unknown torques acting as a disturbance. Additionally, the following properties are assumed

Assumption 8.1.

$$\boldsymbol{\tau}_f = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\pi} \tag{8.7}$$

where $\boldsymbol{\pi} \in \mathbb{R}^p$ is a constant vector of unknown parameters and $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times p}$ is the so called regressor matrix, whose entries, on the contrary, are assumed to be known.

Assumption 8.2.

$$\left\|\boldsymbol{\tau}_f\right\| \leq \beta_1 \left\|\boldsymbol{q}\right\| + \beta_2 \left\|\dot{\boldsymbol{q}}\right\| + \beta_3 \tag{8.8}$$

where $\beta_1, \beta_2, \beta_3 \in \mathbb{R}$ are positive scalars.

Viscous and Coulomb friction, as well as constant torques and Cartesian forces, are classical examples of disturbances that can be modeled in this way.

According to the dynamic sliding PID control, the sliding variable is defined as

$$\boldsymbol{s} = \dot{\boldsymbol{y}} + \boldsymbol{D}_s \boldsymbol{y} \,, \tag{8.9}$$

and the nominal reference as

$$\dot{\boldsymbol{y}}_r = \dot{\boldsymbol{y}} - \boldsymbol{s}_r = -\boldsymbol{D}_s \boldsymbol{y} + \boldsymbol{s}_d - \boldsymbol{Q}_s \boldsymbol{\sigma} \,, \tag{8.10}$$

being $\boldsymbol{s}_r$ the output of the system (8.3). Given the system (8.6) with Assumption (8.1) - (8.2), let $K_H, \kappa \in \mathbb{R}$ be positive scalars, $\boldsymbol{D}_s, \boldsymbol{K}_s, \boldsymbol{Q}_s \in \mathbb{R}^{(n-1)\times(n-1)}$ positive definite diagonal matrices and

$\boldsymbol{P} \in \mathbb{R}^{p\times p}$ a symmetric positive definite matrix, then the proposed nonlinear controller is (omitting the dependences)

$$\boldsymbol{\tau} = \boldsymbol{g} + \boldsymbol{J}_N^T \left(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Gamma}_{xn} \\ -\boldsymbol{\Gamma}_{xn}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ v \end{bmatrix} + \begin{bmatrix} \boldsymbol{f}_x \\ f_n \end{bmatrix} \right) , \tag{8.11}$$

where

$$\boldsymbol{f}_x = \boldsymbol{\Lambda}_x \ddot{\boldsymbol{y}}_r + \boldsymbol{\Gamma}_x \dot{\boldsymbol{y}}_r - \boldsymbol{K}_s \boldsymbol{s}_r \tag{8.12}$$

$$f_n = -K_H \tilde{H} v + \boldsymbol{Z} \Big(\boldsymbol{Y} \hat{\boldsymbol{\pi}} - \nabla_{\boldsymbol{q}} U \Big) . \tag{8.13}$$

Using the state vector $\boldsymbol{\chi} = (\boldsymbol{q}, \dot{\boldsymbol{y}}, v, \boldsymbol{s}_d, \boldsymbol{\sigma}, \hat{\boldsymbol{\pi}})$, the closed loop system is:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+_M} \dot{\boldsymbol{y}} + \boldsymbol{Z}^T v \tag{8.14a}$$

$$\boldsymbol{\Lambda}_x \dot{\boldsymbol{s}}_r + \Big(\boldsymbol{\Gamma}_x + \boldsymbol{K}_s \Big) \boldsymbol{s}_r = \boldsymbol{J}^{+_M T} \boldsymbol{Y} \boldsymbol{\pi} \tag{8.14b}$$

$$\Lambda_n \dot{v} + \Big(\Gamma_n + K_H \tilde{H} \Big) v + \boldsymbol{Z} \nabla_{\boldsymbol{q}} U = \boldsymbol{Z} \boldsymbol{Y} \tilde{\boldsymbol{\pi}} \tag{8.14c}$$

$$\dot{\boldsymbol{s}}_d = -\kappa \boldsymbol{s}_d \tag{8.14d}$$

$$\dot{\boldsymbol{\sigma}} = \operatorname{sgn}(\tilde{\boldsymbol{s}}) \tag{8.14e}$$

$$\dot{\hat{\boldsymbol{\pi}}} = -\boldsymbol{P} \boldsymbol{Y}^T \boldsymbol{Z}^T \tilde{H} v , \tag{8.14f}$$

where the signals $\boldsymbol{s}_r$, $\dot{\boldsymbol{s}}_r$ and $\boldsymbol{s}$ are functions of the state only.

Theorem 8.1. *For any $H_d > 0$, the nonlinear autonomous system* (8.14) *has bounded solutions locally converging to a closed attractive orbit defined by*

$$\mathscr{L}_d = \left\{ \boldsymbol{\chi} \mid \tilde{H} = 0,\ \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}, \boldsymbol{s}_d = \boldsymbol{0}, \boldsymbol{\sigma} = \boldsymbol{\sigma}^*, \hat{\boldsymbol{\pi}} = \hat{\boldsymbol{\pi}}^* \right\} ,$$

for some constant $\boldsymbol{\sigma}^*$, $\hat{\boldsymbol{\pi}}^*$ satisfying the conditions

- $\Big(\boldsymbol{\Gamma}_x + \boldsymbol{K}_s \Big) \boldsymbol{Q}_s \boldsymbol{\sigma}^* = \boldsymbol{J}^{+_M T} \boldsymbol{Y} \boldsymbol{\pi}$
- $v \boldsymbol{Z} \boldsymbol{Y} \Big(\hat{\boldsymbol{\pi}}^* - \boldsymbol{\pi} \Big) = 0 .$

Proof. The proof is based on Theorem A.3 formulated in appendix A.

Using the positive semidefinite C^1 function of the state (8.5), it is shown in [PVAL$^+$03] that $\mathscr{A} = \{ \boldsymbol{\chi} \mid \boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}, \boldsymbol{s}_d = \boldsymbol{0}, \boldsymbol{\sigma} = \boldsymbol{\sigma}^* \}$ is the largest invariant set within $E = \left\{ \boldsymbol{\chi} \mid \dot{V}_s = 0 \right\}$. In the proof the derivative of V_s along the flow of the system is computed, leading to

$$\dot{V}_s = \tilde{\boldsymbol{s}}^T \dot{\tilde{\boldsymbol{s}}} = -\tilde{\boldsymbol{s}}^T \boldsymbol{Q}_s \operatorname{sgn}(\tilde{\boldsymbol{s}}) + \tilde{\boldsymbol{s}}^T \dot{\boldsymbol{s}}_r . \tag{8.15}$$

Given Assumption (8.2) and choosing $\boldsymbol{K}_s$ big enough it is possible to show that $\dot{\boldsymbol{s}}_r$ is bounded. If the minimum eigenvalue of $\boldsymbol{Q}_s$ is also chosen big enough, then the sliding mode condition $\tilde{\boldsymbol{s}}^T \dot{\tilde{\boldsymbol{s}}} \leq -\mu |\tilde{\boldsymbol{s}}|$, with $\mu > 0$, is enforced for all time. From that the convergence of $\tilde{\boldsymbol{s}}$ to zero follows and, given its expression, also the convergence of $\boldsymbol{y}(\boldsymbol{q})$ and $\dot{\boldsymbol{y}}$. Therefore, $\mathscr{A}$ will be the largest invariant set within E, with $\Big(\boldsymbol{\Gamma}_x + \boldsymbol{K}_s \Big) \boldsymbol{Q}_s \boldsymbol{\sigma}^* = \boldsymbol{J}^{+_M T} \boldsymbol{Y} \boldsymbol{\pi}$.

