

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA
In Automatica e Ricerca Operativa

Ciclo XX

Settore scientifico disciplinari di afferenza: ING-INF/04

TITOLO TESI

Passive Simulation and Interconnection:
Application to Haptic and Teleoperation Systems

Presentata da: **Borghesan Gianni**

Coordinatore Dottorato

Prof. Claudio Melchiorri



Relatore

Prof. Claudio Melchiorri



Esame finale anno 2008

Passive Simulation and Interconnection:
Application to Haptic and Teleoperation Systems

Borghesan Gianni

March 8, 2008

Contents

1	Introduction	5
1.1	Haptic interfaces	5
1.2	Teleoperation systems	7
2	Port Hamiltonian Framework	9
2.1	Physical systems	9
2.2	Explicit Port Hamiltonian Systems	12
2.3	Passivity and Port Hamiltonian Systems	13
3	Discrete Time Port Hamiltonian System	15
3.1	A passivity preserving numerical simulation algorithm	20
3.2	Update Strategy	21
4	Fault Conditions: a Unified Treatment	27
4.1	Dinamic Deadlock	28
4.2	Energy Leap	28
4.3	Unified treatment of fault conditions	29
4.4	Unilateral spring	31
5	Passive Interconnection	35
5.1	Passive Continuous-Discrete time Connection	36
5.2	Passive up/down sampling Interconnection	40
6	Simulation	43
6.1	Single rate Port Hamiltonian systems	43
6.2	Multirate Port Hamiltonian Systems	46
6.2.1	Simulation of multi-rate system	46
6.3	Bouncing ball: the PCDC and unilateral spring blocks	48

7 Experiments	51
7.1 Description of the experimental setup	51
7.2 Experimental results	53
8 Application of Passive Connection to Teleoperation	55
8.1 Delayed Teleoperation and passivity	55
8.2 System Description	56
8.3 Simulations	58
8.3.1 Small delay	58
8.3.2 Delayed Teleoperation	58
8.4 Experiments	63
9 Conclusion and final remarks	67

Chapter 1

Introduction

This thesis is centered on topics that mainly concern two application fields: Haptics and Teleoperation. In this chapter a brief description of the problems that are normally encountered is given, followed by an introduction that glimpse on how these issues will be dealt with.

1.1 Haptic interfaces

Haptic interfaces are devices that provide the user with informations through the kinesthetic sense. Interfaces designed for this purpose are of great interest for a wide range of applications, such as the simulation of car assembly, in which the feasibility of a manual assembly is investigated, [10], or medical training, such as simulation of minimally invasive surgery, [21]. Also Haptic interfaces are used as master robot in teleoperation systems, where the user commands a slave robot in order to influence a remote environment, [2].

The develop of an application that simulate the kinesthetic rendering of a physical object meets different problem; one of the most discussed is how to render a virtual environment with fidelity and in a stable fashion. The most common haptic interfaces are designed with a clear purpose: to reduce their mechanical friction. This characteristic allows a faithful rendering of the free space i.e. the portion of the virtual environment where no object is met, and, therefore, no force is displayed to the user. At the same time, low mechanical friction reduces the maximum stiffness achievable while interacting a virtual wall (the portion of space occupied by a rigid object) in a stable fashion. Part of the literature written about the control of these devices focuses on this problem; in works as [11] the capability of a particular device to render a virtual wall is analytically studied; this approach has been deeply developed in [1,13,14], providing insight on how the different

characteristics of the haptic interface (static and dynamic mechanical friction, frequency of data acquisition and position quantization) limit the maximum achievable stiffness. In particular, a haptic interface coupled with a virtual wall suffers of two different instabilities: one is determined by position quantization, that brings the system into a limit cycle which amplitude is limited to few times the encoder resolution, and is perceived by the user as a persistent vibration that occurs each time an interaction with a stiff object takes place. The other source of instability is time discretization, and causes the system to oscillate in an unstable fashion. Both behaviors can be read from the point of view of energy generation: in particular, when the power generated (by the analog digital conversion and the digital controller) is more than the energy dissipated by mechanical friction, instabilities occur.

Some works, conversely, are centered on the stabilization of the mechanical system, with an effort on control synthesis, instead of stability analysis. Some of the very first works that follows this guideline are [15, 16], in which the focus is shifted to the system energetic behavior, represented by means of the power network framework (a framework in which both the mechanical devices and the digital models are modeled in such a way that input and output are power conjugated, and is it possible to observe how the power flows through the whole system). The control adopted by Hannaford is based on the observation of energy flow through the network, the detection of the power generated by the digital systems, and dissipation by means of a modulated damper. This technique has been also applied to other control problems, such as telemanipulation [19], and non-collocated feedback control of flexible links [18]. This works heavily uses the passivity property that guarantees the passivity, and thus stability, of the whole system once all the connected subsystems are proved to be passive. While the previously cited works divide the overall system mainly in only two subsystems (the robot and the controller), other authors prefer to decompose the control system more systematically, studying the passivity of analog/digital conversion independently from the passivity of digital control system. In [28, 29, 31, 30] this concept has been adopted, and a number of techniques has been described. In particular an algorithm to passively simulate a system described by its continuous time differential equations expressed in the port Hamiltonian formalism is presented.

Other control architectures that are investigated in the haptic field are structured as multirate environment. The need of introduce a multirate digital system arises from the necessity of divide the high frequency local controller, capable to render high stiffness virtual wall, from a low frequency virtual environment, that is delegated to update the state of the simulated systems, an operation that grows in complexity as the virtual environment is populated with more objects. This problem has been addressed both as analysis of multirate systems by means of the \mathcal{Z} -transform [8, 7], and with methods akin to computer graphics, as in [27], as well as with network representation, [20]. In particular,

the presence of different techniques to represent dynamical systems, each time modeled to fit the particular application, prevented the convergence to a general architecture.

In this thesis many of the mentioned problems will be tackled, following the path traced in [30]; to do so the Port Hamiltonian formalism, that will be briefly introduced in Chapter 2 will be used. Chapter 3 introduces the algorithm that allows to simulate in a passive fashion a continuous system modeled by means of the Port Hamiltonian Formalism, discussion that is carried through Chapter 4, where the occurrences in which the algorithm is unable to correctly simulate the system. This Chapter, after reviewing the solution already proposed in literature, introduces a novel formal way to handle the fault conditions, that also allows to model new systems as the unilateral spring. In Chapter 5 is discussed the passive interconnection of physical systems to virtual environment modeled by means of the Port Hamiltonian formalism, or more generally, by means of a power network; moreover, the interconnection of models simulated at different frequencies is discussed. The theoretical work is then validated by simulation, in Chapter 6, and experimentally, in Chapter 7.

1.2 Teleoperation systems

Haptic interfaces are also used in teleoperation systems, i.e. a system composed by two robots, one of which is a the interfaces handled by the user, and the other is the robot that performs a task in the remote environment, and the necessary communication infrastructure as well as the digital controllers. The particularity of these systems is that the user can sense, by means of the haptic feedback, how the remote robot interacts with its environment. The capability of perceive the kinesthetic feedback, in addition to the sight, helps the user to perform the task on hand.

Usually, Teleoperation systems are affected by a delay in the communication channel that routes the informations between the 2 robots. The delay degrades the performances of the system, and, eventually brings it to instability.

Chapter 8 shows an other application of the Passive Connectors introduced in Chapter 5 after a brief description of teleoperation system.

Chapter 2

Port Hamiltonian Framework

This chapter formally introduces the instruments used throughout this work. The Port Hamiltonian system definition, along with the passivity criteria, allows to model and analyse the stability of *physical systems*.

2.1 Physical systems

Dynamical systems can be defined by a state space \mathcal{X} , an input signal space \mathcal{U} , an output signal space \mathcal{Y} , and a two function: *the state evolution law* (2.1) and the *output map* (2.2):

$$\dot{x} = f(x, u) \tag{2.1}$$

$$y = h(x) \tag{2.2}$$

where $x \in \mathcal{X}$, $u \in \mathcal{U}$, $y \in \mathcal{Y}$ are the state, input, and output spaces. This general model can be used to represent physical systems, that are characterized by an important feature: the concept of *energy*. Moreover, the system states, that describe the system dynamical behavior, are directly related to the energy stored in the system, therefore such state are defined *energy variables*. A physical system has a dynamic behavior if and only if energy exchange takes place either between its energy variables (that represent the different parts of the system that can store energy), or with the external world; thus the system energy E is defined by its *Hamiltonian* function:

$$E = \mathcal{H}(x) \tag{2.3}$$

Since the system must be modeled in such a way it exchanges power with the world, the spaces \mathcal{X} and \mathcal{Y} must be *dual spaces*, whose *duality product* represent power; in the following the needed definition are presented.

Domain	Effort	Flow
Translational Mechanics	Force	Velocity
Rotational Mechanics	Torque	Angular Velocity
Electric	Voltage	Current
Hydraulic	Pressure	Volume Flow
Thermodynamical	Temperature	Entropy Flow

Table 2.1: Effort and Flows of various Physical Domains

Definition 1. (*Dual Space*)

Let \mathcal{V} be a vector space, its dual space \mathcal{V}^* is the set of linear maps from \mathcal{V} to \mathbb{R} :

\mathcal{V}^* is a vector space with the same dimension of \mathcal{V} , and the elements belonging \mathcal{V}^* and \mathcal{V} are said to be dual respect each other.

Definition 2. (*Duality Product*)

Given $v \in \mathcal{V}$ and $v^* \in \mathcal{V}^*$, the duality product is defined as:

$$\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V}^* \rightarrow \mathbb{R} \quad \langle v, v^* \rangle = v^*(v) \quad (2.4)$$

For each physical domain can be individuated a pair of variables, defined as *effort* e and *flow* f , so that:

- $e \in \mathcal{E}, f \in \mathcal{F} = \mathcal{E}^*$,
- the duality product represents power.

Variables characterized by this properties are called *power conjugated*. Table 2.1 reports the definition of effort and flows in the various physical domains. Definitions 1, 2 allows to introduce the concept of *power ports*:

Definition 3. (*Power Port*)

Let $\mathcal{F}, \mathcal{E} = \mathcal{F}^*$ be the flow and effort vector spaces; a power port is defined as $P = \mathcal{E} \times \mathcal{F}$. Given $f \in \mathcal{F}$ and $e \in \mathcal{E}$, the product $\langle e, f \rangle$ is the power traversing the power port.

Power ports are the medium through a physical system exchange power with the external world, and, in particular with other physical systems. The system are interconnected by an energy preserving structure.

Example 2.1. (*Mass and Spring systems*)

Consider a mass spring system as depicted in Fig. 2.1. It is modeled as two systems, and an interconnection structure, Fig. 2.2. Let q be the spring elongation, p be the mass

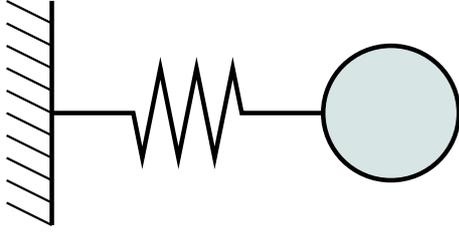


Figure 2.1: Spring and mass system

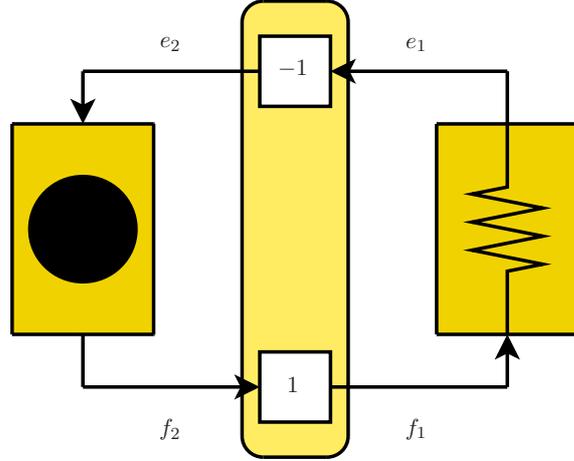


Figure 2.2: Spring and mass model, and the interconnection structure

momentum, k and m the spring stiffness, and mass inertia, respectively. It is well known which is the expression of energy for this system:

$$E = \frac{kq^2}{2} + \frac{p^2}{2m} \quad (2.5)$$

That can split into the two contributes of the spring potential energy E_p , and mass kinetic energy E_k . The spring and mass systems are governed by (2.6) and (2.7), respectively.

$$\begin{cases} \dot{q} = f_1 \\ e_1 = qk \end{cases} \quad (2.6)$$

$$\begin{cases} \dot{p} = e_2 \\ f_2 = p/m \end{cases} \quad (2.7)$$

From (2.6), (2.7) it is possible to compute the amount of energy that is instantly extracted from the mass, \dot{E}_k , and from the spring, \dot{E}_p :

$$\dot{E}_p = \frac{\partial E_p}{\partial q} \dot{q} = kx f_1 = e_1 f_1 = P_1 \quad (2.8)$$

$$\dot{E}_k = \frac{\partial E_k}{\partial p} \dot{p} = \frac{p}{m} e_2 = f_2 e_2 = P_2 \quad (2.9)$$

Since the interconnection structure is power preserving,

$$P_1 + P_2 = 0, \quad (2.10)$$

the mathematical expression of the structure can be written as:

$$\begin{cases} f_1 = f_2 \\ e_2 = -e_1 \end{cases} \quad (2.11)$$

From (2.8), (2.9) and (2.10) follows

$$\dot{E}_p = -\dot{E}_k \quad (2.12)$$

i. e. the energy is transferred between the two systems.

2.2 Explicit Port Hamiltonian Systems

Physical systems can be described with a general form, called *Explicit Port Hamiltonian form*, where the adjective “explicit” means that the causality of the system is determined, as the input-output spaces are chosen. The Port Hamiltonian form is defined as:

Definition 4. (*Explicit Port Hamiltonian Systems*)

An explicit Port Hamiltonian system is a continuous time dynamical system characterized by:

- The state space \mathcal{X} , where the state variables $x \in \mathcal{X}$ are energy variables;
- the input and output spaces \mathcal{U} and $\mathcal{Y} = \mathcal{U}^*$, where the input and output variables $u \in \mathcal{U}$ and $y \in \mathcal{Y}$ are power conjugated;
- the behavior of the system expressed in 2.13.

$$\begin{cases} \dot{x} = [J(x) - R(x)] \frac{\partial \mathcal{H}(x)}{\partial x} + G(x)u \\ y = G^T(x) \frac{\partial \mathcal{H}(x)}{\partial x} \end{cases} \quad (2.13)$$

Where $J(x)$ is a skew symmetric matrix ruling the internal power preserving interconnection, $R(x)$ is a symmetric positive semidefinite matrix representing the system dissipation, $G(x)$ is a matrix that describes how the power injected in and extracted from the system is distributed among the power variables, and $\mathcal{H} : \mathcal{X} \rightarrow \mathbb{R}$ is the lower bounded function that computes the system energy, the Hamiltonian function. For time invariant linear systems the matrices J , R and G are constant and the dependency from the state variable x can be omitted.

For a Port Hamiltonian system there is a general form of the instantaneous power balance:

$$\dot{\mathcal{H}} = y^T u - \frac{\partial^T \mathcal{H}(x)}{\partial x} R(x) \frac{\partial \mathcal{H}(x)}{\partial x} \quad (2.14)$$

This equation, and its strict relation with the energy variables will be the main characteristic that motivate the use of the Port Hamiltonian formalism as framework for the algorithms presented in the next chapters.

2.3 Passivity and Port Hamiltonian Systems

Loosely speaking, a system, whose inputs and outputs constitute power ports, is said to be passive if the energy that can be extracted through its power ports is bounded, with the limit of the initial energy E_0 :

$$-\int_{-\infty}^t y(t)^T u(t) \leq E_0 \quad \forall t \in \mathbb{R} \quad (2.15)$$

In order to formally introduce the passivity concept, the following definition must be given:

Definition 5. (*Passive system*)

A system of the form (2.1,2.2), whose output and input signals $x \in \mathcal{X}$ and $y \in \mathcal{Y} = \mathcal{X}^*$ are power conjugated, it is said to be passive if exists a lower bounded function $V : \mathcal{X} \rightarrow \mathbb{R}$, called storage function, such that, for all $u \in \mathcal{U}$, $x_0 \in \mathcal{X}$, and $t \geq 0$ the following holds:

$$V(x(t)) - V(x_0) \leq \int_0^t w(\tau) d\tau \quad (2.16)$$

where $w(t) = \langle u(t), y(t) \rangle = y^T(t)x(t)$ is the supply rate. Without loss of generality, can be assumed that the lower bound of $V(x)$ is zero, in the origin of state space \mathcal{X} .

An important remark follows from Definition 5: if the system evolves autonomously (with null input), deriving (2.16), the following is obtained:

$$\dot{V}(x(t)) \leq y^T(t)u(t) = 0 \quad \forall t \quad (2.17)$$

Thus, being $V(x)$ lower bounded, the system has at least a stable equilibrium point in the minimum of $V(x)$, in the sense of Lyapunov. It can be useful to distinguish also a passive system where the condition expressed by 2.16 is an equality relationship, or a strict inequality.

Definition 6. (*Lossless system*)

A passive system, with storage function $V(x)$, is said to be lossless if holds:

$$V(x(t)) - V(x_0) = \int_0^t y^T(\tau)x(\tau) d\tau \quad (2.18)$$

for $\forall u \in \mathcal{U}$, $\forall x_0 \in \mathcal{X}$, and $t \geq 0$.

Definition 7. (*Strictly passive system*)

A passive system, with storage function $V(x)$, is said to be strictly passive if holds:

$$V(x(t)) - V(x_0) = \int_0^t y^T(\tau)x(\tau) d\tau - \int_0^t S(x(\tau)) d\tau \quad (2.19)$$

for $\forall u \in \mathcal{U}$, $\forall x_0 \in \mathcal{X}$, and $t \geq 0$, where $S : \mathcal{X} \rightarrow \mathbb{R}^+$ is a positive definite function.

From this definitions it can be concluded that the energy injected in a passive system, by its power ports is either dissipated or stored. Formal demonstration and a broader discussion of the theoretical implant of the Port Hamiltonian system, passivity, and physical systems can be found in [\[28\]](#).

