2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Simulation Issues in Haptics

Gianni Borghesan, Alessandro Macchelli, and Claudio Melchiorri

Abstract— **In this paper, two problems related to the simulation of virtual environments for haptic systems are considered. The first problem is how to simulate, in discrete time and with low computational effort, dynamic systems in order to preserve their passivity properties. As a matter of fact, simulation of complex systems in real time may lead to undesired effects, like unstable behaviours of the haptic interface, if proper care is not given to the definition of the simulation algorithm. An algorithm is presented here able to maintain the passivity properties of the physical (simulated) system with a reduced computational complexity. The second problem discussed in this paper is the interconnection of algorithms running at different frequencies, i.e. the control algorithm of the haptic interface (running typically at high frequency) and the algorithm simulating the virtual environment (running at lower frequency). A proper software interface, able to connect these two algorithms in an energetic-consistent manner, is presented and discussed. The general framework of both these techniques is the passivity theory and the so-called port-Hamiltonian formalism.**

I. INTRODUCTION

Control problems in the design of haptic interfaces (HI) mostly address two conflicting requirements. The first is stability, that is, in this context, the property that interaction between the HI and a simulated virtual environment (VE) should not generate persisting oscillations. The second is the force rendering range, called *Z-Width* in [2], that determines how "stiff" a simulated object can be.

While in very simple virtual environments the force reflected by the HI is computed directly by the software simulating the VE, in case of complex environments the computations are usually subdivided in two main tasks. The first computes the force feedback for the HI, while the second one computes the dynamics of the VE objects, typically by Local Model methods that consider a simplified model of the VE in the neighborhood of the HI position [8].

The Local Model (LM) must run at high frequency, since performance decreases as the force feedback is computed less frequently. For computational reasons, this lead to design multirate systems, in which the VE is a low-frequency task, due to the heavy computations associated to complex systems dynamics, and the LM is a high-frequency task that assures both stability and performance. A schematic representation of a multirate control architecture is shown in Fig 1.

Stability problems of both single- and multi-rate simulation can be approached with passivity techniques. It is well known that if all the elements of a closed loop (including the HI) are passive, then the whole system is passive, and therefore stable, as well. In haptics, passivity theory has been mainly adopted with two different approaches:

- to find the limits of the VE/LM stiffness, as a function of the HI mechanical characteristics and of the A/D-D/A signals conversion, so that the energy generated by sampling and conversion is dissipated by mechanical friction ([2], [3], [4], and [5]);
- to build an intrinsically passive VE/LM, that does not rely on the properties of the HI ([6], [7], and [11]).

The interaction stability of multirate environments with a HI is studied in [1], using the $\mathcal{Z}$-transform, in order to investigate both the stability interaction of a HI with a VE/LM containing only springs, and the up/down sampling connections between the VE and the LM. This approach cannot be easily extended to more general VE due to the intrinsic limitation of the $\mathcal{Z}$-transform.

While most of the literature does not separate the stability problems of the VE and of the HI with its controller, we investigate separately each element and then their interconnection: between continuous time and discrete time systems, interfaced by the AD/DA in Fig. 1, and between fast and slow rate discrete time algorithms, connected by a Up/Down sampling interface in Fig. 1.

In particular, in this paper, we describe an algorithm able to simulate in discrete time a continuous-time system preserving its passivity properties. Main feature of this algorithm is that the passivity properties of the continuous-time system are maintained also if it is executed at low-frequency.

We also present an element, called the *Passive Sample and Hold*, able to passively connect simulated physical systems running at different frequencies. In particular, this element allows to connect in an energy-consistent manner the HI control loop, running at high frequency, with the VE simulation algorithm, running typically at lower frequencies.

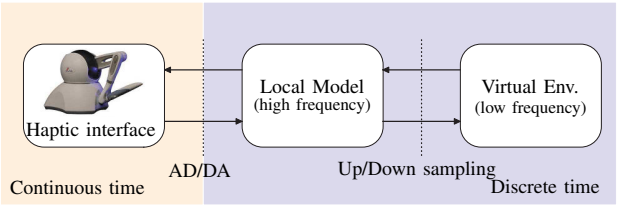

Fig. 1. Typical control architecture of an Haptic Interface.