If $\mathscr{L}_d$ can be proven to be invariant for (8.14), attractive conditionally to $\mathscr{A}$ and that the solutions are bounded, then all the requirements of Theorem A.3 are satisfied and $\mathscr{L}_d$ will turn out to be attractive for the whole system (8.14). To show the invariance of $\mathscr{L}_d$, it is enough to show that it

is invariant conditionally to $\mathscr{A}$, since the latter is itself an invariant set. To this end, consider the system

$$\dot{\boldsymbol{q}} = \boldsymbol{Z}^T v \tag{8.16a}$$

$$\Lambda_n \dot{v} + \left(\Gamma_n + K_H \tilde{H}\right) v + \boldsymbol{Z} \nabla_{\boldsymbol{q}} U = \boldsymbol{Z}\boldsymbol{Y}\tilde{\boldsymbol{\pi}} \tag{8.16b}$$

$$\dot{\tilde{\boldsymbol{\pi}}} = -\boldsymbol{P}\boldsymbol{Y}^T \boldsymbol{Z}^T \tilde{H} v \,, \tag{8.16c}$$

with $\boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}$, i.e. (8.14) conditionally to $\mathscr{A}$. Computing the derivative of $\tilde{H}$ along the flow of (8.16), results in

$$\dot{\tilde{H}} = -K_H v^2 \tilde{H} + v\boldsymbol{Z}\boldsymbol{Y}\tilde{\boldsymbol{\pi}} \,, \tag{8.17}$$

which, together with (8.14f), shows that $\tilde{H}$ and $\hat{\boldsymbol{\pi}}^*$ will not change when starting from $\mathscr{L}_d$. To prove the attractiveness, the following positive definite function is used

$$V_n = \frac{1}{2}\left(\tilde{H}^2 + \tilde{\boldsymbol{\pi}}^T \boldsymbol{P}^{-1} \tilde{\boldsymbol{\pi}}\right) \,, \tag{8.18}$$

with derivative $\dot{V}_n = \tilde{H}\dot{\tilde{H}} = -K_H v^2 \tilde{H}^2 \leq 0$. Boundedness is therefore satisfied by the use of non-increasing functions. From LaSalle's invariance theorem, one can conclude that the positive limit set $Ł_n^+$ is given by the largest positive invariant set M_n within the set $E_n = \left\{\boldsymbol{\chi} \mid v = 0 \, or \, \tilde{H} = 0\right\}$, i.e. the set where $\dot{V}_n = 0$. Beside $\mathscr{L}_d$, M_n contains also equilibrium points satisfying the condition $n(\boldsymbol{q}^*) = \boldsymbol{Z}\boldsymbol{Y}\tilde{\boldsymbol{\pi}}$. Starting close enough to $\mathscr{L}_d$ so that the solutions are bounded in a neighborhood of $\mathscr{L}_d$ that does not contain any equilibrium points, they will approach $\mathscr{L}_d$ as $t \to \infty$. Finally, $\mathscr{L}_d$ is uniquely determined by one parameter, e.g. the value of $n(\boldsymbol{q})$, and therefore it is a closed[1] attractive orbit in the state space. □

8.3.1. Controller discussion

The proposed control law, as the methods it is based on [PVAL$^+$03, SL87], cannot guarantee asymptotic stability because no stationary value can be known a priori for $\boldsymbol{\sigma}$ and $\hat{\boldsymbol{\pi}}$. These variables are part of the state of the closed loop system together with the original state of the robot. The latter will always locally converge to a state satisfying the conditions: $\tilde{H} = 0$ and $\boldsymbol{y}(\boldsymbol{q}) = \dot{\boldsymbol{y}} = \boldsymbol{0}$, as expressed in Theorem 8.1. As a result, the physical states of the closed loop system, i.e. the state of the robot, will always converge to the same values.

The torque generated by the controller, as shown in Fig. 8.3, is due to three contributions: a dynamic reshaping of the system, the torque provided by the sliding mode part of the controller and finally the one generated by the adaptive part. The last two are dynamically decoupled and responsible for forcing the system to evolve on a 1 - dimensional submanifold and for creating the attractive closed orbit respectively. The dynamic reshaping part of the controller compensates for the gravity torque and the coupling terms in the Coriolis matrix.

In the analysis it is required that $\boldsymbol{y}(\boldsymbol{q})$ and $\dot{\boldsymbol{y}}$ converge to zero. Therefore, a chattering free sliding mode approach is a perfect candidate for the task. On the other hand, a sliding mode control cannot be used for the zero dynamics. Although the energy error can be used as sliding variable with relative degree $r = 1$, after differentiation the control input would not appear multiplied by the sliding variable itself, as it is for example in [SL87]. An adaptive method can be used in this case. It is also clear that, once it has been chosen to adapt the parameters in such a way that convergence is guaranteed for the zero dynamics, the same parameters cannot be adapted with a different law

[1] It can be seen as the level set of a positive definite and radially unbounded function.

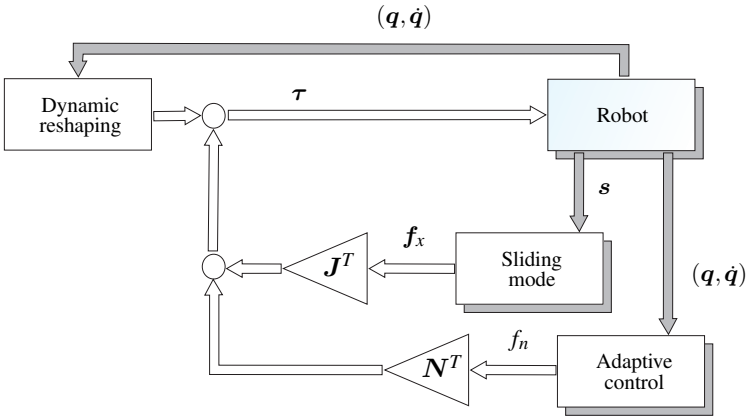

Figure 8.3.: The proposed controller is given by three subparts.

to guarantee the convergence to the constraint submanifold. In other word, it is possible to design the adaptation law for one of the two subparts of the controller. This explains why an adaptive approach has not been used also to enforce the constraints.

8.4. Experiment

The proposed control law is evaluated in an experiment with the humanoid robot TORO described in [EWO+14] and developed at the German Aerospace Center (DLR). The robot has 27 degrees of freedom with a height of 1.7 m and a mass of about 77.5 kg.

Friction model It is assumed that the unknown torques τ_f in (8.6) can be expressed as

$$\tau_f = \operatorname{diag}(f_v)\dot{q} + \operatorname{diag}(f_s)\operatorname{sgn}(\dot{q}) \,, \tag{8.19}$$

where $f_v, f_s \in \mathbb{R}^n$ are the vectors of the unknown coefficients of the viscous and Coulomb friction respectively. The previous torques can be expressed using

$$Y = \begin{bmatrix} \operatorname{diag}(\dot{q}) & \operatorname{diag}(\operatorname{sgn}(\dot{q})) \end{bmatrix}, \qquad \pi = \begin{bmatrix} f_v^T & f_s^T \end{bmatrix}^T \quad .$$

Desired configuration and constraint function In this example the constraint submanifold is given directly in the joint space. TORO is using its legs to maintain balance, while the 12 joints of the arms are forced on a 1 - dimensional submanifold defined by the constraint function

$$y_i(q) = q_i - q_{d_i} - q_{c_i} \qquad i = 2, \ldots, 12 \,, \tag{8.20}$$

where q_c is used to couple the elbow joint of the right arm to the first joint of the shoulder of the same arm, i.e. $q_{c_4} = 5q_1$ while the rest of the entries are zero. Finally q_d is chosen to be the desired configuration shown in Fig. 8.4. It is worth to notice that choosing $q_{d_1} = 0$, then the condition $y(q_d) = 0$ is satisfied.

Potential function A simple choice for the virtual potential is given by

$$U(q) = \frac{1}{2} k_n \left\| q - q_d \right\|^2 \,, \tag{8.21}$$

where $k_n = 40\,\mathrm{Nm/rad}$. $U(q)$ so defined is clearly positive definite on the 1 - dimensional submanifold and with its minimum at q_d.