Chapter 3

Discrete Time Port Hamiltonian System

This chapter deals with the problem of discrete time passive simulation of system expressed by means of continuous ordinary differential equations. The proposed solution is the application of an algorithm that simulates a Port Hamiltonian not only using a numerical integration of its differential equations, but also enforcing a consistence between the state variable value and the power flowing through the system power ports.

The issue of physical systems simulation is very wide and presents different facets, depending by the purpose the physical engine is designed for. An important topic in this field of interest is the real time simulation, where the physical model is used for control purpose; in this case the resemblance between the real system and its digital implementation is often sacrificed in advantage of small, fixed, computation time, and numerical stability; for this reason modeling with concentrated parameters is often preferred to distributed parameters. Literature offers a large choice of integration algorithms, that allow to benefit of some characteristics in preference of others; The two main characteristics that are desirable, in order to design a real time environment, are:

Fixed integration step: This property allows to determine the execution time with more certainty than for multi step integration algorithm (i.e. simulation methods that split recursively the time step in order to achieve a bounded relative integration error, but where the system input is constant for the whole time step). This property is desirable but not strictly mandatory, since a maximum computation time can be imposed, [26,25], none less no guarantee can be given on the error bounds. Variable

step integration methods, in which the input must be fed at variable time intervals are not suited for real time simulation.

Explicit Algorithm: This need, that is the more troubling, refers that the discrete time algorithm must calculate the output value using only the information given by the previous output; in the case of fixed step algorithm, an integration algorithm can be expressed in the form of a transfer function applying the \mathcal{Z} -transform. The explicitly of the integration algorithm relates directly to a strictly proper transfer function in the \mathcal{Z} domain, assuming that the original continuous time system is strictly proper.

The implicitly of the algorithm is necessary for two main reason:

- When a system is attached to a physical driver through a digital analog converter, three tasks are normally executed: input reading, output writing, and computation. If the digital controller is not strictly proper, the system must read the data, compute the output, and write the output. this introduces an inevitable delay. If the needed computation time is not negligible a one step delay must be introduced in the system modeling, thus bringing again the digital system to a strictly proper system. Otherwise, if the digital system is strictly proper, the tasks are scheduled as follows: write output, read input, computation of the next time step output; in this case, if the digital analog conversions use a small amount of time, compared to the bandwidth of the mechanical actuator, it is reasonable assume that the state of the mechanical system does not change and the input and output are referred to the same time ($t = KT$ in Fig. 3.1(b)). Fig. 3.1 shows the described tasks scheduling policy for a sample time of length T .
- In order to reduce the design complexity, the system modularity is a sought characteristic; if all the sub-systems are implicit, then algebraic loops are avoided by definition. However, this reason is less important than the preceding, since algebraic loop can be resolved, studying each singular problem, analytically.

The Forward Euler and the (Explicit) Runge-Kutta methods are both explicit integration methods; Unluckily, this methods do not give any assurance on passivity preservation. The following example shows the result of a linear system discretization via the Forward Euler Method:

Example 3.1. (*Forward Euler Mass-Spring system discretization*)

Let consider a mass spring system, as the one depicted in Fig 2.1, modeled by means of the Port Hamiltonian formalism; The behavior of the system is expressed by:

$$\begin{cases} \dot{x} = J \frac{\partial \mathcal{H}(x)}{\partial x} + Gu \\ y = G^T \frac{\partial \mathcal{H}(x)}{\partial x} \end{cases} \quad (3.1)$$

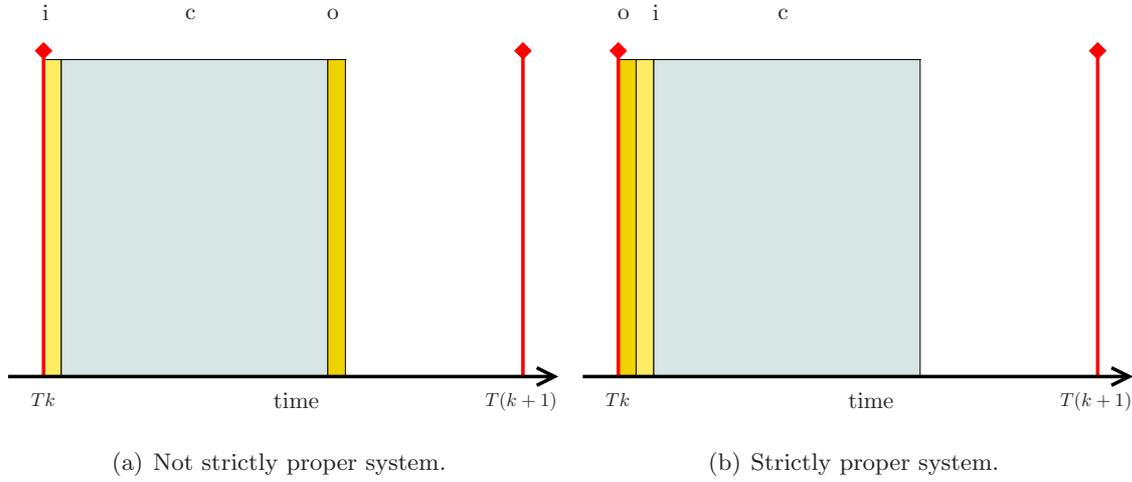


Figure 3.1: Tasks scheduling policy. i: read input, o: write output, c: digital controller computation.

where

$$x = \begin{pmatrix} q \\ p \end{pmatrix} \quad \text{where } q \text{ is the spring elongation, } p \text{ the mass momentum,}$$

$$\mathcal{H}(x) = \frac{q^2 k}{2} + \frac{p^2}{2m} \quad \text{is the system energy,}$$

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{is interconnection matrix,}$$

$$G = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{is input/output map,}$$

u is the force applied to the mass from the environment,
 v is the mass velocity.

It is interesting to study the evolution of the state variables when the system is autonomous; since no power travels through the power port and the system is lossless, the following holds:

$$\mathcal{H}(x(t)) = \mathcal{H}(x_0) \quad \forall t \quad (3.2)$$

Therefore the system state is bound to evolve in a set (continuous blue line in Fig. 3.2).

Applying the discretization method to (3.1) the following discrete system is obtained:

$$\Delta x(kT) = J \frac{\partial \mathcal{H}(x(kT))}{\partial x} + Gu(kT) \quad (3.3)$$

$$x((k+1)T) = x(kT) + \Delta x(kT) \quad (3.4)$$

$$y(kT) = G^T \frac{\partial \mathcal{H}(x(kT))}{\partial x} \quad (3.5)$$

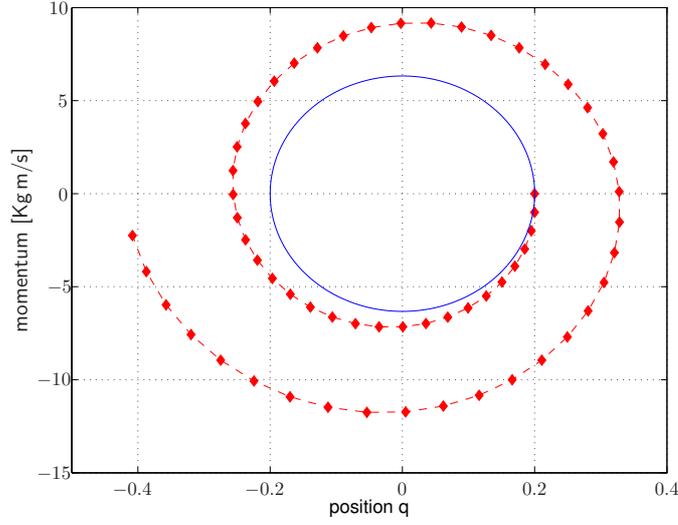


Figure 3.2: Euler integration method: free evolution of a spring-mass system, in the state space; with the following parameters: spring stiffness $k = 5000 \text{ N/m}$, mass inertia $m = 0.2 \text{ Kg}$, initial position $q_0 = 0.2 \text{ m}$, zero initial velocity, sample time $T = 1 \text{ ms}$.

where T is the time step assumed in the discretization process. In Fig. 3.2 the behavior of (3.4-3.5) is shown for a realistic parameter choice for a haptic application Fig. 3.2 presents the state evolution of the continuous time system (blue solid line) and the evolution obtained with an Euler integration (red dashed line), even in very short simulation time, is it possible to observe how the energy of the system increases at each step, and the system is unstable. The same result can be formally deduced in the \mathcal{Z} -transform domain by studying the discrete time system, where the continuous time integrators (mass and spring) are replaced by their discrete counterparts:

$$1/s \longrightarrow \frac{T}{z-1} \quad (3.6)$$

If the external force on the mass u and its velocity y are chosen as input and output respectively, the associated transfer function is:

$$\frac{y(z)}{u(z)} = \frac{T(z-1)}{z^2 - 2z + 1 + \frac{kt^2}{m}} \quad (3.7)$$

with poles in $z_p = 1 \pm \iota T\sqrt{k/m}$, i.e. outside the unitary circle. that belongs to the the vertical line passing through $(1,0)$ in the \mathcal{Z} plane. This result is expected since it is well know that the forward Euler discretization method maps the left semi plane in the \mathcal{S} plane in the semi plane $\{z = \sigma + j\omega \in \mathcal{Z} | \sigma \leq 1\}$ in the \mathcal{Z} plane, and, in particular, the line $\{s = \sigma + j\omega \in \mathcal{S} | \sigma = 0\}$ (that includes the poles of the continuous time system taken as example) maps in $\{z = \sigma + j\omega \in \mathcal{Z} | \sigma = 1\}$ ([9]).

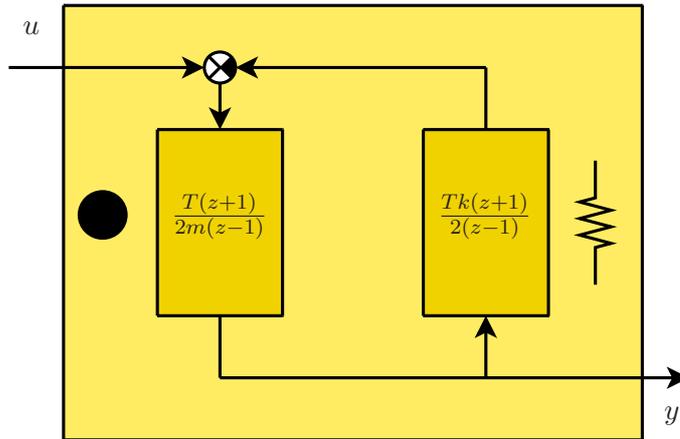


Figure 3.3: Mass Spring system discretized by means of bilinear transformation

A stable discrete system can be obtained by means of implicit integration method, at least for linear systems; the trapezoidal integration method, that is analogous to the Tustin discretization method, maps stable poles in the \mathcal{S} plane to stable poles in the \mathcal{Z} plane and maintains the system passivity property, [12]; the bilinear (or Tustin) transformation from the \mathcal{S} to the \mathcal{Z} plane is:

$$s \longrightarrow \frac{2(z-1)}{T(z+1)} \quad (3.8)$$

Example 3.2. (*Implicit Integration method*)

let consider the system of Example 3.1, as the interconnection of a mass and a spring system as shown in Fig. 3.3; the complete transfer function, using the same parameter choice as before, is:

$$\frac{u(z)}{y(z)} = \frac{2T}{kT^2 + 4m} \frac{z^2 - 1}{z^2 + 2z \frac{kT^2 - 4m}{kT^2 + 4m} + 1} \quad (3.9)$$

The transfer function poles are:

$$z_p = -\frac{kT^2 + 4m}{kT^2 - 4m} \pm \sqrt{\left(\frac{kT^2 + 4m}{kT^2 - 4m}\right)^2 - 1} \quad (3.10)$$

This poles have unitary modules regardless the value of T , as expected, since the bilinear transform maps the imaginary axis of the \mathcal{S} plane to the unitary circle in the \mathcal{Z} plane. Thus, the resulting discretized system preserves the passivity properties, since a linear system is passive (loss-less) if and only if it is stable (marginally stable). The main problem that prevents the application of this method in haptics and, in general, in any real time application, is its implicit nature: every system discretized by an implicit transformation results in a non strictly proper system, thus not suitable to be connected to a similar system, or to be used as the haptic interface controller. The system of Fig. 3.3 is not

implementable since the spring and mass transfer functions have zero relative grade, so their interconnection produces an algebraic loop.

3.1 A passivity preserving numerical simulation algorithm

The simulation of a dynamical system so that the passivity property is preserved can be obtained by enforcing the power or energy balance equations; let $H(k)$ be the system energy at time $t = kT$, and $\Delta H(x)$ the system energy variation in $t \in [kT, (k+1)T)$, so that

$$H(k+1) = \Delta H(k) + H(k) \quad (3.11)$$

If the system is expressed by means of the Port Hamiltonian formalism, the energy variation in a time step can be obtained by integration of power balance equation (2.14):

$$\Delta H(k) = \int_{kT}^{(k+1)T} y(\tau)u(\tau) - \frac{\partial^T \mathcal{H}(x(\tau))}{\partial x} R \frac{\partial \mathcal{H}(x(\tau))}{\partial x} d\tau \quad (3.12)$$

Assuming

$$\left. \begin{aligned} u(t) &= u(kT) := u(k) \\ y(t) &= y(kT) := y(k) \\ x(t) &= x(kT) := x(k) \end{aligned} \right\} \forall t \in [kT, (k+1)T) \quad (3.13)$$

The energy variation is:

$$\Delta H(k) = T \left(y(k)u(k) - \frac{\partial^T \mathcal{H}(x(k))}{\partial x} R \frac{\partial \mathcal{H}(x(k))}{\partial x} \right) \quad (3.14)$$

Note 1. The energy at time kT is expressed by $H(k)$ and computed by means of (3.14), and it may differ from the value of Hamiltonian function $\mathcal{H}(x)$ that depends only by the state, when fault condition arises, (see Chapter 4 and in particular Note 3).

In order to preserve passivity in the discrete time simulation, the state $x(k+1)$ should be taken in consistency with the value of $H(k+1)$. Let I_{k+1} be the set of eligible states whose associated energy $\mathcal{H}(x)$ is equal to the new system energy $H(k+1)$:

$$I_{k+1} := \left\{ x \in \mathcal{X} \mid \mathcal{H}(x) = H(k+1) \right\} \quad (3.15)$$

Consequently, the state $x(k+1)$ is not updated by means of (3.3), (3.4) but with the following algorithm:

Algorithm 1. (State update)

1. Given the actual state $x(k)$ and input $u(k)$ the new energy level can be computed by means of 3.14, and the output by means of 3.5,

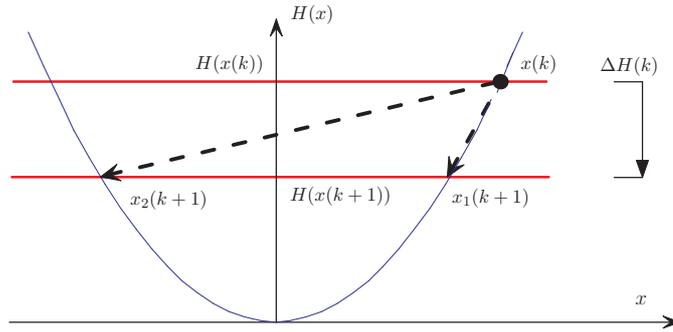


Figure 3.4: Graphical representation of state update for systems with one state variable.

2. The set of eligible states I_{k+1} is found by means of definition 3.15,
3. the new state $x(k+1)$ is chosen in the set I_{k+1} with a proper update strategy.

The update strategy is a key point of Algorithm 1 and will be extensively discussed in the next section.

3.2 Update Strategy

The choice on how the next state is chosen inside the set of eligible state is very important, since the system, which energy consistence is enforced by (3.14), must evolves in way that resembles the continuous time system, and, as the integration step time goes to zero, the discrete and continuous system must have the same state evolution. This condition has been expressed formally in [30] as

$$\dot{x}(kT) = \lim_{T \rightarrow 0} \frac{x(k+1) - x(k)}{T} \quad (3.16)$$

If the simulated system has only a state variable, as mass or spring systems, the set I_{k+1} contains, at most, two element for every energy level. The most natural strategy is to choose the nearest state to the previous, $x(k)$, or consider the sign of the derivative obtained by means of 2.13 as in [28]. Fig. 3.4 graphically illustrates this procedure: at time Tk the system state is $x(k)$; in the subsequent step the system energy drops of ΔH . This energy level individuates the set $I_{k+1} = \{x_1(k+1), x_2(k+1)\}$, and the update strategy has to choose between them; if the nearest point is chosen the next state will be $x(k+1) = x_1(k+1)$.

The solution proposed for the one state variable system cannot be extended as it is to systems with two or more state variable, since the sets I have infinite cardinality, and in the state space \mathcal{X} is not defined the distance between two states, so it not possible to discern the nearest state respect a given state. The solution proposed is based on the

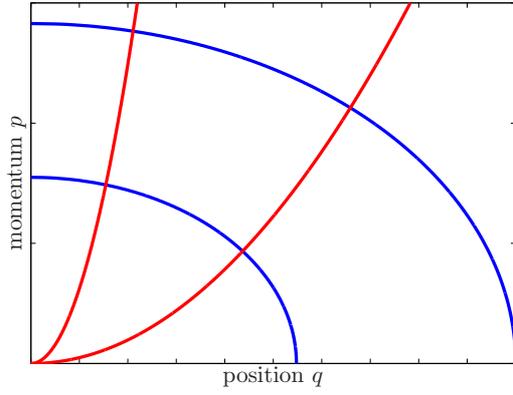


Figure 3.5: Level Curves (blue) and Flux Lines (red) for a mass (damper) system

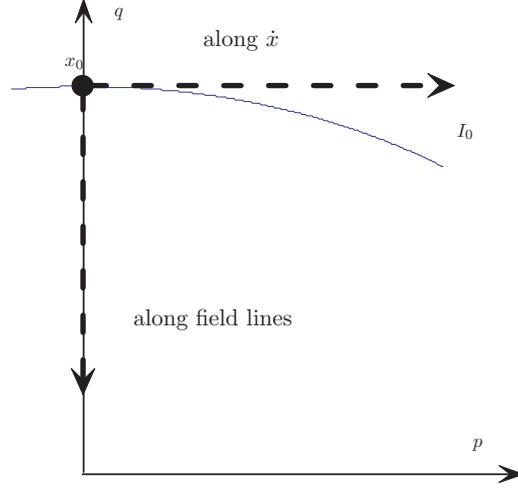


Figure 3.6: Unsuccessful update strategies

movement along two curves, the derivative direction, computed by means of 2.13, and the direction of the co-vector field gradient defined by the energy gradient. The following definition must be introduced:

Definition 8. (*Level Curves*)

Given a Hamiltonian function $\mathcal{H}(x)$ and a set of states I_k defined as in (3.15) the Level Curve associated to a given energy level $H(k)$ is the parametric expression of the set I_k , i.e. the x that solves

$$\mathcal{H}(x) - H(k) = 0 \quad (3.17)$$

Definition 9. (*Field lines*)

Given a Hamiltonian function \mathcal{H} the field lines can be obtained by integration of

$$\frac{dx(s)}{ds} = \frac{\partial \mathcal{H}(x(s))}{\partial x} \quad (3.18)$$

where $s \in \mathbb{R}^+$ is a “time” variable, that parametrizes the curve.