II. SIMULATION OF PORT HAMILTONIAN SYSTEMS

Since we are interested in obtaining passive (hence stable) elements, we adopt the Port-Hamiltonian formulation to

G. Borghesan, A. Macchelli, C. Melchiorri are with the DEIS,University of Bologna,PO 40136, Italy
email: {gborghesan, amacchelli, cmelchiorri}@deis.unibo.it

describe dynamical systems. The Port-Hamiltonian representation of a physical system is given by:

$$\dot{x} = [J - R]\frac{\partial H(x)}{\partial x} + Gu \quad (1)$$

$$y = G^T \frac{\partial H(x)}{\partial x} \quad (2)$$

where $x \in \mathbb{R}^n$ is the state vector, the *interconnection matrix* J is skew symmetric, $R \geq 0$ is the *dissipation matrix*, G is the *input-output matrix*, and $H(x)$ is the *system energy function*. In a Port-Hamiltonian systems, the product of the input vector u and the output vector y must be power, and the system is passive iff $H(x) \geq H_{min}$, $\forall x$. The power balance of the system is:

$$\dot{H} + \left(\frac{\partial H(x)}{\partial x}\right)^T R \left(\frac{\partial H(x)}{\partial x}\right) + y^T u = 0 \quad (3)$$

In order to simulate complex (nonlinear) systems, a convenient approach is to adopt a numerical integration method. Since a real time environment is generally used for controlling haptic systems, it is reasonable to assume that a fixed step integration method, as the forward Euler, is implemented. In this case, the resulting formulation for Eq. 1-2 is:

$$\Delta x(k) = [J - R]\left.\frac{\partial H(x)}{\partial x}\right|_{x=x(k)} + Gu(k) \quad (4)$$

$$x(k+1) = x(k) + T\Delta x(k) \quad (5)$$

$$y(k) = G^T \left.\frac{\partial H(x)}{\partial x}\right|_{x=x(k)} \quad (6)$$

where $x(k)$ is the discrete-time state and $u(k)$ the input at $t = kT$, being T the simulation step. Obviously, the error introduced by Eq. 4-6 may be relevant if T is not properly chosen.

Example 1. Consider an autonomous spring-mass system with:

- spring constant k ad mass m,
- state $x = (q,\ p)^T$, where q is the spring elongation, and p is the mass momentum,
- energy function $H(x) = kq^2/2 + p^2/(2m)$,
- energy gradient $\partial H(x)/\partial x = (kq, p/m)^T$.

Since the system is not dissipative and the input is null, the state should remain on a set of isoenergetic states, represented with the solid line in Fig. 2. On the other hand, the discrete state computed by the Euler algorithm, Eq. 4-6, diverges (arrows), and the system energy $H(x)$ increases. Thus, a system simulated in this manner is not stable, even though the original one is passive. Note that, for computational reasons, it is preferable to select large values of T, and therefore this type of phenomenon may be easily introduced.

A. Discrete time Port Hamiltonian systems

In order to simulate dynamic systems and preserve their passivity properties, it is desirable that in each integration step the energy of the system is consistent with Eq. 3.

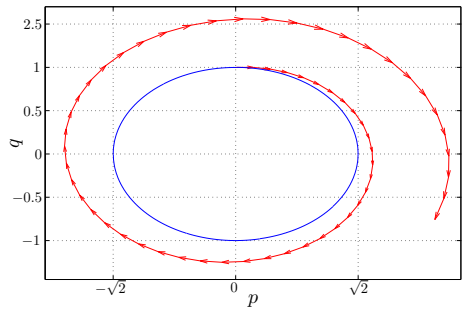

Fig. 2. Euler integration method in state space for $k = 0.5$, $m = 1$, $x_0 = (1, 0)^T$, $T = 0.2$

Therefore, if $\Delta H(k) = H(x(k+1)) - H(x(k))$, the following equation must hold at each sample period:

$$\Delta H(k)/T + \left[\left(\frac{\partial H(x(k))}{\partial x}\right)^T R \left(\frac{\partial H(x(k))}{\partial x}\right) + y(k)^T u(k)\right] = 0 \quad (7)$$