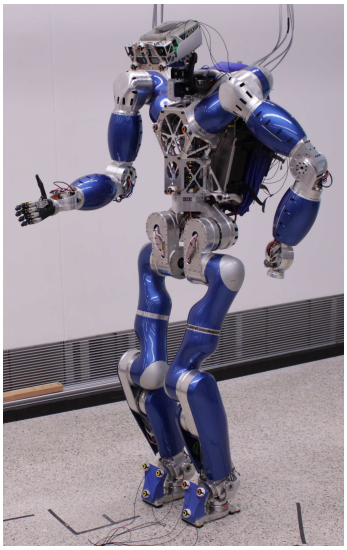

Figure 8.4.: Desired configuration.

Results Given the definition of $\boldsymbol{y}(\boldsymbol{q})$, the resulting motion will be an oscillation similar to a handshake in which the first shoulder joint works as limit cycle generator with the elbow joint coupled to it, while the remaining ones will keep the desired position.

In order to highlight the role of the friction compensation the gain K_H is set very low ($K_H = 5\,\mathrm{s/rad^2}$). Because of that, the convergence of the energy error is poor and, in case of no compensation, the friction is the predominant effect and no oscillation is achieved, although the robot is pushed three times by the experimenter. The plot of the energy is shown in Fig. 8.5, where the three high peaks correspond to the three interactions between the robot and the experimenter. On the other hand, using the friction compensation, the energy is oscillating around the desired value after few seconds, as shown in Fig. 8.6. In both cases the value of the desired energy is $H_d = 0.2\,\mathrm{J}$.

Table 8.1.: Gains used for one arm in the experiment.

Parameter	Value	Unit
κ	30	$1/\mathrm{s}$
$\boldsymbol{Q}_s$	0.07	$1/\mathrm{s}$
$\boldsymbol{D}_S$	$\mathrm{diag}(32,60,50,66,51,51)$	$1/\mathrm{s}$
$\boldsymbol{K}_s$	$\mathrm{diag}(5,2.5,2,1.5,1,1)$	$\mathrm{Nms/rad}$
$\boldsymbol{P}$	$\mathrm{diag}(10,\ldots,10,1,\ldots,1)$	$\mathrm{rad^2/Nms^2}$

The values of the gains used for the experiment are listed in Table 8.1.

Discussion In Fig. 8.6 the energy oscillates around its desired value, although convergence was theoretically expected. Measurement noise, flexibility of the structure and model uncertainties are a possible cause for this phenomenon. The mismatch between the real model and the one used by the controller is supposed to be the main reason for the oscillations in the recorded signal. The behavior can, in fact, be reproduced in simulation when using an incorrect model for the controller and disappears when the model is perfectly known, as it can be seen in Fig. 8.7. In the simulation the robot starts with a higher deviation from the submanifold compared to the

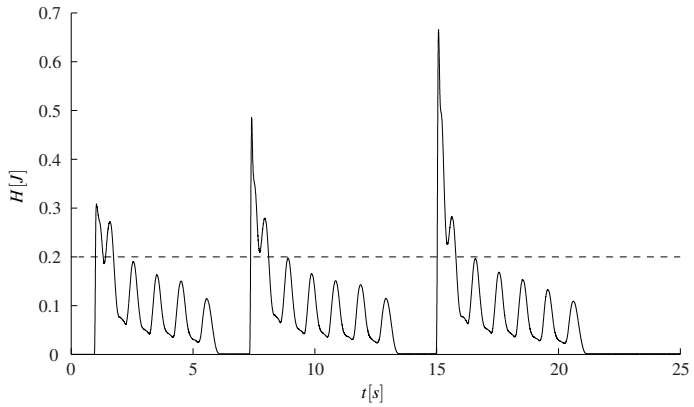

Figure 8.5.: Energy function (continuous line) and desired value (dashed line) without friction compensation obtained in the experiment.

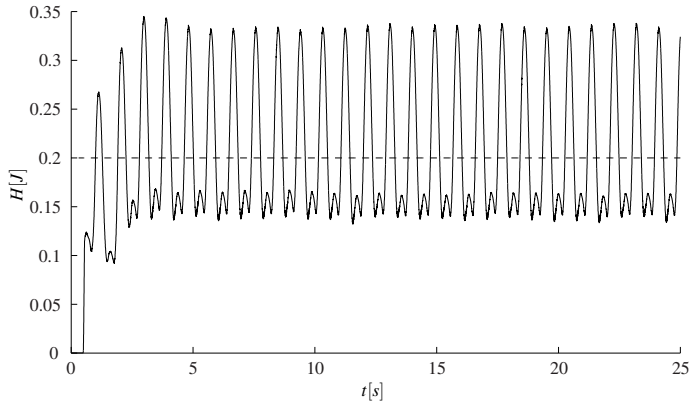

Figure 8.6.: Energy function (continuous line) and desired value (dashed line) with friction compensation obtained in the experiment.

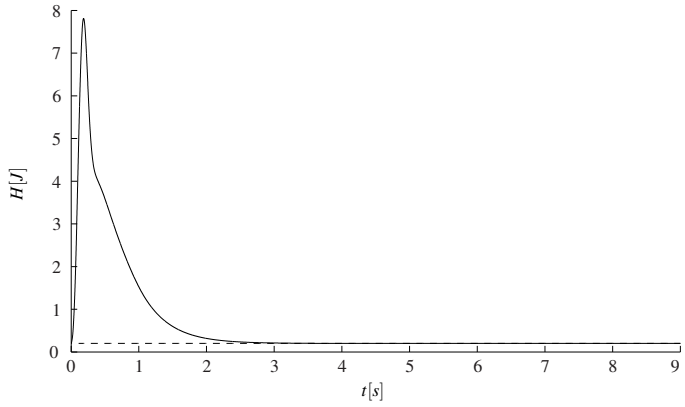

Figure 8.7.: Energy function (continuous line) and desired value (dashed line) with friction compensation obtained in simulation.

experiment. This causes the big overshoot[2] in the plot, since the total kinetic energy is used in the definition of $H(\boldsymbol{q}, \dot{\boldsymbol{q}})$, but still a perfect convergence is achieved. The friction model used is as in (8.19), where the values of the parameters are unknown for the same controller used for the experimental setup. Uncertainty in the dynamic matrices are not considered by the current state of the controller. Nevertheless, the experiment gives the opportunity to test the robustness of the proposed approach. Although it was not formally shown, the behavior results to be periodic even if the energy error does not converge exactly to zero, showing that the effect of the model uncertainty is a distortion of the expected attractive closed orbit.

8.5. Summary

In this chapter, the problem of generating attractive closed orbits for rigidly actuated manipulators in presence of unmodeled disturbances was addressed. In order to cope with friction always present in real systems, the results presented in chapter 4 were extended using a combination of state of the art approaches for uncertain models. Conventional friction compensation techniques cannot be applied directly if the design of the control is required to be kept separated in two steps, as in the frictionless case. The reason is that no interference between the two parts of the controller is allowed when compensating the friction. The approach has been validated in simulation and experiment using the humanoid robot TORO, confirming the improvement in the performances compared to the previous approach.

[2]During this phase the term proportional to the energy error is acting like a damping, so no unexpected oscillations will be produced.

CHAPTER 9

Conclusion

The last decades have seen a steadily increasing attention from the robotic community into the design of innovative drive concepts, that can allow the systems to perform highly dynamical tasks. Elastically actuated robots changed how elasticity has been usually considered in robotics. While the classical mechanical and control design processes tended to reduce the elasticity to a minimum in order to simplify the analysis and control of the robots, the new actuation principles intentionally include elastic elements to profit from the possibility of storing energy and protect the motors from abrupt interaction between the links and the environment. As legged locomotion is demanding in terms of performant actuation, often the design of legged robots tends to include elastic joints. Numerous works in the biomechanics field have shown, additionally, that the presence of elasticity in the structure can explain the principles of locomotion. Therefore, the goal of this work was to propose a solution for exploiting the presence of elastic joints when used in a legged robots.

The contributions of this thesis can be summarized as:

- insights on the dynamic model of a free floating robot leading to a partially decoupled structure of the model
- methods to map a single mass template model (SLIP) on a multi-body robot
- design of a static state feedback control for orbital stabilization of rigidly actuated robots and extension to a robust version
- design of a dynamic state feedback control for orbital stabilization of elastically actuated robots
- adaptation of an existent optimization based approach for balancing in order to obtain a jumping pattern via energy regulation
- design of a control law for regulation of the end effector of a free floating robot via momentum decoupling
- extensions of the applicability of semidefinite Lyapunov functions, which are used in the stability analysis.

