Proceedings of DSCC 2008
ASME 2008 Dynamic Systems and Control Conference
October 20-22, 2008, Ann Arbor, Michigan, USA

DSCC2008-XXXXX

ON PASSIVE NON-ITERATIVE VARYING-STEP NUMERICAL INTEGRATION OF MECHANICAL SYSTEMS FOR HAPTIC RENDERING

Dongjun Lee* **and Ke Huang**
Interactive & Networked Robotics Laboratory
Department of Mechanical, Aerospace & Biomedical Engineering
University of Tennessee
Knoxville, TN 37996
Email: {djlee,khuang1}@utk.edu

ABSTRACT

This paper consists of three parts. First, with a slightly-different, yet, more physically-plausible, discrete supply-rate (i.e. power), we propose a non-iterative (i.e. fast) and variable-step numerical integration algorithm for (scalar) discrete-passive mechanical systems, consisting of constant mass and damper, and a certain class of nonlinear spring. In the second part, we propose a fast passive collision handling algorithm with a spring-damper type virtual wall, which, to detect exact time of contacts, requires at most three intermediate non-iterative computations within each integration-step. We then propose a way of how to passively connect this discrete-passive, non-iterative, and variable-step mechanical integrators (with passive collision handling) to a continuous haptic device.

1 INTRODUCTION

Passive and explicit[1] numerical integration of mechanical systems is very desirable for haptic rendering, which in general requires fast updating. However, the well-known work [1] shows that such explicit passive numerical algorithms do not exist, unless the virtual tool mass is larger than a certain "minimum-mass". The implication of this is that, especially to correctly use the virtual-coupling [2] requiring discrete-passivity of virtual environments, we need either 1) to use implicit algorithms, which often result in slower updating-rate; or 2) to increase virtual tool mass as specified by "minimum mass" in [1], which may lead in less transparent system. See [3,4].

In the first part of this paper (Sec. 3), with a slightly-different, yet, seemingly more physically-plausible, discrete supply rate (i.e. power), we propose a novel non-iterative (i.e. iterations such as Newton-Raphson's method are not necessary), discrete-passive, and variable-step numerical integration algorithm for scalar mechanical systems, consisting of constant mass/damper and a certain class of nonlinear spring (e.g. spring potential is polynomial of order up to quintic). Although being implicit, its non-iterative nature still enables us to run this algorithm very fast. We believe that non-iterativeness of integration algorithms is all we need for fast-updating haptic rendering rather than their explicitness, although this explicitness automatically implies non-iterativeness. On the other hand, discrete-passivity of our proposed algorithm allows us to be completely free from the aforementioned transparency-compromising "minimum-mass" requirement.

Unilateral constraints, or, in this paper, interactions with a spring-damper-type unilateral virtual wall, are very important for haptic rendering, as they are ubiquitous in the real-world. Realistic simulation of such unilateral virtual walls should uphold passivity of the interaction (i.e. virtual walls should not generate energy by itself), since it is one of the most fundamental properties of the real-world.

In the second part of this paper (Sec. 4), we propose a novel passive collision handling algorithm, which enforces (discrete) passivity of the interaction between a virtual mass and an unilateral spring-damper virtual wall. The main idea of this passive collision handling is to split the integration time-steps T_i (see Fig. 1) at the exact collision times, if such collisions happen, so that we can avoid (passivity-breaking) spring energy jumps due to (undetected) deep-penetration of the virtual-mass into the wall, which then may create huge pushing-force, and thereby, excessive energy-injection into the virtual-mass, during the next

*Address all correspondence to this author.

[1] That is, the future state of numerical integration (e.g. x_{i+1}, v_{i+1} in Fig. 1) depends only on the previous one (e.g. x_i, v_i in Fig. 1). The term "implicit" is its antonyms, implying the future state not only depends on the previous one but also on itself too.

integration step T_{i+1}. This collision detection algorithm is also non-iterative (needs to solve only quadratic equation, for which closed-form solution available), and the maximum number of required collision detections within each integration time-step T_i is guaranteed to be less than two. Therefore, this passive collision handling algorithm can run very fast without being trapped in an infinite-loop. Some results related to this are [5] (stability rather than passivity), [6] (impulse-based rendering), and [7] (passivity of combined discrete virtual wall and continuous haptic device, rather than discrete passivity of virtual wall as done here).