This two families of curves benefit of the following characteristic:

- the two family of lines intersect perpendicularly, and
- each field line intersect every level curves.

Fig. 3.5 shows the plot in the state space of a spring mass system;

In order to explain how the update strategy has been designed two examples, that exploit the state space along the two lines indicated, the field lines, and the derivative direction, are reported, and graphically illustrated in Fig. 3.6.

Example 3.3. (*Unsuccessful update strategies*)

Consider an autonomous mass-spring system with no dissipation and initial condition x_0 ; being the system isolated and lossless, the energy remains constant during the system evolution, as the sets of eligible state: $I_k = I_0, \forall k$.

Consider the derivative direction: it is a line tangent to the level curve, in the starting state x_0 ; consider also the flux line: in the case depicted in Fig. 3.6 it is the positive q axis, what, again, intersects the level curve in x_0 . The update strategies fail because the state is dead locked, and does not evolve along the level curve as expected. Therefore even if the system is energy consistent, it does not bear resemblance with the original physical system.

It follows that the update strategy must enforce a choice of the state so that the simulated system resembles the real one.

Here, a solution that is proposed to this issue consists of a two step algorithm: in first place the state is moved by means of a traditional integration procedure, as a forward Euler integration step; this “dummy” state individuates a flux line, therefore the state can be chosen as the intersection of desired level curve and the chosen flux line. The next example compares the proposed strategy with the previous, unsuccessful, strategies:

Example 3.4. (*Update strategy*)

Let consider the system of Example 3.3; consider that some of the energy is dissipated or absorbed by its power ports, so that the next level curve, denoted by I_1 in Fig. 3.7, is encircled by I_0 . Let consider first the update strategy that moves the state along the field line: The state $x_a(1)$ is found; this state belongs to I_1 and is energy consistent, but a closer look shows that even if the position q changes, the velocity is still null, thus generating a behavior that is not physically consistent. Instead, if the state is moved along the derivative direction (that is the line on which x_0 and $\bar{x}_b(1)$ lie), the convergence is not assured.

The proposed two step procedure moves the state in $\bar{x}_b(1) = \dot{x}T + x_0$ (the “dummy” state) that individuates a flux line. The new state $x_b(1)$ is the intersection between the flux line and the level curve corresponding to the set I_1 of eligible states for this step.

The presented update strategy has two main advantages: since every level curve is intersected by each flux line, the procedure cannot fail to produce the next state. Secondly, the inputs are involved in the computation of state derivative (2.13); this fact allows to distinguish, in a system with two or more power ports, different states for each input values so that the energy variation is the same and thus individuates the same set of eligible states I_k .

Note 2. *The explicit calculation of the intersection between flux lines and level curves can be impossible: in the case of mass spring model used in the previous example the flux line*

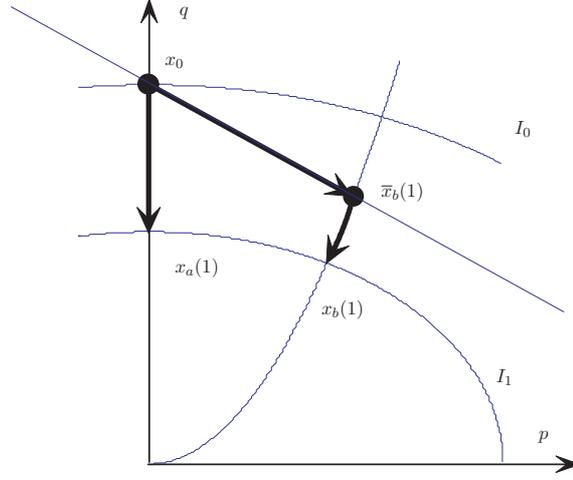


Figure 3.7: Update strategies: along the field line ($x_0 \rightarrow x_a(1)$), along the derivative direction ($x_0 \rightarrow ?$), and the two step procedure ($x_0 \rightarrow \bar{x}_b(1) \rightarrow x_b(1)$).

expression, parametrized by the $s \in \mathbb{R}^+$ variable is:

$$p(s) = p_0 \left(\frac{q(s)}{q_0} \right)^{km} \quad (3.19)$$

where $x = (q_0, p_0)^T$ is a generic state belonging to the flux line (as the dummy state). The level curve corresponding to the set $I_k = \{x \in \mathcal{X} \mid \mathcal{H}(x) = H(k)\}$ is an ellipsoid:

$$kp^2(s) + q(s)^2/m = 2H(k) \quad (3.20)$$

The intersection can be computed explicitly only when $km = 1$ and the ellipsoid degenerates in a circumference, and flux lines in its radius. Two solutions has been adopted:

Bisection method: the intersection is approximated by a bisection method on (3.19), employing the \mathcal{H} as cost function. In this way there is a (bounded) energy error, that can be corrected in later iterations (see Chapter 4).

Approximated flux lines: flux lines are approximated by means a curve that intersect every level curve and whose intersection can be computed explicitly, as the family of lines:

$$p(s) = p_0 \left(\frac{q(s)}{q_0} \right) \quad (3.21)$$

and the intersection can be computed explicitly.

Both methods has been tested and produces good results, and can be easily extended to systems with more than two state variables. Both approximations assume that (3.18) can be explicitly solved; if this assumption fails (3.18) can be resolved numerically until a

state $x(k)$ is sufficiently close to the desired set I_k . In this case a real time integration is bargained with an integration along a curve, that is easier to solve. None less, the convergence is not longer assured.

As explained in the following chapter, the algorithm described is not able to tackle some particular condition, in which an energy error must be allowed.

Chapter 4

Fault Conditions:

A unified treatment in discrete simulation of Port Hamiltonian systems

The algorithm presented in the previous sections cannot be used in some particular conditions, namely when the computed energy is outside of domain of the energy variable, or when the system, due to its strictly proper nature, is locked in the state where the energy is at its minimum; These problems have been described and solved in [31,28]. In this Chapter a generalization of the former methods is proposed, so that all the fault conditions are treated with the same principle. This formal description allows also to easily define new fault conditions; This idea has been applied to formulate the model of an energy preserving unilateral spring.

In order to introduce the topic of fault conditions of passivity preserving Port Hamiltonian system simulation, a small review of the previous work, along with the the proposed solutions, is resumed below.

The fault conditions are divided in two categories:

Dinamic Deadlock: arises when the state is in the energy minimum,

Energy leap: arises when the energy dissipated by the system or extracted through the system power port is more than the store energy.

While the first one is very rare, as it happens when the system is disturbed from its stable condition, the second can happen more frequently, especially in systems with only one state variable. The proposed strategies employs a variable that store the energy error, the energy that is “virtually” generated (or, rarely, dissipated) during the whole simulation of a given system; this variable is defined as *Bookkeeping* variable ($BK(k)$) and depends only by the integration step (k).

4.1 Dinamic Deadlock

Let consider a system expressed as (2.13), with a Hamiltonian function $\mathcal{H}(x)$ that assumes its minimum value of zero in the origin of the state space ($\mathcal{H}(0) = 0$). If the current state $x(k)$ is null, the previous state $x(k - 1)$ is null, and the input $u(k)$ is not null at the time step k , the system is expected to store energy in one or more of its state variables, resulting in a dynamic evolution. In discrete time, if the system is strictly proper, the output will be different from zero with (at least) one sample delay. With Algorithm 1 the system falls in a *dynamic deadlock*: the energy variation $\Delta H(k)$, computed with (3.14), results null, being:

$$x(k) = 0 \Rightarrow \frac{\partial \mathcal{H}(x(k))}{\partial x} = 0 \quad (4.1)$$

$$x(k - 1) = 0 \Rightarrow y(k) = 0 \quad (4.2)$$

$$\Rightarrow \Delta H(k) = T \left(y(k)u(k) - \frac{\partial^T \mathcal{H}(x(k))}{\partial x} R \frac{\partial \mathcal{H}(x(k))}{\partial x} \right) = 0 \quad (4.3)$$

That, in turn, implies that also the next state will be $x(k + 1) = 0$, virtually locking the system dynamic evolution in the origin of the state space.

This behavior can be solved with a normal integration step: thus, for this step only, a normal integration step is used, as the Forward Euler expressed by (3.3-3.5), in order to compute $x(k + 1)$. This operation is not passivity preserving, but injects an amount of energy equal to $\mathcal{H}(x(k + 1))$. This energy can be bookkept and dissipated in later algorithm iterations.

4.2 Energy Leap

This fault condition describes the critical condition in which the energy variation $\Delta H(k)$ computed with (3.14) will bring the energy stored in the system below its energy minimum H_{min} :

$$H(k + 1) = H(k) + \Delta H(k) < H_{min} \quad (4.4)$$

In this case the next state cannot be computed since the set of eligible state is empty, $I_{k+1} = \emptyset$. The solution proposed in [28], called *Energy Leap*, consists in choosing a “symmetrical” state as the next state $x(k + 1)$, belonging to the same set I_k of the actual state $x(k)$; if the energy minimum lies in origin of the state space \mathcal{X} :

$$x(k + 1) = -x(k) \quad (4.5)$$

This strategy is proved to work for system with one state variable; in the next Example this energy leap strategy will be applied to a system such as these.

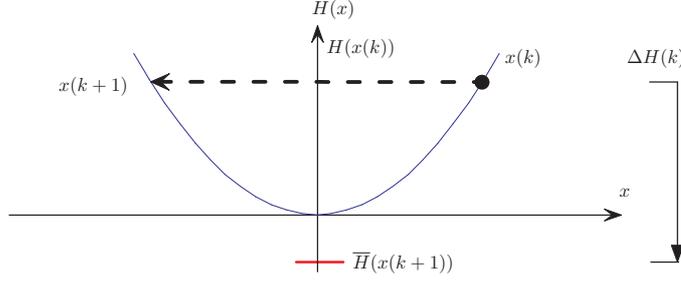


Figure 4.1: Graphical representation of energy leap procedure in one variable systems.

Example 4.1. (*Simmetrical Energy Leap*)

Let consider a system with a state variable, such as a spring. At the step k the condition expressed by (4.4) is verified; referring to Fig. 4.1, it is clear that the set of eligible states \bar{I}_{k+1} with energy $\bar{H}(x(k+1))$ is empty. Applying the described produce as next state is chosen $x(k+1) = -x(k)$. This operation breaks the system passivity, and generates an energy equal to $\Delta H(k)$.

The Example 4.1 ends with an energy generation: this energy must be recorded in the Bookkeeping variable BK , in order to dissipate it later iterations; supposing that in the integration k no fault condition occurs and $BK \neq 0$. The values of the next energy level $H(k+1)$ and $BK(k+1)$ are:

$$H(k+1) = H(k) + \Delta H(k) - \alpha BK(k) \quad (4.6)$$

$$BK(k+1) = (1 - \alpha)BK(k) \quad (4.7)$$

where $\alpha \in (0, 1]$. This constant influences the “passivity recovery”, the time needed to dissipate the energy error. For a value of α near to 1, the system recovers quickly, but also a strong effect on the system dynamics can be expected.

4.3 Unified treatment of fault conditions

From the previous sections emerges that there are some condition in which Algorithm 1 fails to individuate a set of eligible states in which the update strategy must choose the next state. The proposed solutions to tackle fault conditions invert the algorithm “causality”: When no fault condition is met, the choice of state depends on the energy:

$$H(k+1) \Rightarrow I_{k+1} \Rightarrow x(k+1)$$

Conversely, when a fault condition is met, the state is chosen and both the system energy, and the energy error is computed depending on the choice of the state:

$$x(k+1) \Rightarrow H(k+1) \Rightarrow BK(k+1)$$

In order to unify the faults handling, the (3.11) is modified as follows:

$$H_{new} = \Delta H(k) + H(k) \quad (4.8)$$

where H_{new} is a “dummy” variable. In order to keep track of energy errors, the following must hold at each step:

$$H_{new} - BK(k) = H(k+1) - BK(k+1) \quad (4.9)$$

if no fault condition is met and any energy error is not corrected, then:

$$H(k+1) = H_{new} \quad (4.10)$$

$$BK(k+1) = BK(k) \quad (4.11)$$

in consistency with (3.11).

In case that a fault condition arises, the new state $x(k+1)$ must be computed with other methods, such as a forward Euler integration step, employing (3.3-3.4). Once the state $x(k+1)$ is found the system energy must be updated:

$$H(k+1) = \mathcal{H}(x(k+1)) \quad (4.12)$$

And, from (4.9), $BK(k+1)$ can be computed:

$$BK(k+1) = BK(k) + H(k+1) - H_{new} \quad (4.13)$$

The (4.9) rules also the energy exchanges from the bookkeeping variable BK to the system energy H , that happens when some of energy error accumulated in the simulation is eventually corrected; if the energy released to the system is $-\alpha BK(k)$ the following holds:

$$H(k+1) = H_{new} - \alpha BK(k) \quad (4.14)$$

$$BK(k+1) = BK(k) + H(k+1) - H_{new} = (1 - \alpha)BK(k) \quad (4.15)$$

Moreover (4.14) can be modified in order to modulate with different strategies the passivity recovery; as design rule, the amount of energy in the bookkeeping variable should be (almost) null when a fault occurs. At this point is possible to formally state the complete algorithm for simulating a continuous time Port Hamiltonian system, that follows directly from Algorithm 1.

Algorithm 2. (*State update*)

A system expressed by 2.13 can be simulate in discrete time executing the following steps:

1. Given the actual state $x(k)$ and input $u(k)$, compute the output by means of 3.5, and H_{new} by means of (4.8).

2. If a fault condition is met:

- the next state $x(k+1)$ is found via (3.3-3.4),
- the energy is updated by means of (4.12),
- the bookkeeping is computed with (4.13).

3. if no fault condition is met,

- if it is possible to dissipate some of the energy stored in the BK:
 - the energy for the next level is computed with (4.14),
 - the bookkeeping variable is update by means of (4.15).
- otherwise
 - $H(k+1) = H_{new}$,
 - $BK(k+1) = BK(k)$.

4. the set I_{k+1} is determined and the state $x(k+1)$ is found applying the update strategy.

The formal description underling the (4.9) allows to model particular systems with a small effort, as for the unilateral spring.

Note 3. Referring to Note 1, it is worth to remark that Algorithm 2 forces the consistency between state variable and energy, since, at the end of each algorithm iteration, holds:

$$\mathcal{H}(x(k)) = H(k)$$

Moreover, it cannot be stated the same if $H(k)$ is computed as the integral $\sum \Delta H(k)$. Instead, employing Algorithm 2, the following holds:

$$\sum_{n=0}^{k-1} \Delta H(n) = H(k) - BK(k) \quad (4.16)$$

4.4 Unilateral spring

This systems are quite central in haptic application, since it is normally an unilateral spring that exchanges the interaction forces between the virtual environment and the haptic interface. An unilateral spring is mainly a linear spring with only one side permanently attached to a mass or a fixed position reference, and it exchanges power with the other systems, only when is compressed by an other mass; its main characteristics are:

- it can render forces in only one direction,
 - it can render forces only when compressed, i.e. when the position of the interacting mass (the haptic interface) penetrates the rest position of the spring.
-

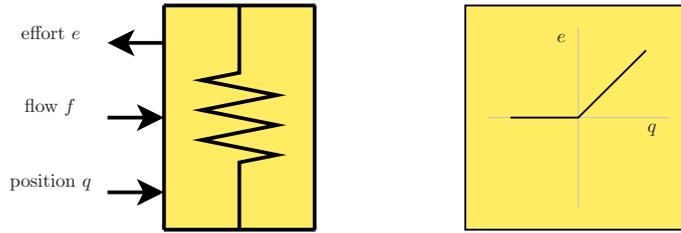


Figure 4.2: Unilateral spring

Seemingly, the above statements express the same condition: if the spring is modeled by means of a the Port Hamiltonian formalism, the input is a flow f ; once the spring is simulated in discrete time, a position drift occurs. For this reason the unilateral spring is feed also with position information in order to determine if an interaction with other (mass) systems takes place. In order to simulate the unilateral spring the following fault conditions must be added:

1. the state space \mathcal{X} is restricted to \mathbb{R}^+ ,
2. if the external position $q(k)$ signals the lack of interaction the next output $e(k+1)$ must be null (*reset condition*).

The previous fault conditions are tackled as follows: if the computed state is negative or the reset condition occurs, the state is reseted to the origin and the generated energy is recorded in the bookkeeping variable.

Note 4. (*Position Drift*)

The reset condition proved to be useful since an experimental setup composed by a virtual environment simulated with Algorithm 2 and a physical interface is affected by a position drift, that becomes not negligible if the experiment is carried over for a long time. There are two source of position drift: one is velocity estimation of the physical system, normally equipped with position sensor, that will be dealt with in Chapter 5.1. Position drift arises every time a system is simulated with Algorithm 2, and, in general, can afflicts any system modeled as a power port network and simulated in real time, since the exchanged signals are force and velocity. A notable exception is the simulation with the Forward Euler, where the state is updated by means of (3.3), i.e. the position is the sum of the input flow sequence; instead in Algorithm 2 the state normally differs from the discrete integral of the input sequence and therefore a position drift is observed. A good method, where applicable, is to consider as position of a mass element attached to a spring the value of the state variable of this spring: consider a mass-spring system, where one end of the spring is grounded, and the input is the external force applied to the mass, as in Fig. 2.1, modeled with two Port Hamiltonian systems, as in Fig. 2.2. As position of the ungrounded spring end, the

state variable q is considered, and the same value is assumed as the mass position. If the position $\bar{q}(k)$ is computed as the sum of the velocity signal:

$$\bar{q}(k) = \sum_{-\infty}^k f_1(n) \quad (4.17)$$

It is likely to observe incongruousness between the force supplied by the spring e_1 and the position $\bar{q}(k)$:

$$\bar{q}(k) \neq e_1(k)/k_s = q(k) \quad (4.18)$$

where k_s is the spring stiffness. This incongruity becomes obvious in some particular condition, i.e. when the spring does not exert a force, but it is not at its rest length. The introduction of the fault conditions described in this section prevent cases as such.