Although on the robotics scene since many years, it is still not clear how to use the full potential of elastic joints. The typical control strategies adopted are still the ones developed when the elasticity in the joints was an unwanted effect, i.e. this control laws tend to compensate for the presence of the spring, which is a disturbance to the desired perfectly rigid actuation. The main focus of this work was then to design control laws aware of the benefits that can be provided by the energy stored in the springs. With the mind on possible applications to the field of legged robotics, the requirement for the controller was to generate an asymptotically stable limit cycle, such that a periodic task could be realized. Jumping, running and walking can be seen, in fact, as periodic tasks. Once again, elasticity is intuitively helpful for producing an oscillation.

The SLIP template model is capable of extrapolating the main features of different gaits of humans and many different other animals. Methods to map this template model on a robot were considered, in order to realize a walking pattern. Already in this first stage it was required to provide the robot with a mechanism to recover from the energy loss at the impacts.

Starting from the simpler case of a rigidly actuated robot, a static state feedback was designed to produce an asymptotically stable limit cycle defined as the level set of an energy function. This is realized by the controller enforcing a set of virtual constraints and regulating at the same time the energy to the desired value. A version of the controller capable of compensating for a class of unmodeled disturbances was also considered. As in the rigid case there are no elastic elements in the joints, the energy function was defined as the sum of the kinetic energy and of a virtual potential introduced by the controller. When using elastically actuated robots, the virtual potential is replaced by the physical elastic energy and the gravitational potential energy. Nevertheless, aiming at keeping the same design idea to produce the limit cycle, i.e. using a combination of virtual constraints and energy regulation, the complexity of the controller had to be increased. The goal in this case is achieved by a dynamic state feedback, meaning that the controller is itself a dynamic system. Precisely, the controller dynamics are mimicking the motor ones. To the best of the knowledge of the author, this control law is the first one to show how the torques produced by the springs themselves can be directly used to fulfill the task. Classic approaches, instead, require that the torques produced by the springs track desired ones designed assuming to have rigid joints (e.g. backstepping), or assume to have sufficiently high joint stiffness in order to use the motor position directly as a control input (e.g. singular perturbation).

The validation of the controllers requires a dynamic library, implemented based on the close form expression and the iterative algorithm presented in the thesis. As a good understanding of the model is always helpful for the design of the control laws, the properties of the dynamic model of the robot were carefully examined. Particular attention was dedicated to the structure of the model of a free floating base robot, as biped robots are clearly not fixed on the floor. The conservation of generalized momentum was used to derive a block-diagonal structure of the model. This concepts were used to design a feedback law to control the end effector of a free floating robot to a desired pose. The case of a robotic arm mounted on a satellite was considered as application scenario, highlighting the similarities in the objectives and modeling between the field of legged and space robotics.

The capability of the energy controller to enforce oscillations was used to obtain periodic jumps with a biped robot. A toy model illustrated the principle behind the idea of having a jumping pattern based on an energy argument. To take into account the underactuation and the unilateral constraints inherent with the definition of a locomotion task, an optimization problem was set up. In particular, an optimization framework designed for balancing control was adapted in order to incorporate the energy controller and produce the desired jumping pattern. Loosely speaking, the idea is to use the energy controller assuming to have full actuation, while the optimization problem takes care of redistributing the interaction forces with the environment in order to ensure

the feasibility of the control action. Although resorting to the solution of an optimization problem prevents to carry on an analytic stability analysis of the closed loop system as it was done for all the other designed control laws, the use of optimization is the state of the art in legged robotics and it has been shown to be very effective in practice.

As the biggest part of this work dealt with the design of innovative control laws to exploit elastic joints, future works will focus on applications of the developed controllers for other locomotion patterns beside the already implemented jumping controller. The latter can be used, for example, as core for a running controller. Obviously, in this case particular attention must be dedicated to the planning phase. Footsteps have to be designed in order to guarantee the success of the control strategy. At the same time, a state machine has to orchestrate the sequence of different phases, i.e. double support, single support or flight phase. Finally, a model predictive control approach could be used to improve the optimization problem. Instead of manipulating the current values of the desired torques, one might consider using their time derivatives to guarantee the feasibility of the future values.

Appendices

APPENDIX A

Stability theorems

In this appendix, some useful notations are recalled. Moreover, the theorems presented are a contribution to the analysis of dynamical systems with semidefinite Lyapunov functions, as they extend known results for equilibrium points to the case of generic bounded invariant sets, i.e. including also limit cycles.

A.1. Definitions

Consider the system

$$\dot{\boldsymbol{\chi}} = \boldsymbol{f}(\boldsymbol{\chi}) \,, \tag{A.1}$$

where $\boldsymbol{\chi} \in \mathscr{X} \subset \mathbb{R}^m$ and $\boldsymbol{f} : \mathscr{X} \to \mathbb{R}^m$ is a Lipschitz continuous function, so that a unique solution exists. The solution of (A.1) starting at $\boldsymbol{\chi}_0$ and evaluated at the time instant t is denoted with $\boldsymbol{\chi}(t;\boldsymbol{\chi}_0)$, so that $\boldsymbol{\chi}(0;\boldsymbol{\chi}_0) = \boldsymbol{\chi}_0$.

In case $\boldsymbol{f} : \mathscr{X} \to \mathbb{R}^m$ is only a Lebesgue measurable function, then (A.1) is understood in the sense of Filippov [Fil88], i.e. (A.1) is replaced by the equivalent differential inclusion obtained using its convex regularization [Fil88]. Existence of solutions can be guaranteed with the notion of upper-continuity of set-valued functions [Fil88]. Additionally, $\boldsymbol{\chi}(t;\boldsymbol{\chi}_0)$ is the absolutely continuous solution satisfying the differential inclusion almost everywhere.

The following definitions apply to a bounded, invariant set Ω of the system (A.1).

Distance $\mathrm{d}(\boldsymbol{\chi},\Omega) \triangleq \min_{\boldsymbol{y}\in\Omega} \|\boldsymbol{\chi} - \boldsymbol{y}\|$

Open ball $B_\varepsilon(\Omega) \triangleq \{\boldsymbol{\chi} \in \mathscr{X} \mid \mathrm{d}(\boldsymbol{\chi},\Omega) < \varepsilon\}$

Closed ball $\bar{B}_\varepsilon(\Omega) \triangleq \{\boldsymbol{\chi} \in \mathscr{X} \mid \mathrm{d}(\boldsymbol{\chi},\Omega) \leq \varepsilon\}$

Sphere $S_\varepsilon(\Omega) \triangleq \{\boldsymbol{\chi} \in \mathscr{X} \mid \mathrm{d}(\boldsymbol{\chi},\Omega) = \varepsilon\}$

Stability Ω is stable if $\forall \varepsilon > 0$, $\exists\ \delta = \delta(\varepsilon) > 0$ such that $\forall \boldsymbol{\chi}_0 \in B_\delta(\Omega) \Rightarrow \boldsymbol{\chi}(t;\boldsymbol{\chi}_0) \in B_\varepsilon(\Omega)$, $\forall t \geq 0$

Asymptotic stability Ω is asymptotically stable if it is stable and $\exists\ \delta > 0$ such that $\forall \boldsymbol{\chi}_0 \in B_\delta(\Omega) \Rightarrow$
$\lim_{t\to\infty} \mathrm{d}(\boldsymbol{\chi}(t;\boldsymbol{\chi}_0),\Omega) = 0$

Conditional stability Ω is conditionally stable to $\mathscr{A}$ if $\Omega \subset \mathscr{A}$ and $\forall \varepsilon > 0$, $\exists\ \delta = \delta(\varepsilon) > 0$ such that $\forall \boldsymbol{\chi}_0 \in B_\delta(\Omega) \cap \mathscr{A} \Rightarrow \boldsymbol{\chi}(t;\boldsymbol{\chi}_0) \in B_\varepsilon(\Omega),\ \forall t \geq 0$

Conditional attractiveness Ω is attractive conditionally to $\mathscr{A}$ if $\Omega \subset \mathscr{A}$ and $\exists\ \delta > 0$ such that $\forall \boldsymbol{\chi}_0 \in B_\delta(\Omega) \cap \mathscr{A} \Rightarrow \lim_{t\to\infty} \mathrm{d}(\boldsymbol{\chi}(t;\boldsymbol{\chi}_0),\Omega) = 0$

Conditional asymptotic stability Ω is conditionally asymptotically stable to $\mathscr{A}$ if $\Omega \subset \mathscr{A}$ is stable conditionally to $\mathscr{A}$ and $\exists\ \delta = \delta(\varepsilon) > 0$ such that $\forall \boldsymbol{\chi}_0 \in B_\delta(\Omega) \cap \mathscr{A} \Rightarrow \lim_{t\to\infty} \mathrm{d}(\boldsymbol{\chi}(t;\boldsymbol{\chi}_0),\Omega) = 0$

A.1.1. Semidefinite Lyapunov functions

The results presented here extend those for equilibrium points that can be found in [IKO96,SJK97].