The third part of this paper (Sec. 5) is devoted to the problem of how to interface this (discrete) passive mechanical virtual environment (with passive collision handling) with a real (continuous) haptic-device. Here, to accommodate variable integration-steps (i.e. T_i, T_j are different in Fig. 1) and possibly asynchronous date-communication between the virtual environment and the real haptic device, we apply the newly-proposed passive set-position modulation framework [8], which enables us to enforce continuous passivity for the haptic device and discrete passivity for the virtual environment individually, even with asynchronous data-sending/receiving between them. Note that the well-known virtual coupling [2] is not directly applicable here, since it requires uniform integration-step (i.e. $T_i = T_j$ in Fig. 1) and synchronous servo-rate for the haptic-device hardware [1,3]. In fact, our framework based on the passive set-position modulation can still enforce passivity even if the data-communication between the haptic-device and virtual-environment has varying-delay/packet-loss (e.g. haptic interaction with a virtual environment sitting in a remote computer over the Internet) [8].

The rest of this paper is organized as follows. We start Sec. 2 with some observations, showing that the often-used discrete passivity condition is not so natural. In Sec. 3, we propose the non-iterative, passive, and variable-step mechanical integrator for scalar mass-damper-(nonlinear) spring systems. Novel passive collision handling algorithm is presented in Sec. 4, and, passive interfacing with a continuous haptic-interface using the passive set-position modulation framework [8] is proposed in Sec. 5. Some concluding remarks and comments on future research is then given in Sec. 6.

2 SOME OBSERVATIONS

Often, the following discrete passivity is aimed (e.g. [1,9]): there exists a finite constant $c \in \Re$ s.t. for all $N \geq 0$,

$$\sum_{i=0}^{N} f_i v_i T_i \geq -c^2 \tag{1}$$

where $v_i, f_i \in \Re$ are the velocity/force at the i-th time-index, and T_i is the (discrete) integration time-step between the i-th and $(i+1)$-th time-indexes. See Fig. 1. This discrete passivity (1) is clearly a direct translation of its continuous counterpart - continuous passivity: for all $T \geq 0$,

$$\int_0^T f(t)v(t)dt \geq -c^2 \tag{2}$$

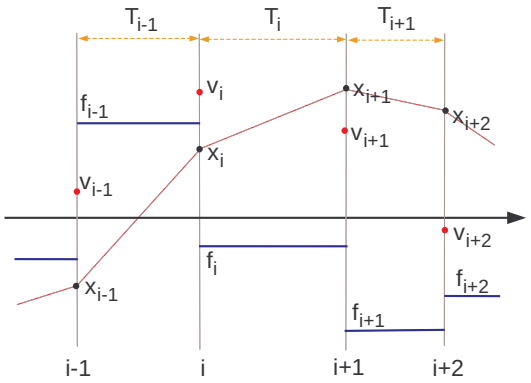

Figure 1. Data update in discrete numerical integration

where $c \in \Re$ is a finite constant.

To see why the discrete passivity condition (1) is not so natural, let us turn back to the continuous dynamics of linear scalar mass-spring-damper system:

$$v(t) = \frac{dx(t)}{dt} \tag{3}$$

$$m\dot{v}(t) + bv(t) + kx(t) = f(t) \tag{4}$$

where (3) is the kinematics equation, while (4) the dynamics equations. This continuous system possesses continuous passivity (2) with $c^2 := mv^2(0)/2 + kx^2(0)/2$ due to its energy conservation property: for all $t_1, t_2 \geq 0$

$$\frac{1}{2}mv^2(t_2) - \frac{1}{2}mv^2(t_1) + \frac{1}{2}kx^2(t_2) - \frac{1}{2}kx^2(t_1) = \int_{t_1}^{t_2} f(\sigma)v(\sigma)d\sigma - \int_{t_1}^{t_2} bv^2(\sigma)d\sigma. \tag{5}$$