The system energy at $t = (k+1)T$ can be computed as:

$$H(k+1) = H(k) + \Delta H(k) = H(k) - T\left[\left(\frac{\partial H(x(k))}{\partial x}\right)^T R \left(\frac{\partial H(x(k))}{\partial x}\right) + y(k)^T u(k)\right] \quad (8)$$

The state variable $x(k+1)$ should belong to a set I_{k+1} in which the energy function assumes the value $H(x(k+1))$:

$$I_{k+1} = \{x(k+1) \in \mathcal{X} \mid H(x(k+1)) = H(x(k)) + \Delta H(k)\} \quad (9)$$

From these considerations, $x(k+1)$ and $y(k)$ can be computed, instead of using Eq. 4-6, with the following:

Algorithm 1

1) Given the state $x(k)$ and the input $u(k)$, compute $\partial H(x)/\partial x$ at $t = kT$, and then the output $y(k)$ Eq. 6;
2) The energy variation $\Delta H(k)$ is determined with Eq. 7;
3) The new energy level $H(x(k+1))$ and the set of eligible states I_{k+1} are computed by means of Eq. 8;
4) A state is selected in I_{k+1}, with a proper *Update Strategy*.

B. State update strategy

A key point in the procedure for computing the state $x(k+1)$ is the state update strategy, i.e. the selection of a proper state in I_{k+1} (step 4 in Algorithm 1). For this purpose, it is useful to define the *level curves* and the *field lines* in the space state plane:

- The *level curves* are the parametric expression of the sets I_k of all the states with the same energy.
- The *field lines* are the curves obtained by integration of:

$$\frac{dx(s)}{ds} = \frac{\partial H(x)}{\partial x} \quad (10)$$

where s is an independent variable that parameterizes the field line ($x(s) : \mathbb{R} \rightarrow \mathbb{R}^n$). The field lines are

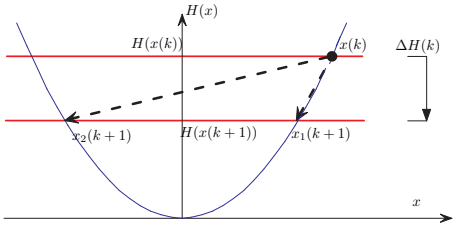

Fig. 3. Graphical representation of state update for systems with one state variable.

always tangent to the co-vector field $\partial H(x(t))/\partial x(t)$ and intersect perpendicularly the level curves.

In systems with one state variable only (e.g. a mass) the state $x(k+1)$ can be found by the energy function and the desired energy value. In most cases, multiple solutions belonging to a numerable set may be found, see Fig. 3, and the criteria for the choice can be quite simple.

On the other hand, systems with more than one state variable present in general an infinite cardinality set of states for each energy level. In the following, only linear systems are considered, since linearity guarantees that the energy is a quadratic function of the state and that all the field lines intersect all the level curves. The latter property is fundamental for the proof of the algorithm convergence.

Consider a system with only one state variable, as a spring or a mass. Its energy is a state function $H(x) : \mathbb{R} \rightarrow \mathbb{R}^+$. For any non-negative given value of H, there are only two possible states. Given the state $x(k)$, then $x(k+1)$ is chosen as the 'nearest' to the actual one. If some ambiguity emerges, the sign of the derivative of the state (from Eq. 1) can be considered. Fig. 3 shows the update strategy for this kind of systems, where $I_{k+1} = \{x_1(k+1), x_2(k+1)\}$ and $x_1(k+1)$ is the new state. In any case, only the definition of a metric permits to determine which is the 'nearest' state in the set I_{k+1}. In general, in case of systems with non homogeneous state variables, the only metrics that appear physically consistent are those based on energy. On the other hand, in this case each point of the level curve I_{k+1} is equidistant from any given state $x(k)$. Thus, a metric based only on energy is not suitable since the result is not unique.

As pointed out, the new state $x(k+1)$ must be selected in the set I_{k+1}. For this purpose, starting from $x(k)$ the most obvious update strategies are to move:

1) along the field lines;
2) along the direction of $\dot{x}$.