Chapter 5

Passive Interconnection:

Analog digital conversion with velocity estimation and Up/Down sampling.

This chapter deals with two important issues: the connection of the digital controller with the haptic interface equipped with position sensors and the interconnection between two systems simulated at different frequencies, in architecture where the digital controller is modeled by means of the Port Hamiltonian system, or more generally, where the controller is realized as a systems network, in which the exchanged signals are power correlated. This problems have been approached employing the power balance equation, resulting in the design of two dynamical systems that tries to minimize the energy error acting on one of their outputs.

This chapter will describe two elements, or more specifically the idea underling their design, able to connect two systems whose inputs and outputs are power correlated; in system connected by more of one signal couple, it is possible to replicate the interconnection element and treat each energy error independently. The interconnection elements are characterized by 2 power ports, and the observation of the power flowing in and out from these ports is the basis of the control action aimed to bring the energy dissipated or generated in the junction as close as possible to zero. The interconnection systems have some analogy with the *passivity observer / passivity controller* (PO/PC) described in [16, 19, 15], but also bears some innovations; the observation of power flow is employed here, but, also the energy dissipation is considered as a error, allowing a undamped system to behaves as expected; moreover, if the energy is kept near zero, there is no “accumulation” of energy dissipated: in the PO/PC, if the observed system behaves passively for a length of time, and then reverse to an active behaviour, the PC does not act until all the energy generated equals the energy dissipated, thus resulting in an unstable behaviour

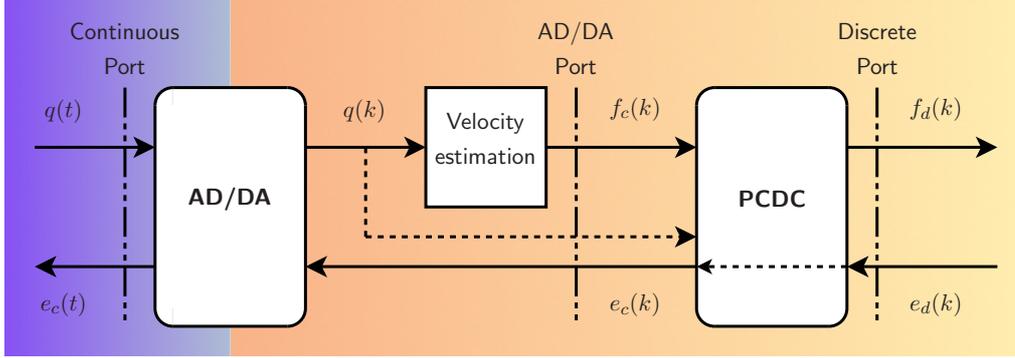


Figure 5.1: The AD/DA and the PCDC components.

until the PC is triggered. This behaviour has been partly eliminated by saturating the “passive energy” that the PO can observe, or with more refined techniques, [5]. Moreover, the PC tries to dissipate all the energy accumulated at once, or, if discrete systems are considered, in one sample time; This strategy can introduce peaks in the velocity or in the force signals: referring to (5.7) and (5.27) a careless use of the PC can generate very high signal peaks, even for small energy correction (i.e. in (5.7) for an almost null e_d and a non null ΔP). The PO/PC designed by Hannaford has been saturated on an experimental basis, i.e. when the PC modifies the forces fed to the haptic interface, the PC control action is limited in order to avoid damages to the mechanisms.

The interconnection systems here described compute a bound on the maximum energy error that can be disposed of. In this way the energy error must be recorded, but the control action on the modified signal can be limited, in particular in the critical condition already cited.

5.1 Passive Continuous-Discrete time Connection

Most of the haptic interfaces are equipped with encoder position sensors; it is well known that the analog digital conversion can break the system passivity and therefore is a source of instability. Moreover, if the digital system is represented as a power network, the digital system expects a velocity as input, and some method to estimate the velocity must be employed, thus forcing the design of a velocity estimator.

Referring to Fig. 5.1 let consider the energy variation $\Delta E_{AD/DA}(k)$ between the ports labeled as *Continuous Port* and *AD/DA Port* in one sample time, $t \in [kT, (k+1)T)$;

$$\Delta E_{AD/DA}(k) = \int_{kT}^{(k+1)T} e_c(\tau) f_{cont}(\tau) d\tau - T e_c(k) f_c(k) \quad (5.1)$$

where T is the sampling period, f_{cont} the velocity of the continuous time system, and e_c the force hold to the the continuous time system, that is equal to the force feed to the

sample and hold e_c from the digital controller:

$$f_a(k) = \int_{kT}^{(k+1)T} f_{cont}(\tau) d\tau = \frac{q(k+1) - q(k)}{T} \quad (5.2)$$

$$e_c(t) = e_c(k) \quad t \in [kT, (k+1)T) \quad (5.3)$$

where $f_a(k)$ is the average velocity of the continuous time system in the time interval $t \in [kT, (k+1)T)$. The 5.1 becomes:

$$\Delta E_{AD/DA}(k) = T e_c(k) \left(\frac{q(k+1) - q(k)}{T} - f_c(k) \right) \quad (5.4)$$

The operation is energy preserving if, for each sample time, $\Delta E_{AD/DA}(k) = 0$:

$$\Rightarrow f_c(k) = \frac{q(k+1) - q(k)}{T} \quad (5.5)$$

The (5.5) cannot be computed in real time, since the next sampled position $q(k+1)$ is unknown at time $t = kT$, as pointed out in [17]. Instead, the discrete derivative is employed to estimate the velocity:

$$f_c(k) = \frac{q(k) - q(k-1)}{T} \quad (5.6)$$

In order to correct this energy the block labeled PCDC (*Passive Continuous-Discrete time Connector*) is inserted between the velocity estimation block and the digital controller, as shown in Fig. 5.1. This block modifies **only** the flow fed as input to the digital controller:

$$f_d(k) = f_c(k) + \frac{\Delta P(k)}{e_d(k)} \quad (5.7)$$

$$e_d(k) = e_c(k) \quad (5.8)$$

where $\Delta P(k)$ is the control action, in terms of power, that the PCDC, imposes to the digital system. The energy variation ΔE between the *Continuous Port* and the *Discrete Port* is computed by means of

$$\Delta E(k) = \int_{kT}^{(k+1)T} e_c(\tau) f_{cont}(\tau) d\tau - T e_d(k) f_d(k) \quad (5.9)$$

$$= T e_d(k) \left(\frac{q(k+1) - q(k)}{T} - f_d(k) \right) \quad (5.10)$$

The energy $E(k)$ generated or dissipated by the whole system represented in Fig. 5.1, at time $t = kT$ is obtained by sum of the single contributions:

$$E(0) = 0 \quad (5.11)$$

$$E(k) = E(k-1) + \Delta E(k-1) \quad (5.12)$$

The main issue that will be discussed in the remainder of this section is the choice of the energy corrected in a sample time $T\Delta P(k)$; This energy must be chosen so that $|E(k)|$ is minimized, but also, it is desirable to avoid an excessive distortion of the flow signal $f_d(k)$ respect to the estimated velocity $f_c(k)$, that acts as a “reference” signal. Note that it is not possible to dissipate the energy generated in the time $t \in [kT, (k+1)T)$ at the beginning of the interval, due to the inability to compute (5.9) at $t = Tk$. Therefore is not possible to obtain a perfect energy balance even if no bound is imposed on the choice of $\Delta P(k)$.

In order to avoid an excessive distortion of the flow signal $f_d(k)$ the maximum value of energy $T\Delta P(k)$ corrected in a sample time must be limited, therefore the following bound is enforced:

$$|f_d(k) - f_d(k-1)| \leq |f_c(k) - f_c(k-1)| \quad (5.13)$$

that, with the (5.7) leads to:

$$\left| f_c(k) + \frac{\Delta P(k)}{e_d(k)} - f_d(k-1) \right| \leq |f_c(k) - f_c(k-1)| \quad (5.14)$$

The *Inverse Triangular Inequality* states that:

$$||a| - |b|| \leq |a + b| \quad (5.15)$$

The left hand side of (5.14) is minimized employing the (5.15):

$$\left| f_c(k) + \frac{\Delta P(k)}{e_d(k)} - f_d(k-1) \right| \geq \left| \left| \frac{\Delta P(k)}{e_d(k)} \right| - |f_c(k) - f_d(k-1)| \right| \quad (5.16)$$

from (5.14) and (5.16), it can be obtained that:

$$\left| \left| \frac{\Delta P(k)}{e_d(k)} \right| - |f_c(k) - f_d(k-1)| \right| \leq |f_c(k) - f_c(k-1)| \quad (5.17)$$

which can be written equivalently as:

$$\left| \frac{\Delta P(k)}{e_d(k)} \right| \leq |f_c(k) - f_c(k-1)| + |f_c(k) - f_d(k-1)| \quad (5.18)$$

$$\left| \frac{\Delta P(k)}{e_d(k)} \right| \geq |f_c(k) - f_c(k-1)| - |f_c(k) - f_d(k-1)| \quad (5.19)$$

The inequality (5.19) is discarded, since only an upper bound to the energy $T\Delta P(k)$ is sought.

The PCDC must also minimize the energy error $E(k)$, therefore the following bounds are enforced:

$$T\Delta P(k) \leq |E(k)| \quad (5.20)$$

$$\text{sign}(\Delta P(k)) = \text{sign}(E(k)) \quad (5.21)$$

This two bounds assure that the PCDC, at most, corrects, during the sample time $t \in [kT, (k+1)T)$ all the energy error $E(k)$ known at time $t = kT$.

Note 5. (*Behaviour of PCDC in critical conditions*)

As already mentioned, the most relevant flaw of the PO/PC is its behaviour when the signal that is power correlated to the one modified by the PC goes to zero. In this case, for small energy corrections, the PC imposes big variations; a closer look to the bound expressed by the (5.18) shows that the PCDC is not affected by this drawback, since, as $e_d(k)$ goes to zero, also $\Delta P(k)$ is forced to zero. In order to avoid zero division in (5.7), $\Delta P(k)/e_d(k)$ is imposed null for values of $e_d(k)$ near to zero.

Note 6. (*2-sample delay in force command*)

The experimental setup, described in Sect. 7.1, and used to validate this procedure is not symmetric, in the sense that a two sample delay affects the communication channel of the signal of the force displayed by the haptic interface. Then, the exchanged energy in a sample time at the two ports must be calculated as follows:

$$\Delta E_c(k) = (q(k+1) - q(k))e_d(k-2) \quad (5.22)$$

$$\Delta E_d(k) = T f_d(k)e_d(k) \quad (5.23)$$

$$E(k) = E(k-1) + \Delta E_c(k-1) - \Delta E_d(k-1) \quad (5.24)$$

where $\Delta E_c(k)/T$ and $\Delta E_d(k)/T$ are the incoming average powers that flow through the continuous and discrete ports. Relation (5.24) replaces (5.12), while the criteria on the choice of ΔP are the same as before.

The procedure can be easily summarized and implemented in the following algorithm:

Algorithm 3. (*Passive Continuous-Discrete time Connection*)

1. Compute $E(k)$, via relation (5.12) or (5.24);
2. Impose $e_c(k) = e_d(k)$;
3. Compute $f_c(k) = (q(k) - q(k-1))/T$;
4. Compute $\Delta P_0 = e_d(k)(|f_c(k) - f_c(k-1)| + |f_c(k) - f_d(k-1)|)$;
5. Compute $\Delta P(k) = \text{sign}(E(k)) \min(P, E(k)/T)$;
6. Compute $f_d(k) = f_c(k) + \Delta P(k)/e_d(k)$.

As implementation aspect has been imposed that, if $e_d(k) \simeq 0$, then $\Delta P(k)$ is chosen null, in order to avoid zero division.

A drawback of this approach is the position drift caused by the correction operated on velocity by the PCDC, as already pointed out in Note 4. The choice of operate on the velocity is dictated by the fact that it is not possible to modify the force feedback

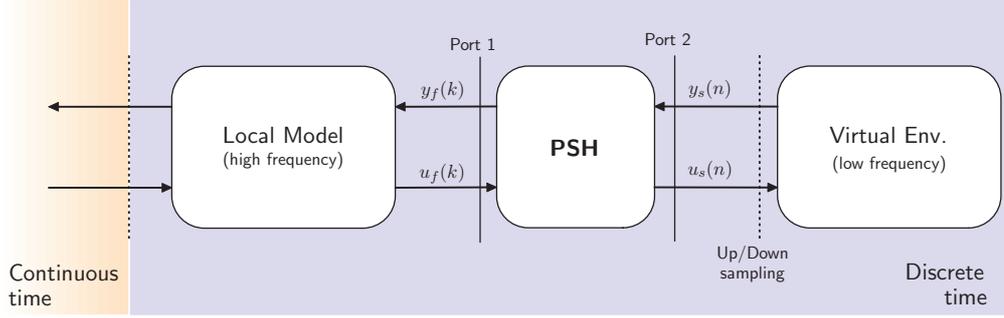


Figure 5.2: Representation of the PSH element.

to the haptic interface because its average velocity is not available and can be estimated only with a delay of a sample period while the force signal is perfectly known. Finally, it is important to note that the reference velocity $f_c(k)$ can be computed differently than in (5.6) depending on the knowledge of a detailed model or the presence of additional sensors.

5.2 Passive up/down sampling Interconnection

In this section is presented a method to passively interconnect two systems running at different frequencies. The key element is a module that acts as an interface between a *fast system*, with sample period T_f , and a *slow system*, with sample period T_s . It is assumed that $T_s/T_f = N$, with $N \in \mathbb{N}$ and that the two systems are synchronized.

Following the same rationale used in Section 5.1 an interconnection element is interposed between the connected system. This element, referred to as the *Passive Sample and Hold* (PSH) in Fig. 5.2, is a dynamical system running at T_f with two inputs, $u_f(k)$ and $y_s(n)$ and two outputs, $u_s(n)$ and $y_f(k)$ in which u_f , y_f and u_s , y_s are power conjugated couples, and u_s , y_s are updated every N steps, since they are connected to the slower system. The discrete time steps are indicated with k and n , with $n = \lfloor k/N \rfloor$, where $\lfloor \cdot \rfloor$ is the integer part operator.

During a period T_s , the energy stored or extracted in the PSH is expressed by:

$$\Delta E(n) = T_s \left(\frac{1}{N} \sum_{k=nN}^{(n+1)N-1} y_f(k)u_f(k) - [u_s(n)y_s(n)] \right) \quad (5.25)$$

As design choice, the outputs of the PSH has been chosen as:

$$y_f(k) = y_s(n) \quad k \in [nN, (n+1)N - 1] \quad (5.26)$$

$$u_s(n) = u_f(nN) + \frac{\Delta P(n)}{y_s(n)} \quad (5.27)$$

As before, $\Delta P(n)$ has to be bounded in such a way the generated or dissipated by the PSH is brought to zero, without excessively distorting the input signal $u_s(n)$. The critical behavior is met when $y_s(n)$ is close to zero. The proposed strategy limits the influence of the power injected by the PSH element on the variation of the signal $u_s(n)$ with respect to the uncorrected one, as follows:

$$|u_s(n) - u_s(n-1)| \leq |u_f(nN) - u_s(n-1)| \quad (5.28)$$

Replacing (5.27) in (5.28) leads to:

$$\left| \frac{\Delta P(n)}{y_s(n)} + u_f(nN) - u_s(n-1) \right| \leq |u_f(nN) - u_s(n-1)| \quad (5.29)$$

The (5.29) can be expanded in:

$$\left| \frac{\Delta P(n)}{y_s(n)} \right| - |u_f(nN) - u_s(n-1)| \leq |u_f(nN) - u_s(n-1)| \quad (5.30)$$

$$\left| \frac{\Delta P(n)}{y_s(n)} \right| - |u_f(nN) - u_s(n-1)| \geq -|u_f(nN) - u_s(n-1)| \quad (5.31)$$

While (5.31) is always satisfied, (5.30) gives the following bound:

$$|\Delta P(n)| \leq 2|u_f(nN) - u_s(n-1)| |y_s(n)| \quad (5.32)$$

As for the PCDC, is desirable that the energy error $E(n)$ decreases, in modulus, to zero; the following bounds are therefore enforced:

$$|\Delta P(n)| T_s \leq |E(n)| \quad (5.33)$$

$$\text{sign}(\Delta P(n)) = \text{sign}(E(n)) \quad (5.34)$$

Where $E(n)$ is obtained by discrete integration of equation (5.25):

$$\begin{aligned} E(n+1) &= E(n) + \Delta E(n) \\ &= T_s y_s(n) \left[\frac{1}{N} \sum_{k=nN}^{(n+1)N-1} u_f(k) - u_f(nN) \right] + E(n) - T_s \Delta P(n) \end{aligned} \quad (5.35)$$

$$E(0) = 0$$

As for the PCDC, the exact amount of energy is exactly known, but at the end of the sample time of length T_s , since the sum indexes of (5.35) span to $k = (n+1)N - 1$. This procedure can be resumed in the following algorithm, where $E(n)/T_s$ has been considered instead of $E(n)$ for sake of convenience.

Algorithm 4. (*Passive sample and Hold*)

1. Impose $y_f(k) = y_s(n)$;
-

2. Compute $\Delta P_0 = 2|u_f(nN) - u_s(n-1)| |y_s(n)|$;
3. Compute $\Delta P(n) = \text{sign}(E(k)) \min(P, E(n)/T_s)$;
4. Compute $u_s(n) = u_f(nN) + \Delta P(n)/y_s(n)$.
5. Compute $E(n+1)/T_s$, via relation (5.35);

As implementation aspect, has been imposed that, if $y_s(n) \simeq 0$, then $\Delta P(k)$ is chosen null, in order to avoid zero division.

As final remark on Algorithm 4, it can be said that $y_f(k)$ and $y_s(n)$ are flows, the PSH block does not introduce any position drift.