Suppose that the system (3)-(4) is initially at rest (i.e. $x(0) = v(0) = 0$) and a non-zero force $f(t) \neq 0$ is applied during $[0,t)$. Then, we have: for all $t \geq 0$,

$$\frac{1}{2}mv^2(t) + \frac{1}{2}kx^2(t) = \int_0^t f(\sigma)v(\sigma)d\sigma - \int_0^t bv^2(\sigma)d\sigma$$

that is, with a non-zero force $f(t)$, its corresponding power-input leads in the change in the system energy, thereby, the motion of the system. So, for the continuous system, dynamic behavior and energetic behavior are consistent with each other.

Now, consider a discrete counterpart of (3)-(4) as given by the following explicit Euler kinematics-dynamics algorithm [1]:

$$x_{i+1} = x_i + T_i v_i \tag{6}$$

$$m\frac{v_{i+1} - v_i}{T_i} + bv_i + kx_i = f_i \tag{7}$$

which, similarly to [10], defines the updating map $L_{\mathrm{msd}} : \mathcal{X} \times$

$\mathcal{V} \times \mathcal{U} \to \mathcal{X} \times \mathcal{V}$ s.t.

$$L_{\mathrm{msd}} : (x_i, v_i, f_i) \to (x_{i+1}, v_{i+1})$$

where $\mathcal{X}, \mathcal{V}, \mathcal{U}$ are the spaces of the discrete position, velocity, and external force. See Fig. 1.

Then, with the same initial condition (i.e. $x_0 = v_0 = 0$ with non-zero $f_0 \neq 0$) as above, if we enforce the widely-used discrete passivity (1), energy-input during the integration step T_0 should be zero, since $v_0 = 0$. Yet, from the dynamics equation (7), the system will start moving, thus, the system energy increases *with no energy input*! Note that this observation still holds even we use different kinematics equations instead of (6) (e.g. implicit Euler or trapezoidal [1]). This clearly shows that the discrete passivity (1) is not so energetically natural. As long as we adopt it, any energetically-correct behavior of the discrete system may not be achievable.

Comparing to the continuous case, the key problem here is that v_0 is fixed to zero during the time-step T_0 even with non-zero force f_0, although this is not the case for its continuous counterpart (i.e. velocity continuously changes). This, we believe, is essentially the idea behind the non-existence proof of explicit passive numerical integrations in [1], that is, f_0 may be arbitrary, yet v_0 is fixed to be a given value during the time of applying f_0. Similar problem of breaking discrete energy preservation with zero velocity is also confronted in [10, 11].

Therefore, we argue that, instead of the seemingly-straightforward discrete passivity (1), we should use (or aim for) another condition. For this, let us consider the total energy change in the discrete system and see how (or if) we can match it with its continuous counterpart (5):

$$\begin{aligned}
&\frac{1}{2}mv_{i+1}^2 - \frac{1}{2}mv_i^2 + \frac{1}{2}kx_{i+1}^2 - \frac{1}{2}kx_i^2 \\
&\quad = m(v_{i+1} - v_i)\frac{v_{i+1} + v_i}{2} + \frac{1}{2}k(x_{i+1} + x_i)(x_{i+1} - x_i) \\
&\quad = \frac{v_{i+1} + v_i}{2}(f_i - bv_i - kx_i)T + \frac{1}{2}k(x_{i+1} + x_i)(x_{i+1} - x_i)
\end{aligned}$$

where we use (7). Although not solving the problem here, this provides some valuable insights: 1) it seems energetically more compelling to choose $(v_{i+1} + v_i)/2$ as the representative velocity during the integration-step T_i than v_i, with the following new discrete passivity condition: there is a finite constant $c \in \Re$ s.t.

$$\sum_{i=0}^{N} f_i \frac{v_{i+1} + v_i}{2} T_i \geq -c^2, \quad \forall N \geq 0 \tag{8}$$

instead of the unnatural (1); 2) using this $(v_{i+1} + v_i)/2$ in the damping (i.e. $b(v_{i+1} + v_i)/2$ for (7)) will also guarantee the negative-definiteness of damping dissipation as in the continuous case; and 3) we also need to eliminate the spring-related terms (i.e. with k) in this discrete energy-relation to match with (5).