Example 2. Suppose to consider the system of Example 1, i.e. a mass linked to a spring, with state $x = (q,\ p)^T$ where q is the elongation of the spring and p the momenta of the mass. Consider an initial state $x_0(q_0, 0)$ and a null input force (i.e. $H(x(k)) = H(x(0))$ and $I_k = I_0, \quad \forall k \geq 0$).

The first update strategy moves the state along the field lines: since the field lines are perpendicular to the level curves, the only eligible state will be x_0. Conversely, with the second update strategy, the state moves along the direction of $\dot{x}$, that intersects I_0 only in x_0. Therefore strategies fail to update the state according to the physical behaviour, as shown in Fig. 4a.

Example 3. Suppose now that some energy flows out from the system. The next state $x(1)$ will belong to I_1, a set of states with an energy lower than x_0 encircled by I_0. If we move along the field lines, a new state $x(1) = (q_1, 0)$ is determined. This means that the spring has moved towards its rest length, but without any velocity variation of the mass. Moreover, the input influences the energetic level I, but does not help in the state selection in the eligible set. If we move along $\dot{x}$, that depends by $u(1)$, the intersection between the direction of $\dot{x}$ and I_1 is not assured. For these reasons, these two state update strategies do not seem to be completely satisfactory.

The solution proposed here is to use a two-step procedure. First, the traditional time integration of ODE (e.g. forward Euler) is used, in order both to keep track of how the system state evolves and to reflect the input effects on the state evolution. In this manner, an intermediate state $\overline{x}(k+1) = x(k) + \Delta x(k) T$ is reached, with $\Delta x(k)$ computed as in Eq. 4. Then, from $\overline{x}(k+1)$ the state is moved towards the set I_{k+1}, in order to correct the energy error produced by the traditional integration method. Since the field lines intersect all the energy levels, if the state evolves from $\overline{x}(k+1)$ along this path then it will reach the desired energy level I_{k+1}.

Fig. 4b illustrates the three methods in case of loss of energy[1]: the movement along the co-vector field ($x_o \rightarrow \overline{x}(1)$), the uncertainty of determining a solution along $\dot{x}$, and the proposed strategy ($x_0 \rightarrow \dot{x}T + x_0 \rightarrow x(1)$). In order

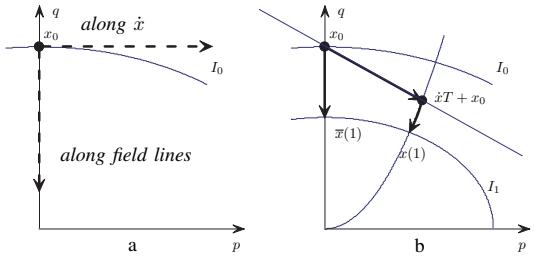

Fig. 4. Graphical representation of the update strategies in the phase space.

to move along a field line until the set I_{k+1} is met, three methods can be adopted:

1) if the explicit equations of field lines and level curves are known, the new state can be found analytically;
2) if the explicit equation of the field lines is known, the new state can be computed by iterating a bisection method, using the energy function as cost function;
3) otherwise the state can be numerically integrated along the field lines until I_{k+1} is met.

The first method can be used in a very limited class of problems, since the curve intersection cannot always be computed in a closed form.

The third method is more general, but it is time consuming

[1]The figure represent a mass/spring system with $km = 2$. Level curves are $kp^2 + q^2/m = 2H_0$; field lines are $p = (q/q_0)^{km} p_0$, with p_0, q_0 generic constant states, H_0 energy associated to the level curve.

and produces an error, i.e. the state $x(k+1)$ will not exactly belong to the set I_{k+1}. On the other hand this error is known, can be bounded, and is in the form of energy. Therefore, it can be registered and corrected in later iterations.
The second method is easy to implement and computationally fast. Also in this case an error is generated, but this can be compensated as in the third method. For these reasons, this technique has been adopted here.

The algorithm described above retains the passivity properties of the system, i.e. the energy errors generated by energy leaps and numerical integration will eventually come to zero.

C. Particular cases

There are two important particular cases that have to be considered separately:

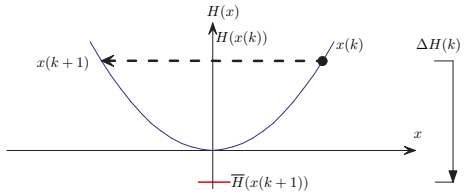

Fig. 5. Graphical representation of energy leap procedure in one variable systems.