Theorem A.1 (Stability)**.** *Let Ω be a bounded, invariant set for* (A.1)*, and let $V(\boldsymbol{\chi})$ be a C^1 function such that $V(\boldsymbol{\chi}) \geq 0$, $V(\Omega) = 0$ and $\dot{V}(\boldsymbol{\chi}) \leq 0$. If Ω is asymptotically stable conditionally to $\mathscr{A} = \{\boldsymbol{\chi} \mid V(\boldsymbol{\chi}) = 0\}$, then Ω is stable.*

Proof. Suppose by contradiction that Ω is unstable. Then exist an ε such that $\nu > \varepsilon > 0$, a sequence $(\boldsymbol{\chi}_{0n})_{n\in\mathbb{N}} \subset B_\varepsilon(\Omega)$, $\lim_{n\to\infty} \mathrm{d}(\boldsymbol{\chi}_{0n},\Omega) = 0$, and a sequence $(t_n)_{n\in\mathbb{N}} \subset \mathbb{R}^+$ in such a way that

$$\begin{cases} \mathrm{d}(\boldsymbol{\chi}(t;\boldsymbol{\chi}_{0n}),\Omega) < \varepsilon & 0 \leq t < t_n \\ \mathrm{d}(\boldsymbol{\chi}(t_n;\boldsymbol{\chi}_{0n}),\Omega) = \varepsilon & \forall n \in \mathbb{N}\ . \end{cases} \tag{A.2}$$

Since $S_\varepsilon(\Omega)$ is compact, a convergent subsequence $\boldsymbol{y}'_n$ can be extracted from $\boldsymbol{y}_n = \boldsymbol{\chi}(t_n;\boldsymbol{\chi}_{0n})$ such that $\boldsymbol{y}'_n \to \boldsymbol{y} \in S_\varepsilon(\Omega)$ as $n \to \infty$. Moreover because of the continuity of the solutions of (A.1) and the invariance of Ω, $t_n \to \infty$ as $n \to \infty$.

Now, $V(\boldsymbol{\chi}(-t;\boldsymbol{y})) = 0$ has to be shown. Let $\tau < 0$ and $N \in \mathbb{N}$ be such that $0 < t_n + \tau < t_n,\ \forall n \geq N$. Because V is not increasing along the solutions of (A.1), it follows that

$$0 \leq V(\boldsymbol{\chi}(t_n + \tau;\boldsymbol{\chi}_{0n})) \leq V(\boldsymbol{\chi}_{0n})\ . \tag{A.3}$$

From $\lim_{n\to\infty} \mathrm{d}(\boldsymbol{\chi}_{0n},\Omega) = 0$, $V(\Omega) = 0$ and the continuity of V, it follows

$$V(\boldsymbol{\chi}(\tau;\boldsymbol{y})) = \lim_{n\to\infty} V(\boldsymbol{\chi}(\tau;\boldsymbol{\chi}(t_n;\boldsymbol{\chi}_{0n}))) = \lim_{n\to\infty} V(\boldsymbol{\chi}(t_n + \tau;\boldsymbol{\chi}_{0n})) = 0\ . \tag{A.4}$$

It remains to prove that $\boldsymbol{\chi}(-t;\boldsymbol{y}) \in \mathscr{A}$ and $\mathrm{d}(\boldsymbol{y},\Omega) = \varepsilon$ cannot hold if Ω is asymptotically stable conditionally to $\mathscr{A}$. Since Ω is asymptotically stable conditionally to $\mathscr{A}$, $\exists\ T = T(\varepsilon) > 0$ such that $\mathrm{d}(\boldsymbol{\chi}(T;\boldsymbol{\chi}_0),\Omega) \leq \frac{\varepsilon}{2}$, with $\boldsymbol{\chi}_0 \in \mathscr{A}$. If one chooses $\boldsymbol{\chi}_0 = \boldsymbol{\chi}(-T;\boldsymbol{y}) \in \mathscr{A}$, then

$$\frac{\varepsilon}{2} \geq \mathrm{d}(\boldsymbol{\chi}(T;\boldsymbol{\chi}_0),\Omega) = \mathrm{d}(\boldsymbol{\chi}(0;\boldsymbol{y}),\Omega) = \mathrm{d}(\boldsymbol{y},\Omega) = \varepsilon\ .$$

Since this is a contradiction, it can be concluded that Ω must be stable. □

Thanks to the properties of positive semidefinite functions it is possible to ensure not just the stability, but also the asymptotic stability of an invariant set Ω. To this end, first two results are recalled.

Lemma A.1 ([IKO96]). *Let V be a nonnegative function with $\dot{V}(\chi) \leq 0$, then $\mathscr{A} = \{\chi \mid V(\chi) = 0\}$ is a positively invariant set and $\mathscr{A} \subset E = \left\{\chi \mid \dot{V}(\chi) = 0\right\}$.*

Lemma A.2 ([Kha02]). *If a solution $\chi(t;\chi_0)$ of* (A.1) *is bounded and belongs to $\mathscr{X}$ for $t \geq 0$, then its positive limit set $Ł^+$ is a nonempty, compact, invariant set. Moreover, $\chi(t;\chi_0)$ approaches $Ł^+$ as $t \to \infty$*

Theorem A.2 (Asymptotic stability). *Let Ω be a bounded, invariant set for* (A.1)*, and let $V(\chi)$ be a C^1 function such that $V(\chi) \geq 0$, $V(\Omega) = 0$ and $\dot{V}(\chi) \leq 0$. If Ω is asymptotically stable conditionally to the largest positively invariant set $\mathscr{M}$ within $E = \left\{\chi \mid \dot{V}(\chi) = 0\right\}$, then Ω is asymptotically stable.*

Proof. In order to prove asymptotic stability one has to show stability and attractiveness.

From Lemma A.1 it follows that $\mathscr{A}$ is a positively invariant set and $\mathscr{A} \subset E$, so since Ω is conditionally stable to $\mathscr{M}$ and $V(\Omega) = 0$ i.e. $\Omega \subset \mathscr{A}$, then it must be conditionally stable to $\mathscr{A}$, hence Ω is stable by Theorem A.1.