Another point we want to make here is about the discrete kinematics equation (6), which is a straightforward translation of the continuous kinematics equation (3), essentially assuming zero acceleration during T_i. However, for the discrete case, the acceleration during T_i is rather *freezed*, thus, constant. This observation leads us to the following kinematic relations:

$$v_{i+1} = v_i + a_i T_i, \quad x_{i+1} = x_i + v_i T_i + \frac{1}{2}a_i T_i^2$$

where a_i is a constant acceleration during T_i. Combining these two, we can then achieve the following discrete kinematics equation:

$$\frac{x_{i+1} - x_i}{T_i} = \frac{v_{i+1} + v_i}{2} \tag{9}$$

which matches well with the above energy argument, suggesting $(v_{i+1} + v_i)/2$ as the representative velocity during T_i. Note that this kinematics equation (9) is actually the trapezoidal rule, which is known to have superior energetic property [1].

3 PASSIVE NON-ITERATIVE VARIABLE-STEP SIMULATION OF MECHANICAL SYSTEMS

As suggested above, here, we adopt $(v_{i+1} + v_i)/2$ as the representative velocity during the integration-step T_i, along with the discrete passivity (8) and the kinematics equation (9). Now, consider a scalar mechanical system, consisting of constant mass/damper and a certain class of nonlinear spring with its potential energy denoted by $\varphi : \mathcal{X} \to \Re^+$. We use the notation $\varphi_i := \varphi(x_i)$, i.e. the spring potential at the time-index i.

Then, similar to (7) and following the observations above, we can think of the following numerical integration algorithm for this mass-damper-(nonlinear)spring system:

$$m\frac{v_{i+1} - v_i}{T_i} + b\frac{v_{i+1} + v_i}{2} + d\varphi_i = f_i \tag{10}$$

where $d\varphi_i$ is the force generated by φ during T_i (to be determined below). Here, we want this discrete dynamics (10) to duplicate the continuous energy preservation relation (2), that is,

$$\begin{aligned}
&\frac{1}{2}mv_{i+1}^2 - \frac{1}{2}mv_i^2 + \varphi_{i+1} - \varphi_i \\
&\quad = \frac{v_{i+1} + v_i}{2}\left[-b\frac{v_{i+1} + v_i}{2} - d\varphi_i + f_i\right]T_i + \varphi_{i+1} - \varphi_i \\
&\quad = -b\left(\frac{v_{i+1} + v_i}{2}\right)^2 + f_i\frac{v_{i+1} + v_i}{2}T_i
\end{aligned} \tag{11}$$

where we use (10) for the second-line, while the third-line is the target relation from (5).

To equate the second and third lines, we need to have

$$d\varphi_i \frac{v_{i+1} + v_i}{2} T_i = d\varphi_i [x_{i+1} - x_i] = \varphi_{i+1} - \varphi_i$$

where we use (9). Then, using the Taylor's series, we have

$$d\varphi_i[x_{i+1}-x_i] = \frac{\partial\varphi}{\partial x}[x_{i+1}-x_i] + \frac{1}{2}\frac{\partial^2\varphi}{\partial x^2}[x_{i+1}-x_i]^2 + \ldots.$$

thus, we can choose $d\varphi_i$ during T_i s.t.

$$d\varphi_i = \frac{\partial\varphi}{\partial x} + \frac{1}{2}\frac{\partial^2\varphi}{\partial x^2}[x_{i+1}-x_i] + \frac{1}{3!}\frac{\partial^3\varphi}{\partial x^3}[x_{i+1}-x_i]^2 + \ldots \quad (12)$$

where all the partial derivatives are computed at $x = x_i$.