Chapter 6

Simulation

This chapter illustrates the results obtained by means of various simulations employing the algorithms described in the Chapters 3, 4 and 5.

The simulation presented in the next sections are implemented with Matlab/Simulink; in particular, the algorithms are embedded in s-function blocks written in the C language. The simulations described are the follows:

- Simulations of a mass spring system implemented with different design choices.
- Simulations of a multi rate system, employing the PSH.
- Simulations of a discrete time system connected to a continuous time system by means of the PCDC.

The simulations employ different implementation choices of the same model, an undamped mass-spring system, and where applicable, the same parameter values of Example 3.1 and Fig. 3.2 are used

6.1 Single rate Port Hamiltonian systems

This section exposes the simulations that validate the Algorithm 2, and compare the different design choices of the same system. The system taken as example is a undamped spring-mass system with a spring stiffness of $k = 5000 \text{ N/m}$ and a mass inertia $m = 0.2 \text{ kg}$, and a simulation step of $T = 1 \text{ ms}$. The system is initialized with a null initial mass velocity and a spring elongation $q_0 = 1 \text{ m}$. The system has been modeled as follows:

- a) with a single Port Hamiltonian system with two state variable, the mass momentum p and the spring elongation q , represent also the mass position;
-

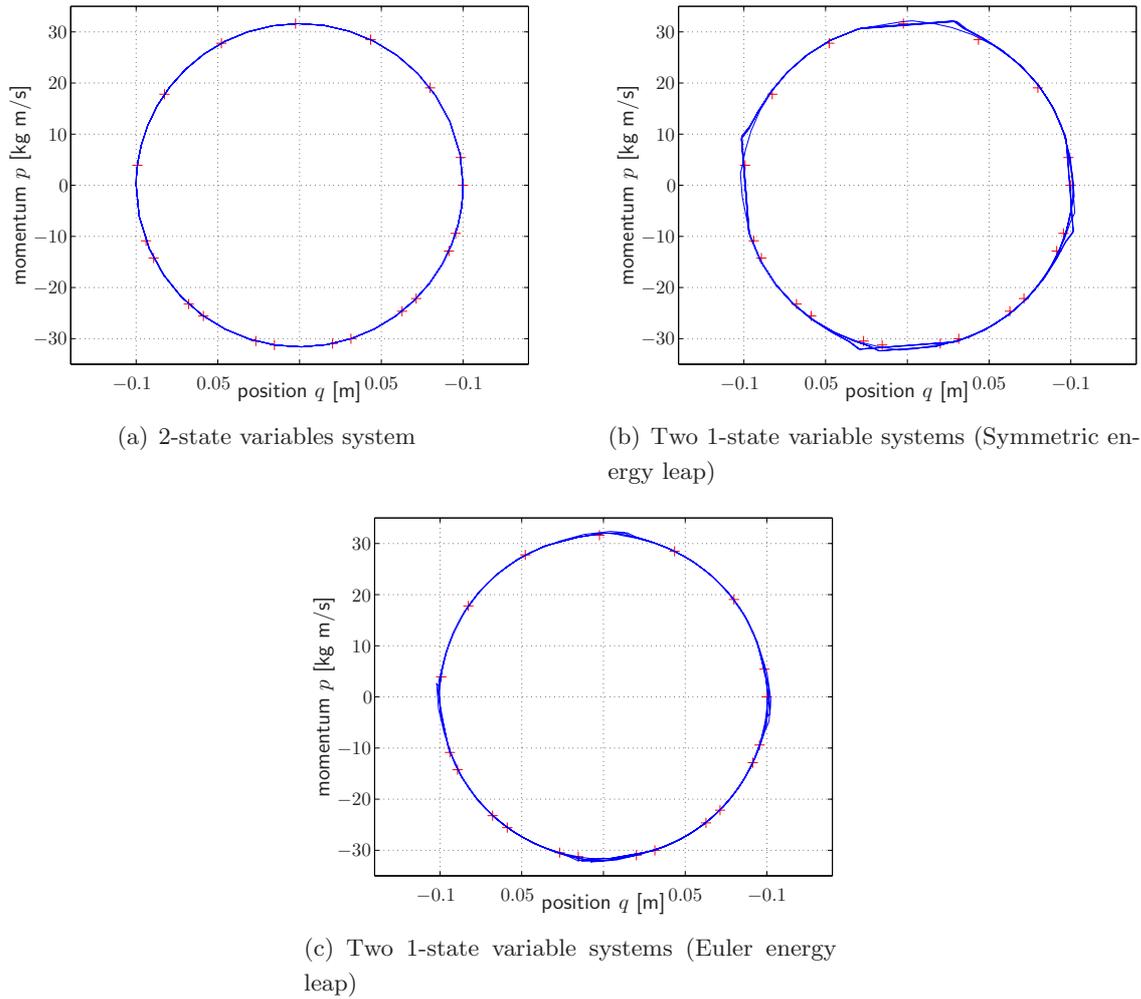


Figure 6.1: Simulation of a single-rate system: momentum Vs. position.

- b) with two Port Hamiltonian systems with a state variable each, q and p , and, as energy leap, has been employed the “symmetrical” strategy;
- c) with two Port Hamiltonian systems with a state variable each, q and p , and, as energy leap, has been employed the forward Euler strategy.

In Fig. 6.1 the system evolution in the state space has been reported for the three simulations, superimposed with the nominal evolution represented with red plus markers. the three models behaves as expected: the model a) remains exactly on the level curves, while the models b) and c) deviate slightly since some energy is generated by energy leaps, as can be observed in Fig. 6.3. None less the passivity recovery is achieved with the used value of $\alpha = 0.3$, and both systems behaves in a stable fashion.

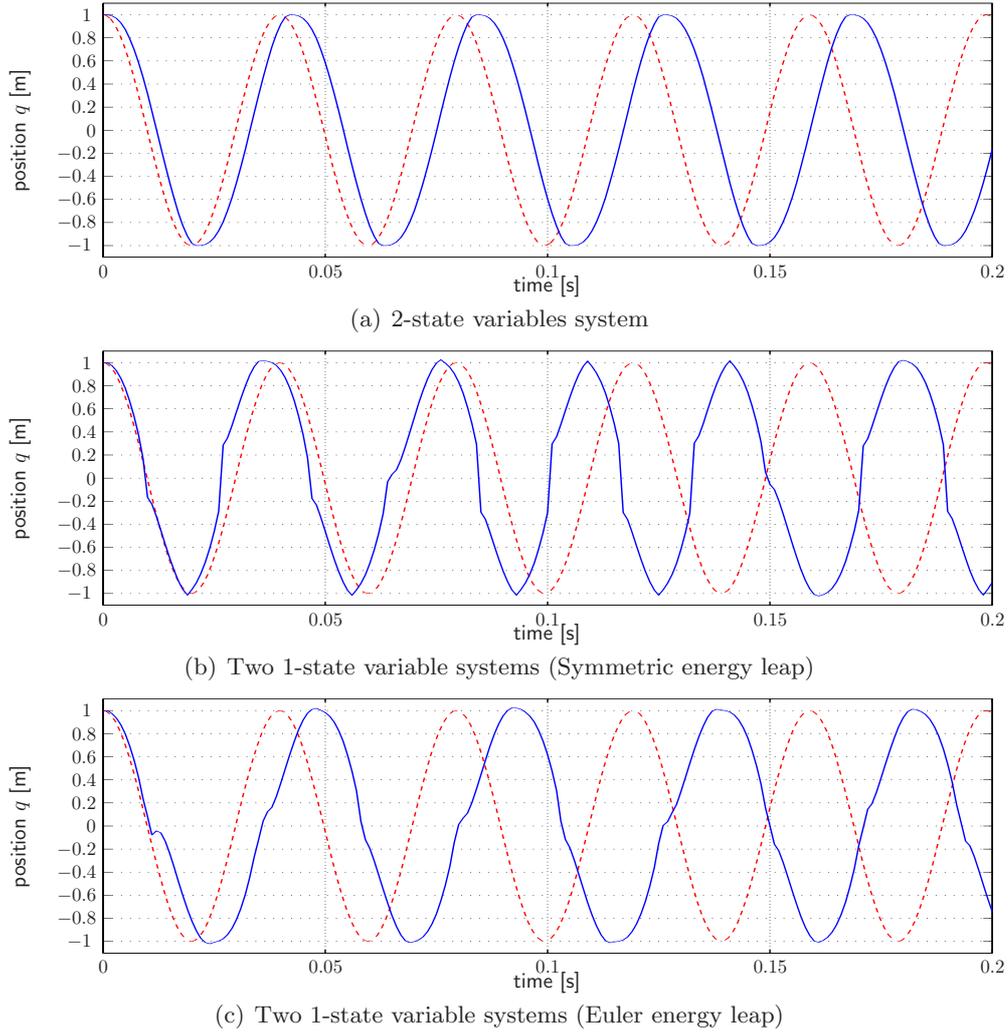
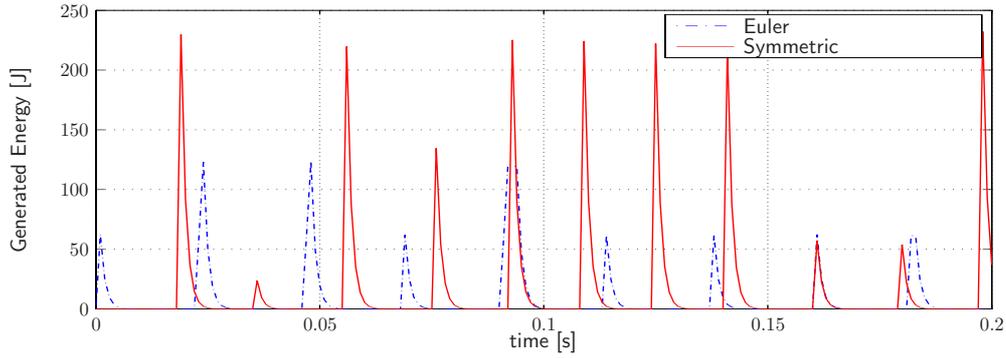
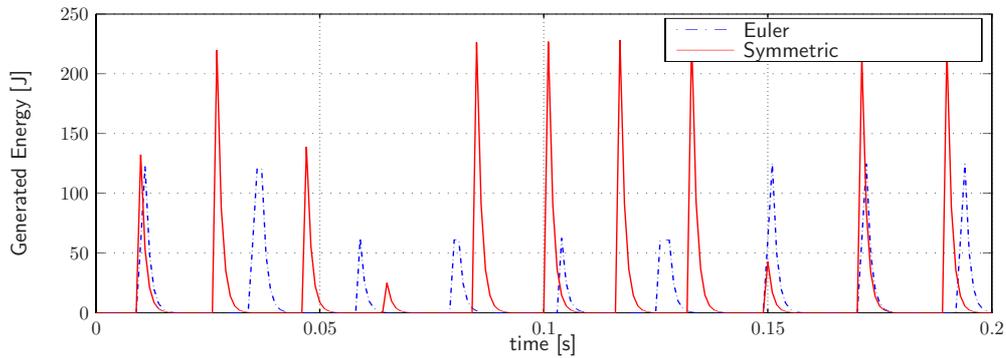


Figure 6.2: Simulation of a single-rate system: position time evolution.

Fig. 6.2 shows the time evolution of the state variable q (the spring elongation and position of the mass) of the three models in solid blue lines, superimposed with the nominal time evolution of continuous time counterpart; in Fig. 6.2(b) and Fig. 6.2(c) can be seen that the models b) and c) generated a fault condition and a energy leap occurs each time one of two systems states crosses the zero; the different strategies used to find the next state in a energy leap modifies the dynamic behavior of the system, distorting the system natural oscillating frequency. In Fig. 6.3 shows the bookkeeping values of the spring and mass that constitute the models b) and c); qualitatively, can be seen that the model mass or spring of the model c), whose bookkeeping variables are plotted with dashed dotted lines, produces less energy when an energy leap occurs. Conversely the system a) shows an high resemblance with its continuous system in its time evolution, as shown in Fig. 6.2(a). The main reason of its goodness is that no energy leap occurs during the simulation, being



(a) Energy generated by the Mass system



(b) Energy generated by the Spring system

Figure 6.3: Simulation of a single-rate system: comparison of the energy generation in the two systems modeled employing the symmetrical or forward Euler energy leap strategy.

the energy of this system a function that depends of both energy variables; the energy flows from a state variable to the other thanks to the interconnection matrix J since the energy is bounded to be constant. It worth noticing that for all the models, and in particular for the models b) and c), the total energy minus the summation of the bookkeeping values is constant, being the system undamped:

$$\sum_i H_i(k) = \sum_i BK_i(k), \quad \forall k, \text{ up to the } i\text{-system} \quad (6.1)$$

as expected from the consideration expressed in Note 3.

6.2 Multirate Port Hamiltonian Systems

6.2.1 Simulation of multi-rate system

In order to qualitatively understand under which conditions the behavior of the Passive Sample and Hold is “acceptable”, several simulations of a multi-rate system have been performed with different choices in the parameters. The benchmark is a simple mass-spring

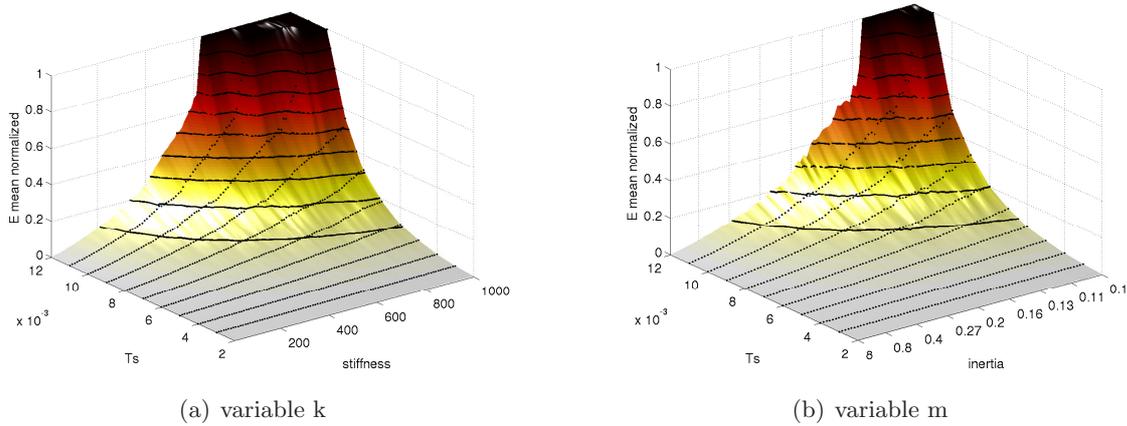


Figure 6.4: Values of E_{norm} in for different choices of the parameters of the system.

system in which the mass is simulated at the slowest sample time T_s , while the spring at the fastest one, T_f . Performances have been compared by introducing the following numerical index:

$$E_{norm} = E_{mean}/E_{sys}$$

Here, E_{mean} is the average value of $E(n)$ (see (5.35)) taken over a sufficiently long time interval so that a steady-state value has been reached, while E_{sys} is the initial energy of the system. As far as the example under study is concerned, since the system is not dissipative, it is clear that E_{sys} is constant and then E_{norm} indicates how much energy is inserted into the system. It's worth noticing that, even if ideally the index should assume a value close to zero, the system remains stable if E_{norm} has a finite value because the PSH always injects a finite quantity of energy. In the Author's opinion, the parameters should be chosen in such a way that E_{norm} is less than 0.2 for the simulated system to perform similarly to the real one.

Two different sets of simulations have been performed. In the first one, the mass has been set equal to $m = 0.2 \text{ kg}$, while the spring stiffness is swept from $k = 10 \text{ N/m}$ to $k = 1000 \text{ N/m}$. On the other hand, in the second set, the spring stiffness is equal to $k = 400 \text{ N/m}$, while the mass is swept from $m = 0.1 \text{ kg}$ to $m = 8 \text{ kg}$. In both cases, the initial velocity is zero, the initial position is $q_0 = 0.2 \text{ m}$, the fast sample time is equal to $T_s = 1 \text{ ms}$, while the slow sample time T_s varies from $T_s = 2 \text{ ms}$ to $T_s = 12 \text{ ms}$. The values assumed by E_{norm} in these different conditions have been reported in Fig. 6.4(a) and 6.4(b). To be able to compare these two simulation sets, in Fig. 6.4(b), the values reported on the *inertia* axis are the inverse of the mass, i.e. $1/m$. This unusual scaling is motivated by the fact that, in continuous time, similar changes in k or $1/m$ lead to a symmetrical effect on the energy flow between the two systems.