1) *Energy leap, bookkeeping*
If $H(x(k+1)) = H(x(k)) + \Delta H < H_{min}$, the next state $x(k+1)$ cannot be determined. A solution to this problem has been proposed in [10]. The next state is selected in the set $I(k)$ (i.e. the energy level $H(x(k))$ does not change), injecting in the system an energy equal to $\|\Delta H(x(k))\|$. This is defined an *energy leap*. In order to preserve the overall passivity, the injected energy is logged (*bookkept*), and dissipated in later algorithm iterations. Unfortunately, the choice of the state $x(k+1)$ is not unique, and depends on the number and nature of the state variables. The only constraint is to keep track of the introduced energy and dissipate it in later algorithm iterations. As example of energy leap, let consider a linear system with only one state variable. If an energy leap occurs, an eligible next state will be $x(k+1) = -x(k)$, as depicted in Fig 5, where $\overline{H}(x(k+1))$ is the energy calculated for the next state, $x(k+1)$ is the next state and $\|\Delta H(x(k))\|$ is the bookkept energy.
2) *Dynamic deadlock*
The second problem occurs when the state is in the energy minimum (H_{min}). In this case, for any value of $u(k)$, the output $y(k)$ computed with Eq. 2, is null (since $\partial H(x)/\partial(x) = 0$), and from Eq. 7 it follows that $\Delta H(k)$ is null as well. The total system energy remains unchanged at the minimum value, and therefore the state does not evolve. This *deadlock* can be resolved by using a normal integration method for this step, as a forward Euler, in order to force the state out of minimum. This problem has been considered in details in [9], [10], and [11]. Note that this solution can possibly injects energy in the system but, since deadlock rarely happens, it cannot bring the system to instability.

III. MULTIRATE PORT HAMILTONIAN SYSTEMS

In this Section, a method to passively interconnect two 'systems' running at different frequencies is presented. The key element is a module that interfaces power variables that flow between a 'fast system', with sample time T_f, and a 'slow system', with sample time T_s. We assume that $T_s/T_f = N,\ N \in \mathbb{N}$ and that the two systems are synchronized.
In literature, two approaches have been proposed for this problem. The first is the method described in [9], [11], and [10] for interfacing continuous and discrete time; this technique has been used to passively connect a haptic interface to a virtual environment using an exact power balance at the end of each sample periods.

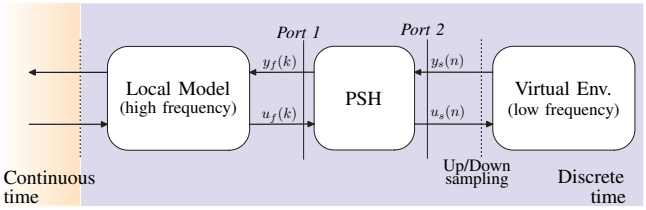

Fig. 6. Representation of the PSH element.

Somehow, this method is similar to the so-called *Passivity Observer/Passivity Controller* described in [6], with the main difference that the latter relies on a modulated damper that dissipates energy when the observed system becomes active.

In this paper, a different solution is proposed. The interface module is characterized by a feedforward action and by the fact that the power error $P(n)$ is logged (somehow, can be considered as the power traveling in a transmission line), and then not necessarily corrected in an iteration only. This element, referred to as the *Passive Sample and Hold* (PSH) in Fig. 6, is a dynamic system running at T_f with two inputs, $u_f(k)$ and $y_s(n)$, and two outputs, $u_s(n)$ and $y_f(k)$, where $u_f y_f$ and $u_s y_s$ are powers, and u_s, y_s are updated every N steps (since they are connected to the slower system). Without loss of generality, the power variables u and y can be regarded as scalars (in a system with multiple power variable couples, each couple generates an error that can be considered individually). The discrete time steps are indicated with k and n, with $n = \lfloor k/N \rfloor$, where $\lfloor\rfloor$ is the integer part operator.
During a period T_s, the power balance is expressed by:

$$P(n) = \frac{1}{N} \sum_{k=nN}^{(n+1)N-1} y_f(k)u_f(k) - [u_s(n)y_s(n)] \qquad (11)$$