Attractiveness is proven by contradiction. Since Ω is stable then $\forall \varepsilon > 0$, $\exists\ \delta = \delta(\varepsilon) > 0$ such that $\forall \chi_0 \in B_\delta(\Omega) \Rightarrow \chi(t;\chi_0) \in B_\varepsilon(\Omega)$, $\forall t \geq 0$. This means that using also Lemma A.2, then $Ł^+$ is a positively invariant set and $Ł^+ \in B_\varepsilon(\Omega) \cap \mathscr{M}$. Now, assume by contradiction that $Ł^+$ is not Ω. Since Ω is asymptotically stable conditionally to $\mathscr{M}$, then $\lim_{t\to\infty} \mathrm{d}(\chi(t;\chi_0), \Omega) = 0$ if $\chi_0 \in B_\varepsilon(\Omega) \cap \mathscr{M}$. Choosing $\chi_0 = y \in Ł^+ \neq \Omega$ one reaches a contradiction. □

Theorem A.3 (Convergence with semidefinite functions). *Let Ω be a bounded, invariant set for* (A.1)*, and let $V(\chi)$ be a differentiable function such that $V(\chi) \geq 0$, $V(\Omega) = 0$ and $\dot{V}(\chi) \leq 0$. If Ω is attractive conditionally to the largest positively invariant set $\mathscr{M}$ within $E = \left\{\chi \mid \dot{V}(\chi) = 0\right\}$ and the solutions are bounded, then Ω is attractive.*

Proof. Let $\alpha = \min_{\chi \in S_v(\Omega)} V(\chi)$, $\alpha > 0$, and take β such that $0 < \beta < \alpha$. Then the set $\Omega_\beta = \{\chi \in B_v(\Omega) \mid V(\chi) \leq \beta\}$ is a compact positively invariant set and by virtue of LaSalle's invariance theorem then every solution starting in Ω_β approaches $\mathscr{M}$ as $t \to \infty$. Since the solutions are bounded, then $\exists\ \delta = \delta(\chi_0) > 0$ such that $\chi(t;\chi_0) \in B_\delta(\Omega)$, $\forall t \geq 0$. Using Lemma A.2, the positive limit set $Ł^+$ is such that $Ł^+ \in B_\delta(\Omega) \cap \mathscr{M}$. Additionally, since Ω is attractive conditionally to $\mathscr{M}$, then $\lim_{t\to\infty} \mathrm{d}(\chi(t;\chi_0), \Omega) = 0$ if $\chi_0 \in B_\delta(\Omega) \cap \mathscr{M}$. Now, assume by contradiction that $Ł^+$ is not Ω. Choosing $\chi_0 = y \in Ł^+ \neq \Omega$ one reaches a contradiction. □

Bibliography

[AB06] Mojtaba Ahmadi and Martin Buehler. Controlled passive dynamic running experiments with the ARL-Monopod II. *IEEE Trans. on Robotics*, 22(5):974–986, 2006.

[ABQ+99] Frank Allgöwer, Thomas A. Badgwell, Joe S. Qin, James B. Rawlings, and Stephen J. Wright. *Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview*, pages 391–449. Springer London, London, 1999.

[Ale90] Robert McNeill Alexander. Three uses for springs in legged locomotion. *Int. Journal of Robotics Research*, 9(2):53–61, 1990.

[APE04] Christine Azevedo, Philippe Poignet, and Bernard Espiau. Artificial locomotion control: from human to robots. *Robotics and Autonomous Systems*, 47(4):203–223, 2004.

[AS10] Mustafa Mert Ankarali and Uluc Saranli. Analysis and control of a dissipative spring-mass hopper with torque actuation. In *Robotics: Science and Systems*, pages 41–48, Zaragoza, Spain, June 2010.

[ASH01] Alin Albu-Schäffer and Gerd Hirzinger. A globally stable state-feedback controller for flexible joint robots. *Advanced Robotics*, 15(8):799–814, 2001.

[ASOH07] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. Journal of Robotics Research*, 26(1):23–39, 2007.

[ASOP12] Alin Albu-Schäffer, Christian Ott, and Florian Petit. Energy shaping control for a class of underactuated Euler-Lagrange systems. In *Proc. 10th Int. IFAC Symp. Robot Control*, pages 567–575, Dubrovnik, Croatia, September 2012.

[Bai85] John Baillieul. Kinematic programming alternatives for redundant manipulators. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 722–728, St. Louis, USA, March 1985.

[Bli89] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22:1217–1227, 1989.

[BOL95] Bernard Brogliato, Romero Ortega, and Rogelio Lozano. Globally stable nonlinear controllers for flexible joint manipulators: a comparative study. *Automatica*, 31(7):941–956, 1995.

[Bro99] Bernard Brogliato. *Nonsmooth Mechanics*. Springer-Verlag, London, 1999.

[CEU02] Carlos Canudas-de-Wit, Bernard Espiau, and Claudio Urrea. Orbital stabilization of underactuated mechanical systems. In *Triennial World Congress of the International Federation of Automatic Control*, pages 893–898, Barcelona, Spain, July 2002.

[CSB96] Carlos Canudas-de-Wit, Bruno Siciliano, and Georges Bastin. *Theory of robot control*. Springer-Verlag, London, 1996.

[CX99] Chieh-Li Chen and Rui-Lin Xu. Tracking control of robot manipulator using sliding mode controller with performance robustness. *ASME Journal of Dynamic Systems, Measurement, and Control*, 121(1):64–70, 1999.

[DCA05] Dalila Djoudi, Christine Chevallereau, and Yannick Aoustin. Optimal reference motions for walking of a biped robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2002–2007, Barcelona, Spain, April 2005.

[De 88] Alessandro De Luca. Dynamic control of robots with joint elasticity. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 152–158, Philadelphia, USA, April 1988.

[DF11] Alessandro De Luca and Fabrizio Flacco. A PD-type regulator with exact gravity cancellation for robots with flexible joints. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 317–323, Shanghai, China, May 2011.

[DL98] Alessandro De Luca and Pasquale Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 504–510, Leuven, Belgium, May 1998.

[DOAS13] Alexander Dietrich, Christian Ott, and Alin Albu-Schäffer. Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3043–3050, Tokyo, Japan, November 2013.

[DS03] Vincent Duindam and Stefano Stramigioli. Passive asymptotic curve tracking. In *Proceedings of the IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, pages 229–234, Seville, Spain, April 2003.

[DS04] Vincent Duindam and Stefano Stramigioli. Port-based asymptotic curve tracking for mechanical systems. *European Journal of Control*, 10:411–420, 2004.

[EGB09] Michael Ernst, Hartmut Geyer, and Reinhard Blickhan. Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant. In *Proc 12th Int Conf on Climbing and Walking Robots (CLAWAR)*, pages 639–644, Baltimore, USA, July 2009.

[EOR+11] Johannes Englsberger, Christian Ott, Maximo A. Roa, Alin Albu-Schäffer, and Gerd Hirzinger. Bipedal walking control based on capture point dynamics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4420–4427, San Francisco, USA, September 2011.

[EWO+14] Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, and Alin Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In *IEEE/RAS Int. Conf. on Humanoid Robots*, pages 916–923, Madrid, Spain, November 2014.

[Fea87] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, Boston, 1987.

[Fil88] Aleksej Fedorovič Filippov. *Differential equations with discontinuous righthand sides*. Mathematics and its applications. Kluwer Academic, Dordrecht, 1988.

[Fra04] Theodore Frankel. *The Geometry of Physics: An Introduction*. Cambridge University Press, New York, 2004.

[GASB+11] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, Hannes Höppner, Stefan Jörg, Mathias Nickl, Alexander Nothhelfer, Florian Petit, Josef Reill, Nikolaus Seitz, Thomas Wimböck, Sebastian Wolf, Tilo Wüsthoff, and Gerd Hirzinger. The DLR Hand Arm System. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3175–3182, Shanghai, China, May 2011.

[GCH+13] Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A. Hoepflinger, and Roland Siegwart. Control of dynamic gaits for a quadrupedal robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3287–3292, Karlsruhe, Germany, May 2013.

[GEO15] Gianluca Garofalo, Johannes Englsberger, and Christian Ott. On the regulation of the energy of elastic joint robots: excitation and damping of oscillations. In *American Control Conference (ACC)*, pages 4825–4831, Chicago, USA, July 2015.

[GGdS+16] Alessandro Massimo Giordano, Gianluca Garofalo, Marco de Stefano, Christian Ott, and Alin Albu-Schäffer. Dynamics and control of a free-floating space robot in presence of nonzero linear and angular momenta. In *IEEE Conf. on Decision and Control*, page to appear, 2016.

[GHEO15] Gianluca Garofalo, Bernd Henze, Johannes Englsberger, and Christian Ott. On the inertially decoupled structure of the floating base robot dynamics. In *8th Vienna International Conference on Mathematical Modelling (2015)*, pages 322–327, Vienna, Austria, February 2015.