In order to show the behavior of the PSH, the results of two different experiments for

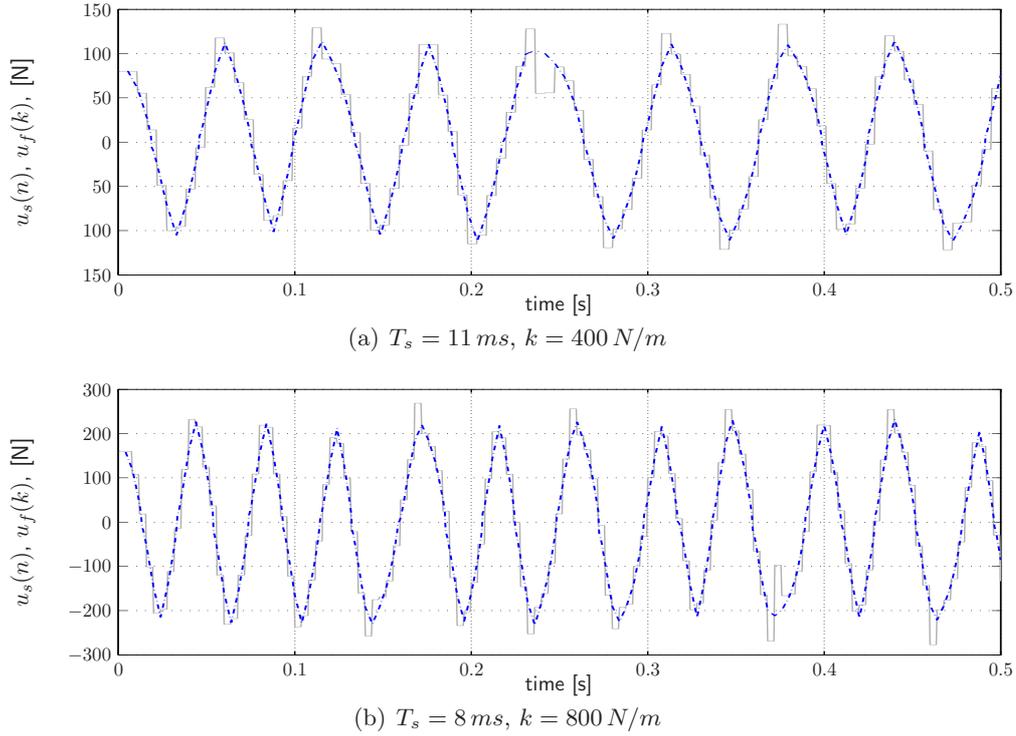
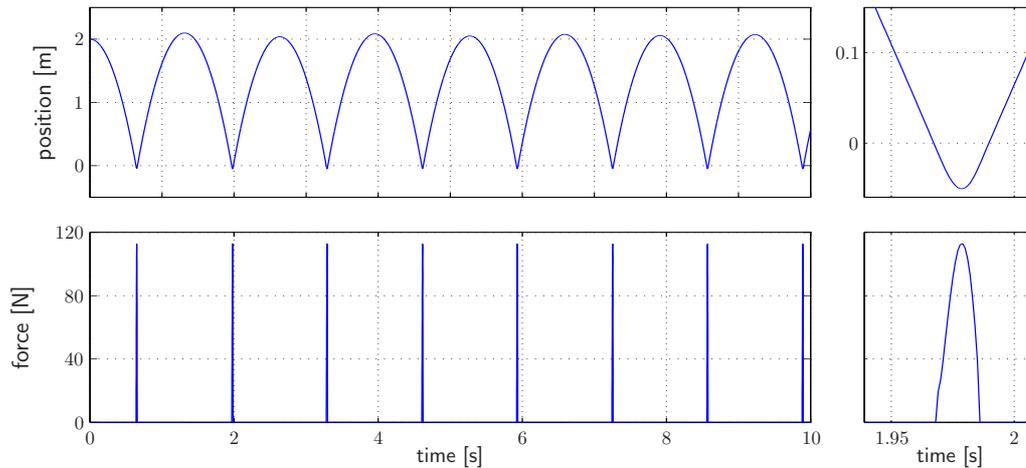


Figure 6.5: Behavior of the multi-rate mass-spring system with different choices of parameters, but with $E_{norm} = 0.2$.

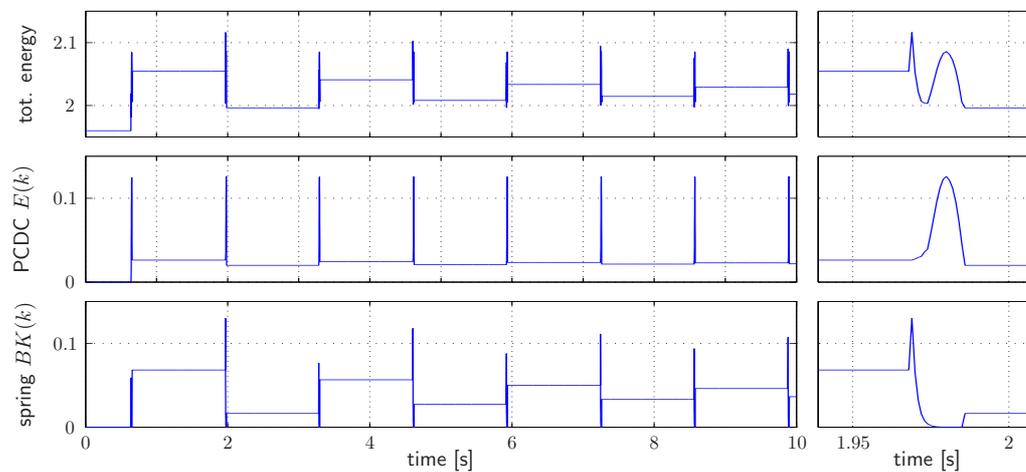
which $E_{norm} = 0.2$ are also illustrated. In Fig. 6.5, the time evolution of u_f (dashed lines) and u_s (solid lines), see (5.27), for different choices of system parameters are reported. In particular, Fig. 6.5(a) shows the system behavior when $m = 0.2 \text{ kg}$, $k = 400 \text{ N/m}$ and $T_s = 11 \text{ ms}$, while Fig. 6.5(b) when $m = 0.2 \text{ kg}$, $k = 800 \text{ N/m}$, $T_s = 8 \text{ ms}$. Note that in both cases the controller output roughly follows the input and no position drift occurs since the u_s and u_f are forces. Moreover, the system is marginally stable, with oscillations greater than the continuous time counterpart because of the energy error.

6.3 Bouncing ball: the PCDC and unilateral spring blocks

In order to validate the unilateral spring modeled with the guidelines depicted in Section 4.4 and the PCDC block described in Section 5.1 the simulation of a “bouncing ball” has been implemented; The ball consists of a mass with an inertia of $m = 0.2 \text{ kg}$, under the effect of the gravity field. The ball falls over a virtual wall of stiffness $k = 3000 \text{ N/m}$ from an height of 1 m , through a non dissipative environment. The mass is treated as a continuous time system, that is interconnected to the discrete time model, which is executed at 1 kHz , by means of the PCDC element. The position of the mass is directly connected to the spring model and acts as reset condition.



(a) Mass position and contact force.



(b) Total system energy (top), PCDC energy (middle), and spring generated energy (bottom).

Figure 6.6: Bouncing ball simulation.

Figures 6.6(a) and 6.6(b) show that the system is stable. More in detail, Fig. 6.6(b) reveals that a small amount of energy is generated during the short impact period, since the mechanisms that correct the energy errors are not able to dissipate the energy generated in such a small time period. The total energy is calculated as the kinetic plus potential energy of the continuous time mass, summed to the potential energy of the spring. The energy generated is the sum of the energy inside the interconnection element and the bookkeeping variable of the spring. The difference between the total energy and the generated energy is constant. This system can be unstable for high values of k ; in particular if the mechanisms that dissipate the power generated by the discrete time simulations and the digital analog conversion are not able neither to bring to zero nor to bound the energy error.

Chapter 7

Experiments

This chapter presents the experimental results obtained by the interaction of a haptic interface with a virtual environment made up with a 2-state Discrete time Port Hamiltonian system (Chapter 3), an Unilateral Spring, (Section 4.4), and the Passive Continuous Discrete time Connection (Section 5.1).

7.1 Description of the experimental setup

The experimental setup hardware consists of an Omega haptic interface and two PCs, the first running Linux Suse, on which the Omega drivers are installed, and the second one equipped with a digital counter board and running QNX. The encoders are directly acquired by the QNX machine which takes care of the real time simulation of the virtual environment at 1 kHz and sends the force signal to the Linux machine through Ethernet.

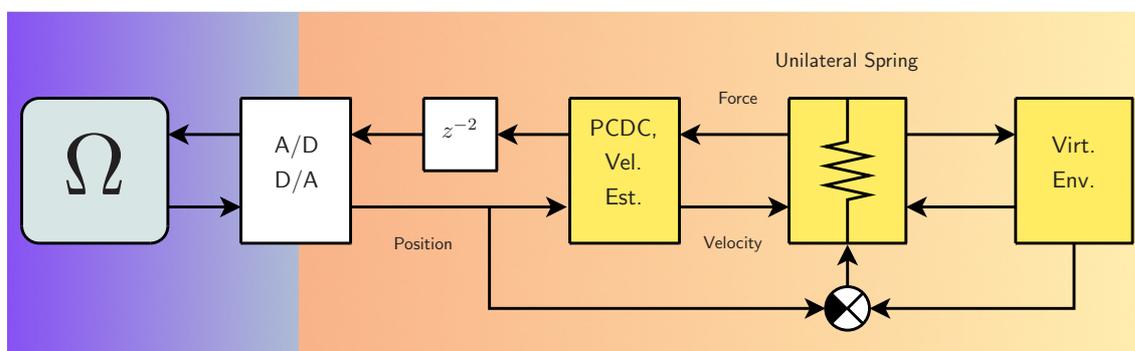
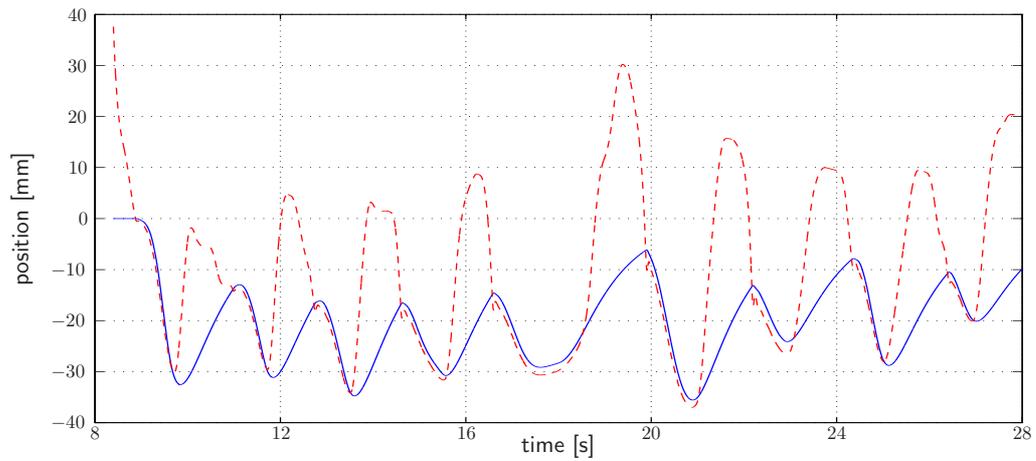
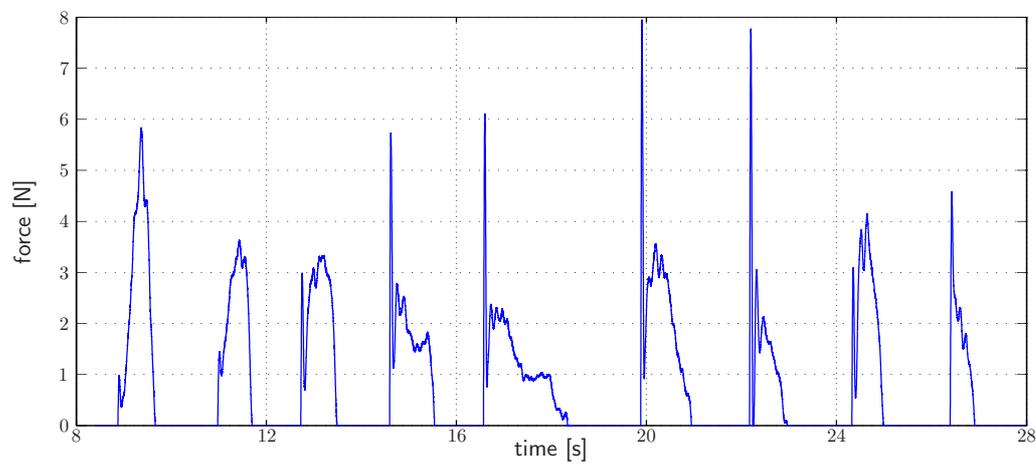


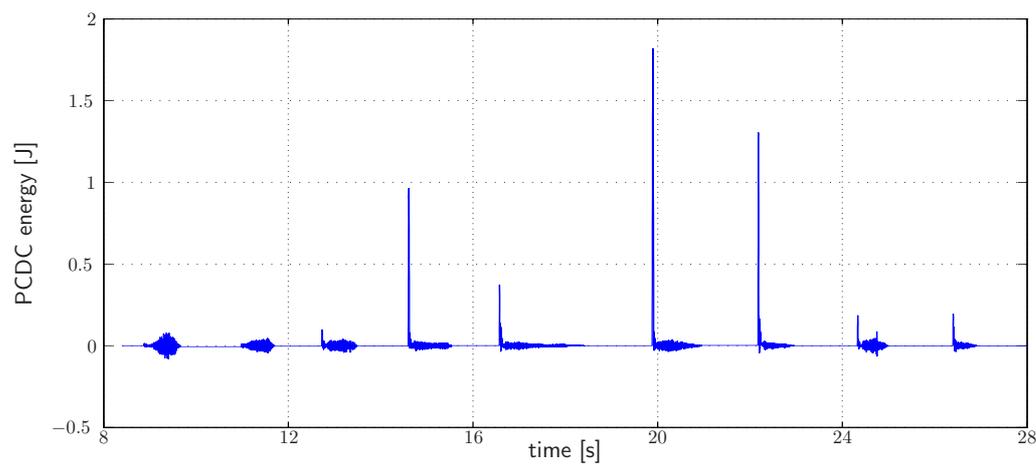
Figure 7.1: Schematic diagram of the experimental setup.



(a) Mass (solid line) and haptic interface (dashed line) position.



(b) Force applied to the haptic interface.



(c) PCDC energy: positive energy is generated, negative dissipated.

Figure 7.2: Experimental results on the Omega haptic interface.

The Linux machine is connected via USB to the haptic interface. A schematic representation of the setup is reported in Fig. 7.1. The force signal delay is estimated in two millisecond. However, due to the time granularity of the real time system (equal to 1 ms), the estimation is not exact, and this fact leads to an error on power exchange computation. In order to cope with the delay associated with the force transmission, the PCDC element is modified as described in Note 6.

The virtual environment represents a soft object, modeled as a spring-mass-damper system ($k = 40 N/m$, $m = 10 kg$, $b = 50 Ns/m$) and implemented as a 2-state variables Port Hamiltonian system, interconnected to an unilateral spring with stiffness $k = 2500 N/m$, that acts as a proxy. The reset condition of the unilateral spring is computed as the difference between the haptic interface position and the position of the mass spring damper system, that is the state of the spring, that has been made accessible has output with this purpose only.

7.2 Experimental results

Experimental results have been reported in Fig. 7.2. More in detail, Fig. 7.2(a) represents the position of both the haptic interface (dashed line) and the mass virtual environment (solid line). As expected, the virtual environment follows the haptic interface when the contact takes place, and recovers its rest position with an exponential law until the next contact. In Fig. 7.2(b) is displayed the interaction force, i.e. the force fed to the haptic interface and the virtual environment, it is possible to see that the virtual environment, due to a rather big inertia, offers a stiff contact when the haptic interface encounters it with a significant velocity. Finally, Fig. 7.2(c) describes the energy stored in the PCDC: the energy error changes only when the haptic interface is in contact with the virtual environment, and the force signal is not null; none less, the energy error is almost null but when the contact takes place.

Chapter 8

Application of Passive Connection to Teleoperation ¹

As mentioned before, The connection systems described in Chapter 5, can be applied to generic systems where the digital controllers are modeled as a power network, and are particularly suited to be used where the flows of energy are taken into account in the study of passivity as principal tool to grant stability. Due to this premises, it appears natural that the PCDC can be easily integrated in the teleoperation schemes that are based on the Passivity Observers /Bilateral Passivity Controllers, as the ones described in [4, 5] or Wave Variables, [2, 23, 24], since it allows to maintain a correct information on the energy inserted in or extracted from the system. In particular, two PCDCs have been integrated in a architecture that is very similar to the one proposed in [6] where the concept of Time Domain Passivity with energy reference is firstly introduced.

8.1 Delayed Teleoperation and passivity

A teleoperator system is composed mainly by two robots, called master and slave robots, and a communication / control architecture. These systems, are used to allow a user to operate remotely, employing the master robot as a haptic interface, and the slave to interact with the remote environment. In this chapter will be briefly reviewed the

¹The contents of this chapter are the results of the work carried over at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), in cooperation with Ing. Jordi Artigas, and will be published in [4].

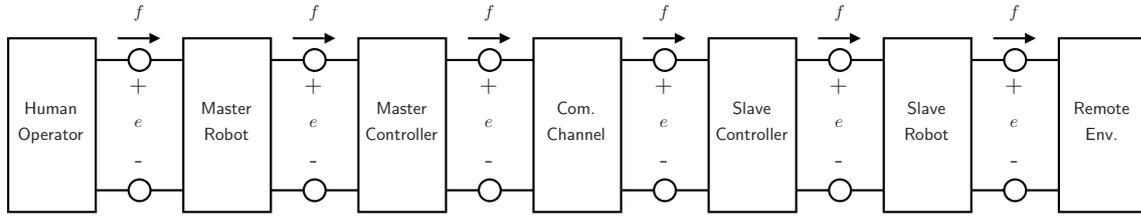


Figure 8.1: General network representation of a teleoperation system

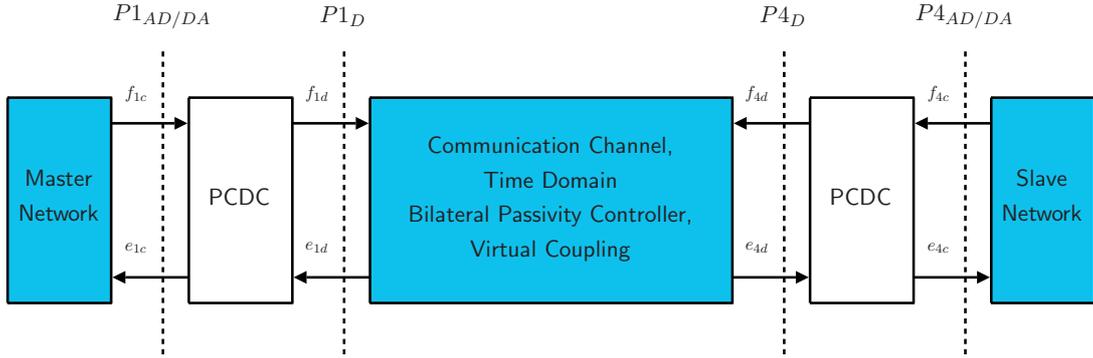
concepts involved in the design of a teleoperation system, in particular the ones related to energy and passivity; the goal of a teleoperation system is to achieve high transparency, while preserving the stability, that are two contrasting aspects, [22]. The concept of ideal transparency is well expressed in [32], where the user should feel the interaction as the two robots were two masses rigidly connected. This idea, if observed by the point of view of energy flows, means that the two robots exchange energy losslessly.

Conversely, when a realistic system is set up, it includes energy leaks, i.e. elements that modify the power flows in the system network, respect to its ideal counterpart. Moreover, if energy leaks have an active nature, they become source of instability for the system.