For instance, if we employ the frequently-used quadratic spring potential $\varphi(x) = k(x-y)^2/2$ with a constant y, we have

$$d\varphi_i = k\left(\frac{x_{i+1}+x_i}{2} - y\right)$$

and the discrete dynamics (10) is given by: with rewriting the kinematics equation (9),

$$m\frac{v_{i+1}-v_i}{T_i} + b\frac{v_{i+1}+v_i}{2} + k\left(\frac{x_{i+1}+x_i}{2} - y\right) = f_i \quad (13)$$

$$\frac{x_{i+1}-x_i}{T_i} = \frac{v_{i+1}+v_i}{2} \quad (14)$$

which defines passive variable-rate integration update algorithm: $L^p_{\text{msd}} : \mathcal{X}\times\mathcal{V}\times\mathcal{U} \to \mathcal{X}\times\mathcal{V} : (x_i, v_i, f_i) \mapsto (x_{i+1}, v_{i+1})$.

Here, note that this algorithm (13)-(14) (with linear spring) is implicit, since, for instance, the second and third terms of (13) contain the future information v_{i+1}, x_{i+1}. Yet, if we rewrite (13) using (14) w.r.t. x_{i+1} (or v_{i+1}, resp.), we can solve (13) for x_{i+1} (or v_{i+1}, resp.) without any time-consuming iterations such as Newton-Raphson method. Once we achieve this x_{i+1} (or, v_{i+1}, resp.), we can then obtain v_{i+1} (or x_{i+1}, resp.) via (14). Therefore, we can perform this passive and variable-rate numerical integration very fast, even if it is implicit. We believe that this non-iterativeness of numerical algorithms is all we need for fast haptic rendering rather than its explicitness, although, of course, explicitness of algorithms automatically implies their non-iterativeness.

Note that this algorithm (13)-(14) will still be non-iterative even with nonlinear spring, as long as its potential function φ can be well-approximated by Taylor series of order up to five (i.e. quintic polynomial), since, in this case, $d\varphi_i$ in (12) will be a quartic polynomial (i.e. fourth order), thus, the dynamics equation (13), along with (14), will become just a quartic polynomial equation w.r.t. x_{i+1}, for which a closed-form solution exists. If the potential φ can not be represented or approximated by this class of nonlinear functions, we will then need some iterations to solve the nonlinear algebraic equation (13) w.r.t. x_{i+1}.

Simulation results of this passive, variable-rate, and non-iterative integration algorithm (13)-(14) applied to mass-(linear)spring system (no damping), along with those of continuous Runge-Kutta fourth-order algorithm and Euler explicit

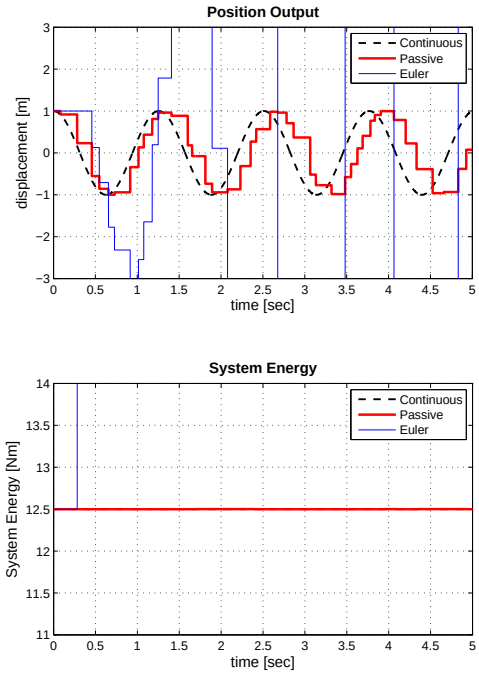

Figure 2. Simple harmonic oscillation of mass-spring system: integration-step varying randomly between 0.05 to 0.2sec.

kinematics-dynamics algorithm [1], are shown in Fig. 2. The integration-steps are varying randomly between 0.05 to 0.2sec for our and explicit Euler algorithms, while fixed to be 0.001sec for the Runge-Kutta algorithm. From Fig. 2, it is clear that our proposed algorithm maintains the stability of simple harmonic oscillation and, even more, keeps the total energy to be constant exactly as in the continuous-dynamics simulation. In contrast, the Euler explicit algorithm produced energy by itself and becomes unstable very quickly. We also perform another simulation with some external force (not shown here), and found that our proposed algorithm closely matches passivity-relation (i.e. energy-input $\sum F_i f_i (v_{i+1}+v_i)/2$) of the continuous case, while the Euler algorithm again goes quickly unstable. Some delays between our algorithm and the Runge-Kutta is due to the fundamental difference in the integration-steps.