[GHS89] Fathi Ghorbel, John Y. Hung, and Mark W. Spong. Adaptive control of flexible joint manipulators. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1188–1193, Scottsdale, USA, May 1989.

[GO16a] Gianluca Garofalo and Christian Ott. Limit cycle control using energy function regulation with friction compensation. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):90–97, 2016.

[GO16b] Gianluca Garofalo and Christian Ott. Steps towards energy efficiency in elastically actuated robots. In Mohammad O. Tokhi and Gurvinder S. Virk, editors, *Advances in Cooperative Robotics: Proceedings of the 19th International Conference on CLAWAR 2016*, pages 780–782, London, UK, September 2016. World Scientific.

[GO17] Gianluca Garofalo and Christian Ott. Energy based limit cycle control of elastically actuated robots. *IEEE Trans. on Automatic Control*, page to appear, 2017.

[GOAS12a] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Walking control of fully actuated robots based on the bipedal SLIP model. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1999–2004, Saint Paul, USA, May 2012.

[GOAS12b] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Walking control of fully actuated robots based on the bipedal SLIP model. In *Proceedings of the Conference on Dynamic Walking*, Pensacola, USA, May 2012.

[GOAS13a] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Asymptotically stable limit cycles generation by using nullspace decomposition and energy regulation. In *DGR-Tage*, Munich, Germany, October 2013.

[GOAS13b] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. On the closed form computation of the dynamic matrices and their differentiations. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2364–2359, Tokyo, Japan, November 2013.

[GOAS13c] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Orbital stabilization of mechanical systems through semidefinite Lyapunov functions. In *American Control Conference (ACC)*, pages 5735–5741, Washington DC, USA, June 2013.

[GPB+11] Manolo Garabini, Andrea Passaglia, Felipe Belo, Paolo Salaris, and Antonio Bicchi. Optimality principles in variable stiffness control: The VSA Hammer. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3770–3775, San Francisco, USA, September 2011.

[GSB06] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Compliant leg behavior explains basic dynamics of walking and running. *Proceedings of the Royal Society B*, 273:2861–2867, November 2006.

[HDO16] Bernd Henze, Alexander Dietrich, and Christian Ott. An approach to combine balancing with hierarchical whole-body control for legged humanoid robots. *IEEE Robotics and Automation Letters (RA-L)*, 1(2):700–707, 2016.

[HHC07] Sang-Ho Hyon, Joshua G. Hale, and Gordon Cheng. Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Trans. on Robotics*, 23(5):884–898, 2007.

[Hol80] John M. Hollerbach. A recursive Lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. on Systems, Man and Cybernetics*, 10:730–736, 1980.

[HOR14] Bernd Henze, Christian Ott, and Maximo Alejandro Roa. Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3253–3258, Chicago, USA, September 2014.

[HRHS10] Marco Hutter, David Remy, Mark A. Höpflinger, and Roland Siegwart. SLIP running with an articulated robotic leg. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4934–4939, Taipei, Taiwan, October 2010.

[HWR+14] Bernd Henze, Alexander Werner, Maximo Alejandro Roa, Gianluca Garofalo, Johannes Englsberger, and Christian Ott. Control applications of toro - a torque controlled humanoid robot. In *IEEE/RAS Int. Conf. on Humanoid Robots*, Madrid, Spain, November 2014.

[IKO96] A. Iggidr, B. Kalitine, and R. Outbib. Semidefinite Lyapunov functions: Stability and stabilization. In *Mathematics of Control, Signals, and Systems*, pages 95–106, 1996.

[Isi95] Alberto Isidori. *Nonlinear control systems*. Springer-Verlag, Berlin, 1995.

[Kha02] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, New Jersey, 2002.

[KK85] Khashayar Khorasani and Petar V. Kokotović. Feedback linearization of a flexible manipulator near its rigid body manifold. *Systems and Control Letters*, 6(3):187–192, 1985.

[KKH10] Devin Koepl, Kevin Kemper, and Jonathan Hurst. Force control for spring-mass walking and running. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 639–644, Montreal, Canada, July 2010.

[KKK+03] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1644–1650, Las Vegas, USA, October 2003.

[KOK+73] Ichiro Kato, Sadamu Ohteru, Hiroshi Kobayashi, Katsuhiko Shirai, and Akihiko Uchiyama. Information-power machine with senses and limbs. In *Proc. CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, pages 12–24, Udine, Italy, September 1973.

[KW11] JG Daniël Karssen and Martijn Wisse. Running with improved disturbance rejection by using non-linear leg springs. *Int. Journal of Robotics Research*, 30(13):1585–1595, 2011.

[LB92] Rogelio Lozano and Bernard Brogliato. Adaptive control of robot manipulators with flexible joints. *IEEE Trans. on Automatic Control*, 37(2):174–181, 1992.

[LGDAS14] Dominic Lakatos, Gianluca Garofalo, Alexander Dietrich, and Alin Albu-Schäffer. Jumping control for compliantly actuated multilegged robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4562–4568, Hong Kong, China, May 2014.

[LGP+13] Dominic Lakatos, Gianluca Garofalo, Florian Petit, Christian Ott, and Alin Albu-Schäffer. Modal limit cycle control for variable stiffness actuated robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4934–4941, Karlsruhe, Germany, May 2013.

[Lib03] Daniel Liberzon. *Switching in Systems and Control*. Birkhäuser, Boston, 2003.

[McG90] Tad McGeer. Passive dynamic walking. *Int. Journal of Robotics Research*, 9:62–82, 1990.

[MLS94] Richard M. Murray, Zexiang Li, and Shankar S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[MM07] Juan Ignacio Mulero-Martinez. Uniform bounds of the Coriolis/centripetal matrix of serial robot manipulators. *IEEE Trans. on Robotics*, 23(5):1083–1089, 2007.

[MMA89] Yasuhiro Masutani, Fumio Miyazaki, and Suguru Arimoto. Sensory feedback control for space manipulators. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1346–1351, Scottsdale, USA, May 1989.

[MSC12] Bruce D. Miller, John Schmitt, and Jonathan E. Clark. Quantifying disturbance rejection of SLIP-like running systems. *Int. Journal of Robotics Research*, 31(5):573–587, 2012.

[Mül07] Andreas Müller. Partial derivatives of the inverse mass matrix of multibody systems via its factorization. *IEEE Trans. on Robotics*, 23(1):164–168, 2007.

[MW06] Xiuping Mu and Qiong Wu. On impact dynamics and contact events for biped robots via impact effects. *IEEE Trans. on Systems, Man and Cybernetics*, 36(6):1364–1372, 2006.

[Nie90] Günter Dieter Niemeyer. *Computational algorithms for adaptive robot control*. PhD thesis, MIT, 1990.

[NP11] Kostas Nanos and Evangelos Papadopoulos. On the use of free-floating space robots in the presence of angular momentum. *Intelligent Service Robotics*, 4(1):3–15, 2011.

[NT88] Salvatore Nicosia and Patrizio Tomei. On the feedback linearization of robots with elastic joints. In *IEEE Conf. on Decision and Control*, pages 180–185, Austin, USA, December 1988.

[NT93] Salvatore Nicosia and Patrizio Tomei. Design of global tracking controllers for flexible-joint robots. *Journal of Robotic Systems*, 10(6):835–846, 1993.

[OASKH08] Christian Ott, Alin Albu-Schäffer, Andreas Kugi, and Gerd Hirzinger. On the passivity-based impedance control of flexible joint robots. *IEEE Trans. on Robotics*, 24(2):416–429, 2008.

[ODAS15] Christian Ott, Alexander Dietrich, and Alin Albu-Schäffer. Prioritized multi-task compliance control of redundant manipulators. *Automatica*, 53:416–423, 2015.

[OGL13] David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35:161–176, 2013.

[OKN08] Christian Ott, Andreas Kugi, and Yoshihiko Nakamura. Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite Lyapunov functions. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1456–1463, Pasadena, USA, May 2008.

[OMVH79] David E. Orin, R. McGhee, M. Vukobratović, and G. Hartoch. Kinematic and kinetic analysis of open-chain linkage utilizing Newton-Euler methods. *Mathematical Biosciences*, 43:107–130, 1979.