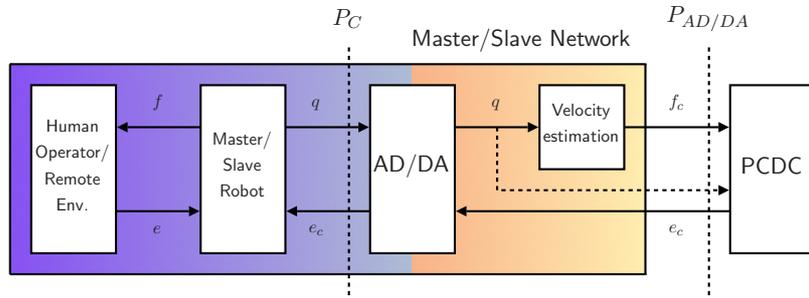
In telemanipulation scenarios, the delay in the communication channel is one of the most common energy leak, and it forces a conservative design of the control architectures, in order to achieve system stability, but lowering the system transparency. This issue has been tackled mainly with two methodology, the Wave Variables [2, 23, 24], or with Time domain Passivity criteria, [4, 5]. In most previous works the energy generated by time discretization is weighted as not relevant respect to the energy generated by time delay; Still, taking care also of this aspect improves the overall stability and performance. This fact are proved by a set of simulations reported in Sec. 8.3, where a particular stress is placed on how the presence of the PCDCs influences the system.

8.2 System Description

In general a telepresence system can be represented with a network representation using the mechanical-electrical analogy where the causality choices are not yet specified, as shown in Fig. 8.1; in such a system both the Human Operator (HO) and the Remote Environment (RE) are considered as part of the network. As for a haptic system, the only subsystems that are allowed to show an active behavior are the HO and the RE, while the rest of the system must be passive. From the general formulation of Fig. 8.1, a particular choice of causality, a velocity-force architecture that does not employ the master controller, is considered. In this case both the causality of the master and slave robot is the natural one, i.e. they provide velocity to the digital system, and the slave controller is a virtual



(a) General view



(b) close-look of the Master and Slave Networks

Figure 8.2: Network representation of the system including the PCDCs

coupling, that represents a damper spring system between the two robots. The considered system is therefore composed by the following elements:

- *Master Network*: Contains the Human Operator, the master device, and the analog/digital conversion plus the velocity estimation from the robot position (AD/DA - Vel. Est.) (see Fig. 8.2(b)).
- *Controllers and Communication Channel*: Is the Communication Channel along with its Bilateral Passivity Controller, and the Virtual Coupling.
- *Slave Network*: Includes the Remote Environment, the slave Robot and its AD/DA - Vel. Est (see Fig. 8.2(b)).
- *PCDC*: The Passive Continuous Discrete time Connector, one at each side.

The following convention with regard to the sign of energy is used:

$$E_P(t) = \int_0^t f(\tau)v(\tau)d\tau + E(0) \geq 0, \quad (8.1)$$

where $E_P(t)$ is the energy of the network, f and e are the port variables denoting velocity entering the block, and force respectively, and $E(0)$ is the energy initially stored in the

network at $t = 0$. This means that if $E_P(t)$ traces a negative slope the energy is flowing from left to right: if $E_{P1_{AD/DA}}$ is considered, a negative slope means that the Master Network is being active; instead, for $E_{P4_{AD/DA}}$, a negative slope implies a passive behavior of the Slave Network.

8.3 Simulations

The master and slave robots are simulated (employing Matlab Simulink) as mass-damper systems with mass $M_r = 1 \text{ kg}$ and viscous friction of $B_r = 0.3 \text{ N s/m}$. The HO is simulated in a closed loop system that generates the force to the master robot in order to follow a sinusoidal reference position with frequency of 0.25 Hz and amplitude of 0.25 m . The simulations are not provided with a Remote Environment, so only free space movements are tested. The control part of the system runs at a sampling time of $T_s = 1 \text{ ms}$. The virtual coupling is obtained by discretization of a continuous time PI controller with gains K_c and B_c ; the plot in Figs. 8.3, 8.4 and 8.5 refer to the energy E_{Px} flowing from left to right through the ports Px as defined in Fig. 8.2.

The goal of these simulations is to show how the presence of the PCDCs, in both in delayed and (almost) undelayed teleoperation scenarios, improves stability and performances.

8.3.1 Small delay

In this simulation a small fixed small delay ($T = 4 \text{ ms}$) is employed allowing to exclude the Bilateral Passivity Controller. The gain K_c of the virtual Coupling has been increased until the system, with both PCDCs deactivated, goes to instability, resulting in a stiffness of roughly 5000 N/m , while B_c has been fixed to 10.4 N s/m .

The same test has been done switching on the PCDCs, obtaining stability with a stiffness up to three times greater. Fig. 8.3 shows the master and slave positions, and energies in various parts of the network; in particular, Fig. 8.3(b) refers to the master and slave positions (q_m, q_s), that are overlapped. Fig 8.3(b) shows that the energy error in the master PCDC, that is several order of magnitude smaller than the energy provided from the master robot at Port E_{P1_D} , Fig. 8.3(c).

8.3.2 Delayed Teleoperation

The simulated system is endowed with a significant delay in the communication channel and also with a Bilateral Passivity Controller that tackles this problem; this point will not be further explained since it evades from the goals of this thesis, please refer to [4,5] for detailed description. The simulations are run with a round trip delay of $T_{RT} = 0.4 \text{ s}$

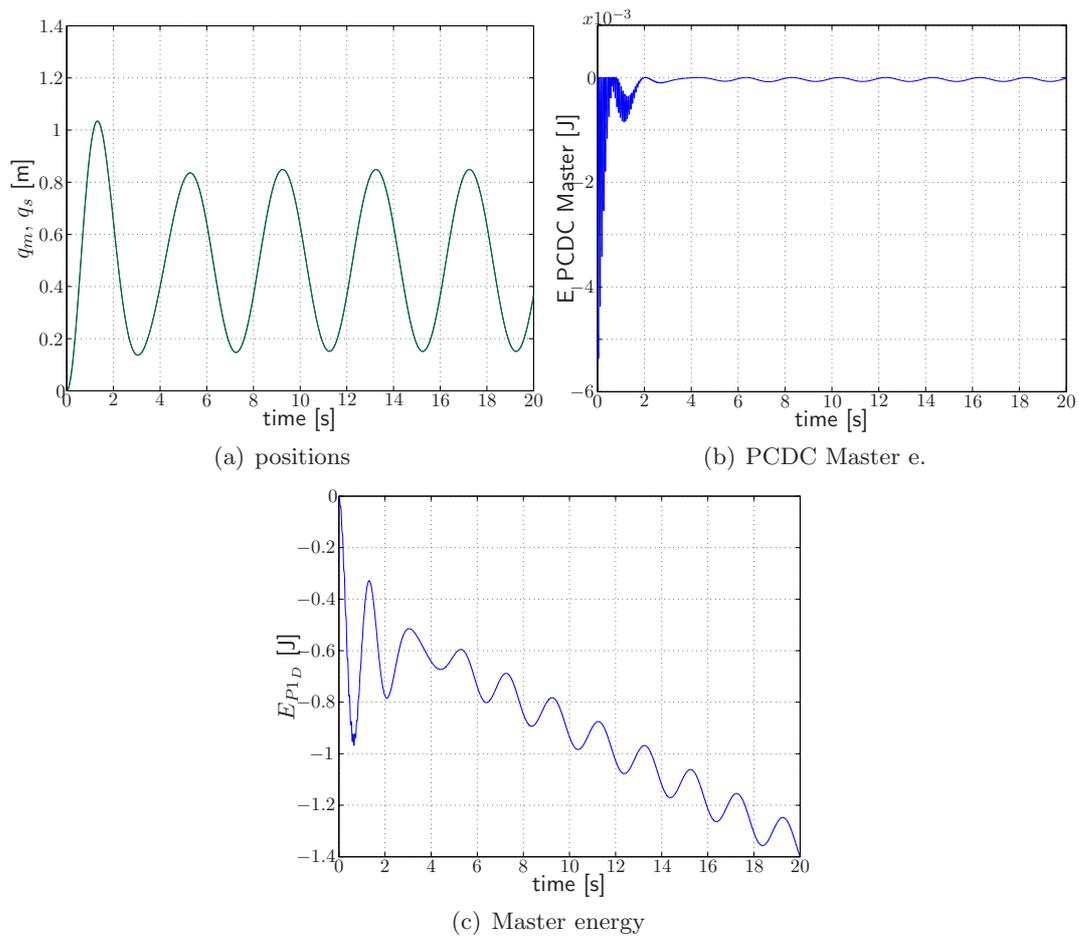


Figure 8.3: Simulation with $T = 0.004$ s, $K_c = 15500$ N/m, $B_c = 10.4$ Ns/m, without the Time Domain Bilateral Passivity Controller

(where $T_{RT} = 2T$), a virtual coupling stiffness of $K_c = 3500 \text{ N/m}$ and damping of $B_c = 10.4 \text{ N s/m}$.

The PCDCs are deactivated in the first simulation (Fig. 8.4) and activated in the latter (Fig. 8.5). The effects of the PCDCs can be evaluated by difference of energy that is injected into the system (Figs. 8.4(b), 8.5(b)). This system works in a stable fashion even if the PCDCs are not inserted, but with the chosen parameters, even if the system results stable, is not able to grant enough performance to purposefully operate a task as can be evaluated by the positions graph, Fig. 8.4(a). Moreover it can be seen that both the Master Network and the Slave Network are resulting in an active behavior; since the slave is moving in free space, the energy generated is due to the velocity estimation; thus the stability is regained thanks to the dissipative mechanism of the Bilateral Passivity Controller and the of the virtual coupling.

Instead, when the PCDCs are activated, the position is significantly less noisy, Fig. 8.5(a), and the flow of energy behaves as expected, from left to right.

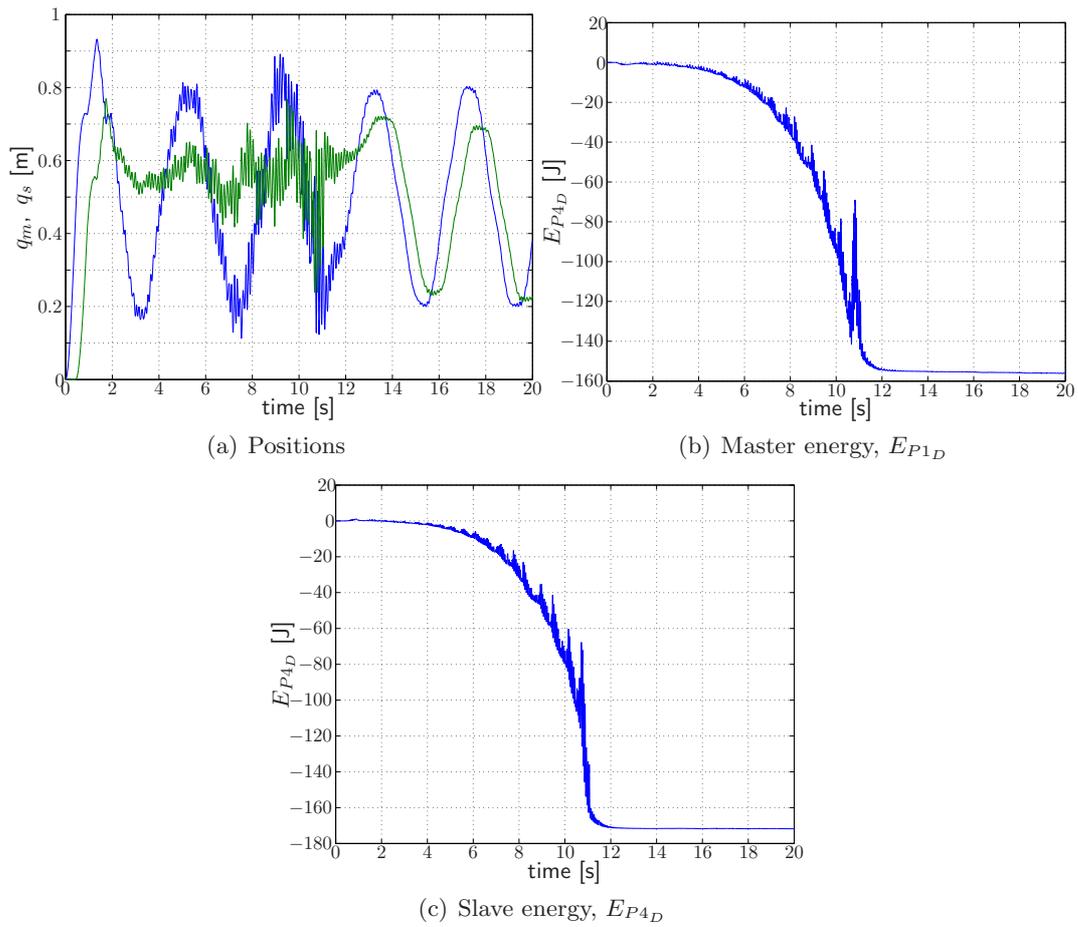


Figure 8.4: Simulation with $T = 0.4$ s, $K_c = 3500$ N/m, $B_c = 10.4$ Ns/m, and PCDCs deactivated

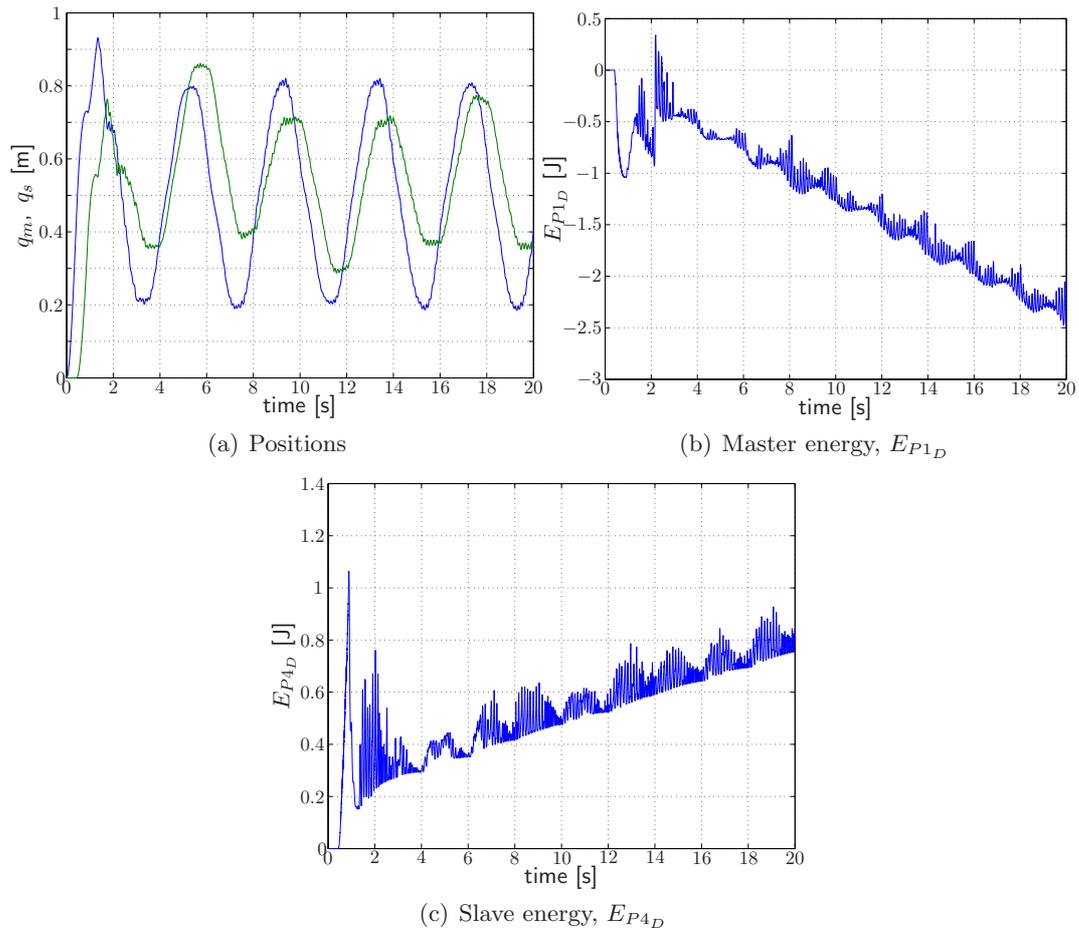
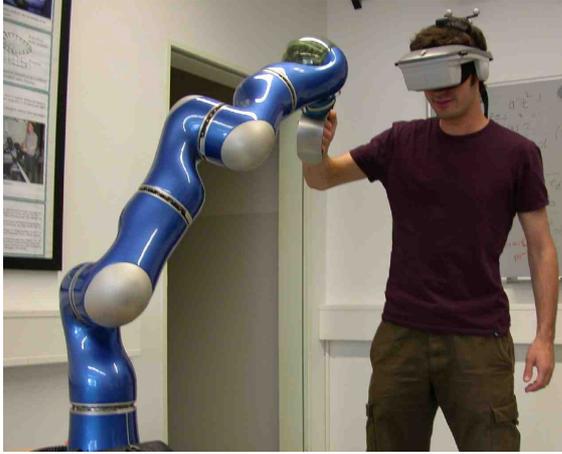
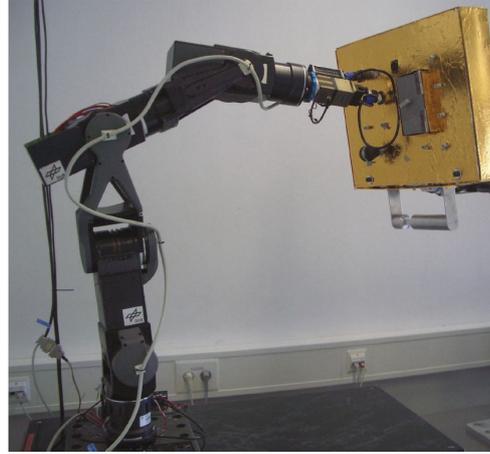


Figure 8.5: Simulation with $T = 0.4$ s, $K_c = 3500$ N/m, $B_c = 10.4$ Ns/m, and PCDCs activated.

8.4 Experiments



(a) LWR III, master robot.