where the left hand side of Eq. 11 represents the variation of power in the PSH element, and the right hand side is the power entering from *Port 1* and exiting *Port 2*, during the period T_s. The two outputs are defined as:

$$y_f(k) = y_s(n) \qquad k \in [nN, (n+1)N-1] \qquad (12)$$

$$u_s(n) = u_f(nN) + \frac{\Delta P(n)}{y_s(n)} \qquad (13)$$

where $\Delta P(n)$ is the power that flows in a period T_s from the power stored in the element PSH to the slow system. $\Delta P(n)$ has to be bounded in such a way the energy stored into the PSH module is brought to zero, without excessively distort the input signal $u_s(n)$. The critical behavior is met when $y_s(n)$ is near zero. The proposed strategy limits the influence of the power injected by the PSH element on the variation of the signal $u_s(n)$ (left hand term of Eq. 14), respect to the uncorrected one (right hand term of Eq. 14).

$$|u_s(n) - u_s(n-1)| \leq |u_f(nN) - u_s(n-1)| \tag{14}$$

Replacing Eq. 13 in Eq. 14:

$$\begin{aligned} &\left| \frac{\Delta P(n)}{y_s(n)} + u_f(nN) - u_s(n-1) \right| \\ &\leq |u_f(nN) - u_s(n-1)| \end{aligned} \tag{15}$$

The *inverse triangle inequality* states that $\big||a|-|b|\big| \leq |a+b|$. Therefore, the left hand term of Eq. 15 can be minimized as:

$$\begin{aligned} &\left| \left| \frac{\Delta P(n)}{y_s(n)} \right| - |u_f(nN) - u_s(n-1)| \right| \leq \\ &|u_f(nN) - u_s(n-1)| \end{aligned} \tag{16}$$

$$\begin{cases} \left| \frac{\Delta P(n)}{y_s(n)} \right| - |u_f(nN) - u_s(n-1)| \\ \qquad \leq |u_f(nN) - u_s(n-1)| \\ \left| \frac{\Delta P(n)}{y_s(n)} \right| - |u_f(nN) - u_s(n-1)| \\ \qquad \geq -|u_f(nN) - u_s(n-1)| \end{cases} \tag{17}$$

These conditions reduce to

$$|\Delta P(n)| \leq 2|u_f(nN) - u_s(n-1)|\,|y_s(n)| \tag{18}$$

Also, it's desirable that $E(n)$ goes to zero. This is obtained if the PSH corrects, in the step of length T_s, at most all the stored energy $E(n)$ (Eq.19), taking care to choose the correct sign (Eq.20); therefore, two additional constraints are added:

$$|\Delta P(n)|T_s \leq |E(n)| \tag{19}$$

$$sign(\Delta P(n)) = sign(E(n)) \tag{20}$$

In comparison with [6], the PSH does not always correct all the energy error in one step (i.e. when the condition expressed by Eq. 18 is tighter than the one of Eq. 19), so that $u_s(n)$ does not excessively vary respect to $u_f(nN)$, especially in some critical cases (e.g. $y_s \simeq 0$).

Once the power released by the system and its outputs are computed, the power stored in the PSH can be updated. From Eq. 11-13, the expression of $E(n)$ is calculated:

$$\begin{aligned} E(n+1) =& E(n) + T_s P(n) \\ =& T_s y_s(n) \left[\frac{1}{N} \sum_{k=nN}^{(n+1)N-1} u_f(k) - u_f(nN) \right] \\ &+ E(n) - T_s \Delta P(n) \\ E(0) =& 0 \end{aligned} \tag{21}$$

For the sake of convenience, instead of $E(n)$, the quantity $E(n)/T_s$ is used in the algorithm implementation, that is described by the following steps.

Algorithm 2

if $y_s(n) == 0$

$1_a)$ $\Delta P(n) = 0$, $E(n+1)/T_s = E(n)/T_s$, and $u_s(n) = u_f(nN)$,

else

$1_b)$ $\Delta P(n) = 2|u_f(nN) - u_s(n-1)|\,|y_s(n)|$,
$2_b)$ $\Delta P(n) = \min(\Delta P(n), |E(n)|/T_s)$,
$3_b)$ if $E(n)/T_s \leq 0$ then $\Delta P(n) = -\Delta P(n)$,
$4_b)$ $u_s(n)$, $E(n+1)/T_s$ are computed with Eq. 13,21.