[ORH11] Christian Ott, Maximo Alejandro Roa, and Gerd Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *IEEE/RAS Int. Conf. on Humanoid Robots*, pages 26–33, Bled, Slovenia, October 2011.

[OS88] Romero Ortega and Mark W. Spong. Adaptive motion control of rigid robots: a tutorial. In *IEEE Conf. on Decision and Control*, pages 1575–1584, Texas, USA, December 1988.

[Ott08] Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Springer Tracts in Advanced Robotics. Springer-Verlag, Berlin, 2008.

[PBP95] Frank C. Park, James E. Bobrow, and Scott R. Ploen. A Lie group formulation of robot dynamics. *Int. Journal of Robotics Research*, 14:609–618, 1995.

[PCDG06] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *IEEE/RAS Int. Conf. on Humanoid Robots*, pages 200–207, Genova, Italy, December 2006.

[PCY99] Jonghoon Park, Wankyun Chung, and Youngil Youm. On dynamical decoupling of kinematically redundant manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1495 –1500, 1999.

[PD93] Evangelos Papadopoulos and Steven Dubowsky. Dynamic singularities in free-floating space manipulators. *ASME Journal of dynamic systems, measurement, and control*, 115(1):44–52, 1993.

[PF99] Anatoli A. Pervozvanski and Leonid B. Freidovich. Robust stabilization of robotic manipulators by PID controllers. *Dynamics and Control*, 9(3):203–222, 1999.

[PP88] Brad Paden and Ravi Panja. A globally asymptotically stable 'PD+' controller for robot manipulators. *International Journal of Control*, 47(6):1697–1712, 1988.

[PPK15] Hae-Won Park, Sangin Park, and Sangbae Kim. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D running of MIT Cheetah 2. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5163–5170, Seattle, USA, May 2015.

[PVAL+03] Vicente Parra-Vega, Suguru Arimoto, Yun-Hui Liu, Gerhard Hirzinger, and Prasad Akella. Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans. Robotics and Automation*, 19(6):967–976, 2003.

[PW95] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 399–406, Pittsburgh, USA, August 1995.

[PŽGO13] Tadej Petrič, Leon Žlajpah, Gianluca Garofalo, and Christian Ott. Walking control using adaptive oscillators combined with dynamic movement primitives. In *22nd International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, pages 204–211, Portorož, Slovenia, September 2013.

[PŽGO14] Tadej Petrič, Leon Žlajpah, Gianluca Garofalo, and Christian Ott. Walking with adaptive oscillator and dynamic movement primitives. *International Journal of Mechanics and Control*, 15(1):3–10, 2014.

[Rai86] Marc H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, 1986.

[RBS10] Juergen Rummel, Yvonne Blum, and Andre Seyfarth. Robust and efficient walking with spring-like legs. *Bioinspiration and Biomimetics*, 5(4):046004, 2010.

[Rod87] G. Rodriguez. Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE Journal Robotics and Automation*, 3:624–639, 1987.

[RS87] Robert E. Roberson and Richard Schwertassek. *Dynamics of Multibody Systems*. Springer-Verlag, Berlin, 1987.

[SB01] Garett A. Sohl and James E. Bobrow. A recursive multibody dynamics and sensitivity algorithm for branched kinematic chains. *ASME Journal of Dynamic Systems, Measurement, and Control*, 123:391–399, 2001.

[SGGB02] Andre Seyfarth, Hartmut Geyer, Michael Günther, and Reinhard Blickhan. A movement criterion for running. *Journal of Biomechanics*, 35(5):649–655, 2002.

[SH90] Nader Saddegh and Roberto Horowitz. Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *Int. Journal of Robotics Research*, 9(3):74–92, 1990.

[Sil82] William M. Silver. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. *Int. Journal of Robotics Research*, 1:60–70, 1982.

[SJK97] Rodolphe Sepulchre, Mrdjan Jankovic, and Petar V. Kokotović. *Constructive Nonlinear Control*. Springer, Berlin, 1997.

[SL87] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. *Int. Journal of Robotics Research*, 6:49–59, 1987.

[SL91] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, New Jersey, 1991.

[SLT92] S. M. Shahruz, G. Langari, and M. Tomizuka. Design of robust PD-type control laws for robotic manipulators with parametric uncertainties. In *American Control Conference (ACC)*, pages 2967–2968, Chicago, USA, June 1992.

[Spo87] Mark W. Spong. Modeling and control of elastic joint robots. *the ASME Journal of Dynamic Systems, Measurement, and Control*, 109:310–318, 1987.

[Spo89] Mark W. Spong. Adaptive control of flexible joint manipulators. *Systems and Control Letters*, 13(1):15–21, 1989.

[SRS88] Hebertt Sira-Ramirez and Mark W. Spong. Variable structure control of flexible joint manipulators. *International Journal of Robotics and Automation*, 3(2):57–64, 1988.

[SSVO08] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2008.

[Str15] Stefano Stramigioli. Energy-aware robotics. In M. Kanat Camlibel, A. Agung Julius, Ramkrishna Pasumarthy, and Jacquelien M.A. Scherpen, editors, *Mathematical Control Theory I: Nonlinear and Hybrid Control Systems*, Lecture Notes in Control and Information Sciences. Springer International Publishing, Switzerland, 2015.

[SV76] Yurij Stepanenko and Miomir Vukobratović. Dynamics of articulated open-chain active mechanisms. *Mathematical Biosciences*, 28:137–170, 1976.

[SWA+02] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2478–2483, Lausanne, Switzerland, October 2002.

[SY05] Sophie Sakka and Kazuhito Yokoi. Humanoid vertical jumping based on force feedback and inertial forces optimization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3752–3757, Barcelona, Spain, April 2005.

[SYLM08] Wael Suleiman, Eiichi Yoshida, Jean-Paul Laumond, and Andrë Monin. Optimizing humanoid motions using recursive dynamics and Lie groups. In *International Conference on Information and Communication Technologies: from Theory to Applications (ICTTA)*, pages 1–6, Damascus, Syria, April 2008.

[TF09] Mitsuru Taniguchi and Kenji Fujimoto. Asymptotic path following and velocity control of port-Hamiltonian systems. In *European Control Conference (ECC)*, pages 236 – 241, Budapest, Hungary, August 2009.

[Tom91] Patrizio Tomei. A simple PD controller for robots with elastic joints. *IEEE Trans. on Automatic Control*, 36(10):1208–1213, 1991.

[Uic65] John J. Uicker. *On the Dynamic Analysis of Spatial Linkages Using 4x4 Matrices*. PhD thesis, Northwestern University, 1965.

[Utk92] Vadim Utkin. *Sliding modes in control and optimization*. Springer-Verlag, New York, 1992.

[UY87] Yoji Umetani and Kazuya Yoshida. Continuous path control of space manipulators mounted on OMV. *Acta Astronautica*, 15:981–986, 1987.

[UY89] Yoji Umetani and Kazuya Yoshida. Resolved motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Trans. Robotics and Automation*, 5:303–314, 1989.

[VB05] Miomir Vukobratović and Branislav Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 2(2):225–227, 2005.

[Ver74] A. F. Vereshchagin. Computer simulation of the dynamics of complicated mechanisms of robot manipulators. *Engineering Cybernetics*, 6:65–70, 1974.

[WC06] Pierre-Brice Wieber and Christine Chevallereau. Online adaptation of reference trajectories for the control of walking systems. volume 54, pages 559–566, 2006.

[WEH11] Sebastian Wolf, Oliver Eiberger, and Gerd Hirzinger. The DLR FSJ: Energy based design of a variable stiffness joint. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5082–5089, Shanghai, China, May 2011.

[WGC+07] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.

[Yos90] Tsuneo Yoshikawa. *Foundations of Robotics: Analysis and Control*. MIT Press, Cambridge, 1990.

[YSKU92] Kazuya Yoshida, Naoki Sashida, Ryo Kurazume, and Yoji Umetani. Modeling of collision dynamics for space free-floating links with extended generalized inertia tensor. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 899–904, Nice, France, May 1992.