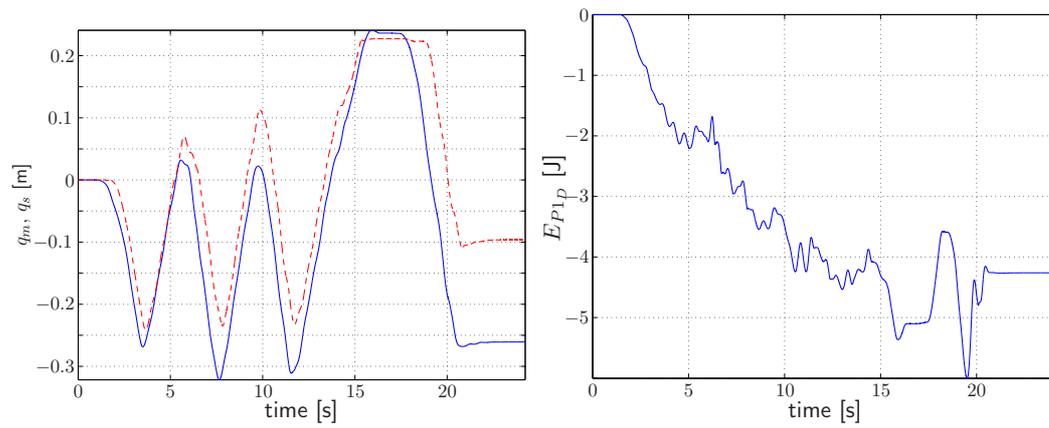


(b) LWR II, slave robot.

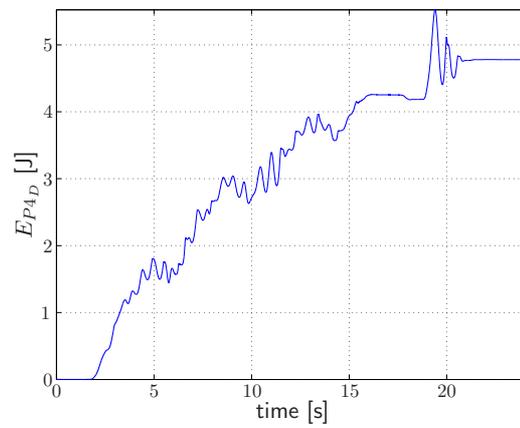
Figure 8.6: Experimental setup.

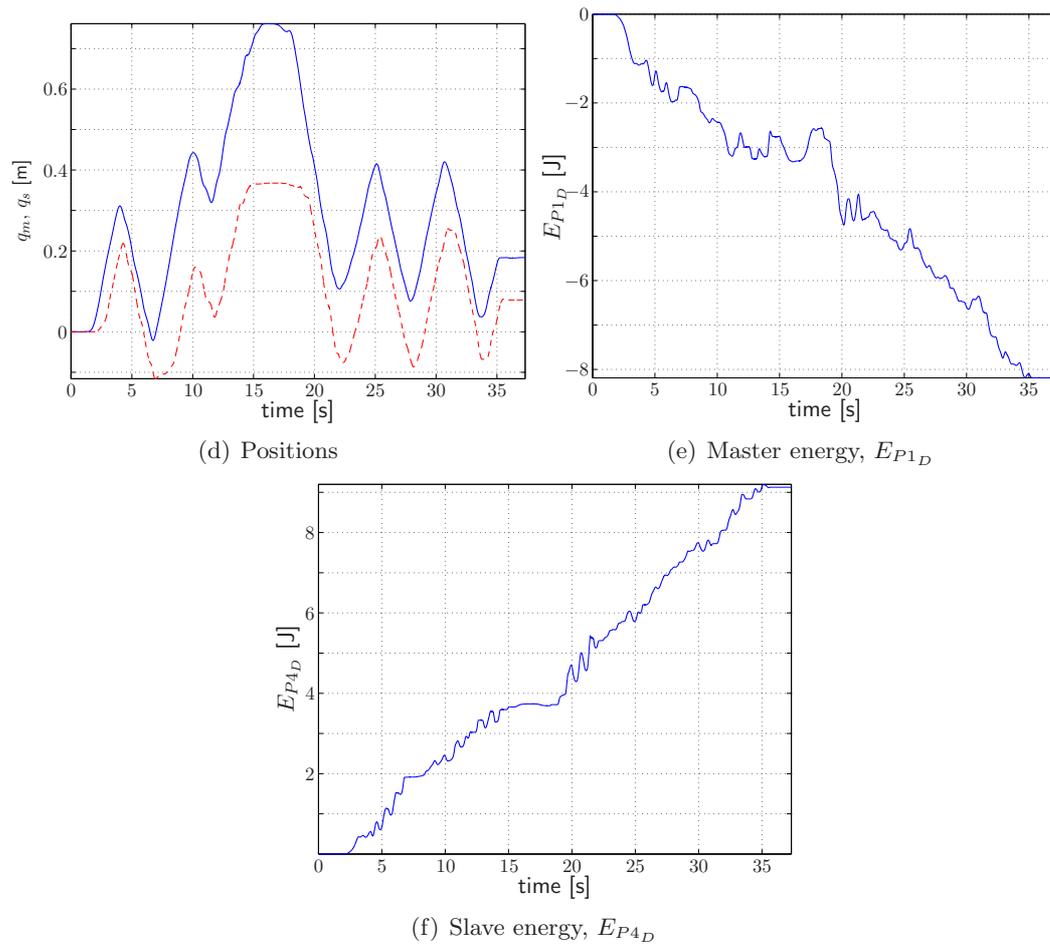
This section shows the results obtained with experiments carried over with a setup consisting of a LWR III and LWR II used, respectively, as a master and slave robots, and shown in Fig. 8.6. These are 6 DoF Light Weight Robots with masses of around $20kg$ developed in the DLR (German Aerospace Center). The LWR III includes a joystick handler as end effector and both robots are real time driven by QNX at a sampling rate of 1 kHz. This setup has been thoughtfully described in [3]. The robots movements have been restricted to a one translational degree of freedom, in order to obtain easy readable data.

For what concerns this thesis, these experiments aim to proof the result obtained by simulation; in particular two experiments, with different delays are presented; the data shown in Fig. 8.4 has been collected using a round trip delay of $T_{RT} = 200ms$, and Fig. 8.7 refers to $T_{RT} = 400ms$. The system has been experimentally proved to be highly unstable due to delay if no Bilateral Passivity Controller is used, with a small delays starting from $T_{RT} = 30ms$. Moreover, the interaction with a Remote Environment is tested; in particular, interaction occurs at $t \approx 16s$ in Fig. 8.4, and $t \approx 15s$ in Fig. 8.7. It can be seen that the experiments follow patterns similar to the simulations.



(a) Positions

(b) Master energy, E_{P1D} (c) Slave energy, E_{P4D}

Figure 8.7: Experimental data with $T_{RT} = 400 \text{ ms}$

Chapter 9

Conclusion and final remarks

This thesis deals with different simulation and implementation issues related to haptics and teleoperation. More precisely, the problem of discrete time simulation of systems described by ODEs is tackled so that energy balance conditions are satisfied at the end of each sample time. A systematic procedure for dealing with book-keeping and energy leap conditions has been illustrated. Furthermore, a novel technique for the power conserving interconnection of digital systems running at different frequencies is illustrated and applied in both haptic and teleoperation scenarios. This methodology is also employed to deal with virtual power generation effects due to analog/digital conversion, that appear when continuous time systems are coupled with discrete time ones. The proposed techniques have been tested by means of simulations and experimental activity employing an Omega haptic interface exploring a simple 1-DoF virtual environment. The results obtained by both simulations and experiments show how the implementation of a virtual environment with the proposed algorithms is feasible and bring some advantages, as the less care needed, with regards to the stability, in the design of the system. Moreover, since the blocks implemented with these algorithms are modular, and each one has a variable that describes its energy error, in case that a system containing one or more of these blocks becomes unstable, it is straightforward to individuate which block has been unable to dissipate the energy, by inspection of the energy errors of each system. The main drawbacks concern the use of velocity, and resulting position drifts; while this phenomena is normally accepted in other applications, like telerobotics, can be disturbing in haptics. None less, the use of a mechanism like a position based reset, as the one used in the described experiment, is a valuable solution to bypass this problem.

The final part of this work shows how a teleoperation scheme which control is based on Time Domain Passivity with Energy Reference, gains benefit by the insertion of two Passive Continuous Discrete time Connectors. Still, future activities will investigate still open issues, such as how to minimize the position drift caused by the simulation algorithm

exposed in Chapters 3 and 4,

An other topic that will be explored is to bring to an unified view the PSH and PCDC, and merge it to the Time Domain Passivity Controllers. in fact these controllers share the same design idea: they must be inserted at one port of a two port system that is affected by energy leaks, and they correct a output signal on the basis of the observed energy at the port that is shared with the inspected system, and an energy reference. The energy reference is normally the energy that is observed to the other system port, but that can be computed, or known, by the controllers only with delay. If these controller are expressed with a homogeneous and unified theory, it will be possible to apply the limit conditions used for PCDC also any Passivity Controller with Energy reference. The further step is the definition of adaptive rules to modulate the maximum energy that the PSH or the PCDC can correct in a sample time, since the energy generation strictly depends on the working conditions and, practically, cannot be estimated before time. As already pointed out, these limits are fixed by common sense, so they can be modified at will; in future activities these limits will be correlated not only with the input/output signals, but also with the energy error, so that these controller will be able to recover system passivity if critical situations are met, allowing more stronger control action if the system risks to become unstable.

List of Figures

2.1	Spring and mass system	11
2.2	Spring and mass model, and the interconnection structure	11
3.1	Tasks scheduling policy. i: read input, o: write output, c: digital controller computation.	17
3.2	Euler integration method: free evolution of a spring-mass system, in the state space; with the following parameters: spring stiffness $k = 5000 N/m$, mass inertia $m = 0.2 Kg$, initial position $q_0 = 0.2 m$, zero initial velocity, sample time $T = 1 ms$	18
3.3	Mass Spring system discretized by means of bilinear transformation	19
3.4	Graphical representation of state update for systems with one state variable.	21
3.5	Level Curves (blue) and Flux Lines (red) for a mass (damper) system	22
3.6	Unsuccessful update strategies	22
3.7	Update strategies: along the field line ($x_0 \rightarrow x_a(1)$), along the derivative direction ($x_0 \rightarrow ?$), and the two step procedure ($x_0 \rightarrow \bar{x}_b(1) \rightarrow x_b(1)$).	24
4.1	Graphical representation of energy leap procedure in one variable systems.	29
4.2	Unilateral spring	32
5.1	The AD/DA and the PCDC components.	36
5.2	Representation of the PSH element.	40
6.1	Simulation of a single-rate system: momentum Vs. position.	44
6.2	Simulation of a single-rate system: position time evolution.	45
6.3	Simulation of a single-rate system: comparison of the energy generation in the two systems modeled employing the symmetrical or forward Euler energy leap strategy.	46
6.4	Values of E_{norm} in for different choices of the parameters of the system.	47
6.5	Behavior of the multi-rate mass-spring system with different choices of parameters, but with $E_{norm} = 0.2$	48

6.6	Bouncing ball simulation.	49
7.1	Schematic diagram of the experimental setup.	51
7.2	Experimental results on the Omega haptic interface.	52
8.1	General network representation of a teleoperation system	56
8.2	Network representation of the system including the PCDCs	57
8.3	Simulation with $T = 0.004 s$, $K_c = 15500 N/m$, $B_c = 10.4 Ns/m$, without the Time Domain Bilateral Passivity Controller	59
8.4	Simulation with $T = 0.4 s$, $K_c = 3500 N/m$, $B_c = 10.4 Ns/m$, and PCDCs deactivated	61
8.5	Simulation with $T = 0.4 s$, $K_c = 3500 N/m$, $B_c = 10.4 Ns/m$, and PCDCs activated.	62
8.6	Experimental setup.	63
8.7	Experimental data with $T_{RT} = 400 ms$	65

Bibliography

- [1] Jake J. Abbot and Allison M. Okamura. Effects of position quantization and sampling rate on virtual-wall passivity. *IEEE Trans. on Robotics*, 21,5:952 – 964, 2005.
 - [2] Robert J. Anderson and Mark W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, May 1989.
 - [3] Jordi Artigas, P. Kremer, Carsten Preusche, and Gerd Hirzinger. Testbed for telepresent on-orbit satellite servicing. In *Proc. of 2nd International Workshop on Human-Centered Robotic Systems (HCRS06)*, Munich, Germany, 2006.
 - [4] Jordi Artigas, Carsten Preusche, Gerd Hirzinger, Gianni Borghesan, and Claudio Melchiorri. Bilateral energy transfer in delayed teleoperation on the time domain. In *Proc. of IEEE International Conference on Robotics and Automation*, 2008. Accepted.
 - [5] Jordi Artigas, Carsten Preusche, and Gerd Hirzinger. Time domain passivity for delayed haptic telepresence with energy reference. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems.*, San Diego, CA, USA, 2007.
 - [6] Jordi Artigas, Jordi Vilanova, Carsten Preusche, and Gerd Hirzinger. Time domain passivity - based telepresence with time delay. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
 - [7] Federico Barbagli, Domenico Prattichizzo, and J. Kenneth Salisbury. Dynamic local models for stable multi-contact haptic interaction with deformable objects. In *Proc. of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 109 – 116, 2003.
 - [8] Federico Barbagli, Domenico Prattichizzo, and J. Kenneth Salisbury. A multirate approach to haptic interaction with deformable objects single and multipoint contacts. *International Journal of Robotics Research*, 24(9):703–715, 2005.
 - [9] Claudio Bonivento, Claudio Melchiorri, and Roberto Zanasi. *Sistemi di controllo digitale*. Esculapio, 1995.
-

-
- [10] Pietro Buttolo, Paul Stewart, and Anne Marsan. A haptic hybrid controller for virtual prototyping of vehicle mechanisms. In *Proc. of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 249–254, 2002.
- [11] J. Edward Colgate and J. Michael Brown. Factors affecting the z-width of a haptic display. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 3205–3210, 1994.
- [12] Javier Garcia de Jalon and Eduardo Bayo. *Kinematic and Dynamic Simulation of Multibody Systems: The Real Time Challenge*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994.
- [13] Nicola Diolaiti, Günter Niemeyer, Federico Barbagli, and J. Kenneth Salisbury. A criterion for the passivity of haptic devices. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 2463–2468, 2005.
- [14] Nicola Diolaiti, Günter Niemeyer, Federico Barbagli, J. Kenneth Salisbury, and Claudio Melchiorri. The effect of quantization and coulomb friction on the stability of haptic rendering. In *Proc. of IEEE World Haptics Conference*, 2005.
- [15] Blake Hannaford and Jee-Hwan Ryu. Time domain passivity control of haptic interfaces. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1863–1869, 2001.
- [16] Blake Hannaford and Jee-Hwan Ryu. Time domain passivity control of haptic interfaces. *IEEE Trans. on Robotics*, 18:1–10, 2002.
- [17] Blake Hannaford, Jee-Hwan Ryu, and Yoon Sang Kim. Sampled- and continuous-time passivity and stability of virtual environments. *IEEE Trans. on Robotics*, 20,4:772–776, 2004.
- [18] Blake Hannaford, Jee-Hwan Ryu, and Dong-Soo Kwon. Control of a flexible manipulator with noncollocated feedback: time-domain passivity approach. *IEEE Trans. on Robotics*, 20(4):776–780, 2004.
- [19] Blake Hannaford, Jee-Hwan Ryu, and Dong-Soo Kwon. Stable teleoperation with time domain passivity control. *IEEE Trans. on Robotics*, 20(2):365–373, 2004.
- [20] Vincent Hayward and Oliver R. Astley. Multirate haptic simulation achieved by coupling finite element meshes through norton equivalents. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 989–994, 1998.
-

-
- [21] Vincent Hayward and Oliver R. Astley. Design constraints for haptic surgery simulation. In *Proc. of IEEE International Conference on Robotics and Automation*, volume 3, pages 2446–2451, 2000.
- [22] Dale A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, 1993.
- [23] Günter Niemeyer. *Using Wave Variables in Time Delayed force Reflecting Teleoperation*. PhD thesis, Massachusetts Institute of Technology, September 1996.
- [24] Günter Niemeyer and Jean-Jacques E. Slotline. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152 – 162, January 1991.
- [25] Gianluca Palli and Claudio Melchiorri. Realtime hardware emulation for rapid prototyping and testing of control systems. In *4th IFAC Symposium on Mechatronic Systems*, 2006.
- [26] Gianluca Palli and Claudio Melchiorri. A realtime simulation environment for rapid prototyping of digital control systems and education. *Automazione e Strumentazione*, 55(2):98–105, 2007.
- [27] June Gyu Park and Günter Niemeyer. Haptic rendering with predictive representation of local geometry. In *Proc. of IEEE Haptics Symposium*, pages 331–338, 2004.
- [28] Cristian Secchi, Stefano Stramigioli, and Cesare Fantuzzi. *Control of interactive robotic interfaces: A port-Hamiltonian approach*, volume 29 of *Springer Tracks in advanced robotics*. Springer Verlag, New York, 2007.
- [29] Stefano Stramigioli, Cristian Secchi, and Cesare Fantuzzi. Delayed virtual environments: a port-hamiltonian approach. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- [30] Stefano Stramigioli, Cristian Secchi, Arjan van der Schaft, and Cesare Fantuzzi. A novel theory for sample data system passivity. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [31] Stefano Stramigioli, Cristian Secchi, Arjan van der Schaft, and Cesare Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Trans. on Robotics*, 21,4:574 – 587, 2005.
- [32] Yasuyoshi Yokokohji and Tsuneo Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling – formulation and experiment. *IEEE Transactions on Robotics and Automation*, 10(5):605–620, October 1994.
-

Thanks

The author kindly thanks his colleagues that helped him during 3 years long study and research period, Luigi Biagiotti, that always found the worst prices-quality trade off of Bologna restaurants, Gianluca Palli, that helped with a lot of linux problems, Raffaella Carloni, that always worried no one get cold, in spite of energy saving and global warming, Riccardo Falconi, alias “Chez Fioron”, that provided dinners, Alessandro Macchelli, coauthor and most up to date source of local news, Andrea Paoli, Matteo Sartini, Luca Gentili, and the other CASY fellows, Roberto Naldi, that organized my gym workout, and Jordi Artigas, that introduced me to telemanipulation, and to Mexican/Cuban places in Munich. The author also thanks his tutor, Prof. Claudio Melchiorri. The last tanks go to Simona, and to the author’s parents, for their unconditioned support.