IV. VALIDATION AND SIMULATION

A method to simulate both single- and multi-rate systems that retains the passivity properties has been described. From this perspective, the worst case is given by marginally passive systems and, therefore, the simulation of a spring/mass system without dissipation is presented. Two implementations of this system are presented: the first using a single Port Hamiltonian system with two state variables (Fig. 7a,c), the second with two interconnected Port Hamiltonian systems, with a state variable each (Fig. 7b,d). Moreover the simulation of a spring system, whose dynamics is computed at fast frequency, and a mass, simulated in a slow environment and connected by the PSH element, is also presented.

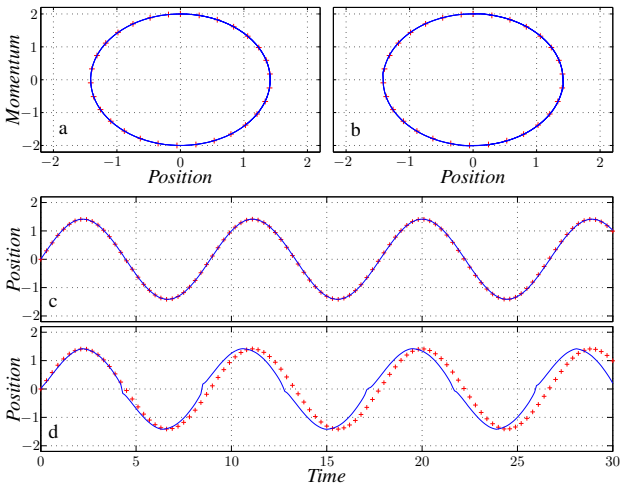

Fig. 7. a,b: Evolution of state variables. c,d: Time evolution of positions.

A. Simulation of single rate systems

The simulation results of single rate systems, i.e. with a single simulation step, are now illustrated. In particular, a mass/spring system is considered with: spring stiffness $k = 1$, mass $m = 2$, initial position $p_0 = 0$, initial velocity $v_0 = 1$, null input. This system has been simulated in two different manners: with only a single two-state variable Port Hamiltonian system, or with two interconnected one-state variable Port Hamiltonian systems, both with a step time of $0.1s$. Fig. 7a,c reports the free evolution of the two state

variable system, while Fig. 7b,d refers to the two-systems simulations. The continuous line represents the evolution of the system simulated with *Algorithm 1*, and the plus sign markers the evolution of a simulation with a variable step integration method (ode45). It is worth noticing, in Fig. 7d, the effect of energy leaps. When the position signal crosses the zero, an energy leap occurs generating a "discontinuity". This effect is not present in the two-state system (Fig. 7c), since the activation of the mechanisms of energy leap and the subsequent bookkeeping happens when the system energy (the sum of both kinetic and potential energy) is near zero, i.e. when both velocity and position are near zero.

Both system energies are constant, in the sense that the difference between the sum of energies and the sum of the energy bookkept is constant. Note that both systems present a variable delay, depending on the length of the simulation step, but larger in the second case.

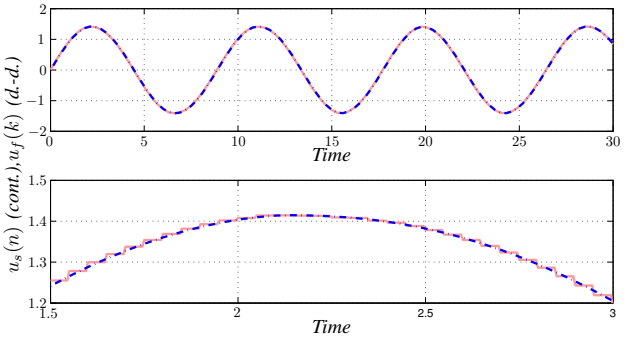

Fig. 8. (Up) Time evolution of PSH input $u_f(k)$ and PSH output $u_s(n)$, (Down) zoom of the previous figure.

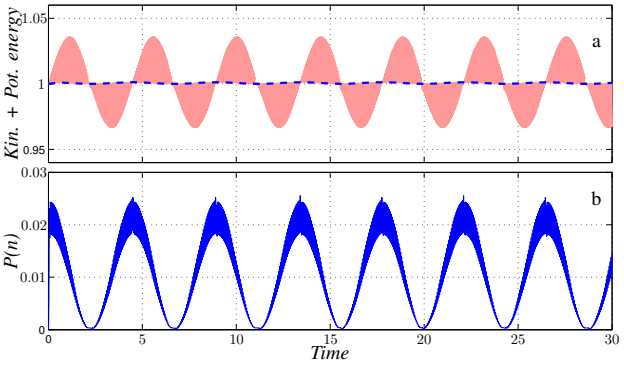

Fig. 9. a: Time evolution of the kinetic and potential energy sum. b: Power inside the PSH element.

B. Simulation of multi rate systems

The simulation of a multi rate mass/spring system connected by the PSH is now discussed. The simulation is run with the same values ($m = 2$, $k = 1$), while the time step lengths of the fast and slow environments are $T_f = 0.001s$ and $T_s = 0.05s$ respectively. As expected, the system results to be marginally stable. Fig. 8 shows the input $u_f(k)$ (dash dotted line) and output $u_s(n)$ (continuous line) of the PSH element; also, $u_f(k)$ is the spring force output, while $u_s(n)$ is the mass force input. Fig. 9.a shows the time evolution of the two subsystems energies sum; the gray area is drawn by the oscillating energy plotted at the faster sample time, while the dashed line represents the same data set plotted at the beginning of the slower sample time (i.e. when the PSH operates the correction and bring the total energy systems near the desired one, 1). Fig. 9.b represents the time evolution of the PSH energy error $E(n)$.

V. CONCLUSIONS

In this paper, an algorithm able to simulate continuous time systems expressed in Port Hamiltonian form, preserving system passivity, has been presented. The method considers linear systems but, being the stability criteria based on passivity, can be extended to nonlinear system. Moreover, the design of an element able to passively interconnect two systems running at different sample times is presented. Future developments will investigate the nonlinear case and how to improve the update strategy in order to have a more physically-related response.

REFERENCES

[1] F. Barbagli, D. Prattichizzo, and K. Salisbury. A multirate approach to haptic interaction with deformable objects single and multipoint contacts. *Int. J. Rob. Res.*, 24(9):703–715, 2005.

[2] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. In *Proc. IEEE ICRA '94*, pages 3205–3210, 1994.

[3] J. E. Colgate, P. E. Grafing, M. C. Stanley, and G. Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. In *Proc. IEEE Virtual Reality Symposium*, pages 202–208, 1993.

[4] N. Diolaiti, G. Niemeyer, F. Barbagli, and J. K. Salisbury. A criterion for the passivity of haptic devices. In *Proc. IEEE ICRA '05*, pages 2463–2468, 2005.

[5] N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. Salisbury, and C. Melchiorri. The effect of quantization and coulomb friction on the stability of haptic rendering. In *Proc. IEEE World Haptics Conference*, 2005.

[6] B. Hannaford and J. Ryu. Time domain passivity control of haptic interfaces. In *Proc. IEEE ICRA '01*, pages 1863–1869, 2001.

[7] B. Hannaford, J. Ryu, and Y. Kim. *Touch in Virtual Environments, Chapter 3, Stable Control of Haptics*. Prentice Hall, 2001.

[8] J. G. Park and G. Niemeyer. Haptic rendering with predictive representation of local geometry. In *Proc. IEEE Haptics Symposium*, pages 331–338, 2004.

[9] S. Stramigioli, G. Blankenstein, V. Duindam, H. Bruyninckx, and C. Melchiorri. Power-port concepts in robotics: the geometrical-physical approach (ICRA tutorial), 2003.

[10] S. Stramigioli, C. Secchi, A. van der Schaft, , and C. Fantuzzi. A novel theory for sample data system passivity. In *Proc.IEEE IROS '02*, 2002.

[11] S. Stramigioli, C. Secchi, A. van der Schaft, and C. Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Trans. on Robotics*, 21,4:574 – 587, 2005.