4 PASSIVE COLLISION HANDLING

Consider a virtual wall (with spring k and damping b) spanning from y to $+\infty$, and suppose that a virtual mass m is outside of this wall at the i-th time-index, that is, $x_i < y$. Suppose further that there is no dedicated collision handling subroutine. Then, during T_i, we will use the discrete dynamics (13) with $b = k = 0$, without knowing *a prior* if collision will occur or not during this T_i. Now, suppose that a collision indeed occurs during this T_i and x_{i+1} computed by using (13) with $b = k = 0$ is now inside the wall s.t. $x_{i+1} \geq y$. Then, at the next time-step T_{i+1}, we suddenly change the system dynamics (13) with non-zero b and k. This induces a sudden jump in the virtual wall's potential energy from

zero to $\varphi_{i+1} = k(x_{i+1} - y)^2/2$, while the kinetic energy of the virtual mass $mv_{i+1}^2/2$ is not changed. This sudden jump in the wall potential energy can violate passivity: by pushing the virtual mass m back with huge force during T_{i+1}, it may inject more energy into m than supplied by f_i during the previous integration step T_i.

To avoid this possibly passivity-breaking virtual wall potential energy jumps, in this work, we find exact time of collisions, and split the time-step T_i at these collision times. By doing so, we can prevent such sudden passivity-violating jumps in the wall spring potential, and, thereby, can enforce discrete passivity (8) of the combined virtual mass and virtual wall. More specifically, for the above scenario, we will first find $0 < T_1 \leq T_i$ s.t.

$$m\frac{v_{i+1} - v_i}{T_1} = f_i, \quad \frac{y - x_i}{T_1} = \frac{v_{i+1} + v_i}{2}$$

which is obtained from (13)-(14) with $x_{i+1} = y$ and $b = k = 0$. Here, note that f_i is constant during the integration step T_i, while $x_i < y$. Combining these two equations, we have:

$$2m(y - x_i - v_i T_1) = f_i T_1^2 \tag{15}$$

which is a quadratic equation, thus, easy to check the existence of its solution T_1 s.t. $0 < T_1 \leq T_i$ without any iterations. If there exists such T_1, we can then say that the virtual mass m has a collision with the wall at T_1 during the integration step T_i. See Fig. 3.

We can also achieve a similar quadratic equation for the case of leaving the wall (i.e. $x_i \geq y$, yet, $x_{i+1} < y$) by combining (13)-(14) with $x_{i+1} = y$ and non-zero b, k. This is also true for the spring potential φ of up to quintic order, since, in this case, from (10) and (12), we still have quadratic algebraic equation w.r.t. T_1 with $d\varphi_i$ given as a function of (known) x_i and $y = x_{i+1}$. Since the following argument is easily extended to the case of $x_i \geq y$ (i.e. starting from inside of the wall) and also for the case of up to quintic-order polynomial spring potential φ, from now on, we will just focus on the case of $x_i < y$ (i.e. starting outside of the wall) and linear spring (i.e. quadratic φ). We will also use T_1, T_2, T_2 as sub-intervals (i.e. set) of T_i or as time-steps (i.e. positive number) associated to successive collisions as depicted in Fig. 3

Now, suppose that if the number of possible collisions during any integration step T_i can be very large or even infinite. This then may result in very slow data-update from the virtual world to the haptic device (due to many subroutines of above collision detections), or even may trigger infinite-loops in the integration algorithm. As shown below, however, this turns out to be not possible. In fact, the maximum number of possible collisions within one integration step T_i (with x_i, v_i, f_i fixed) can not exceed two, implying that neither enter-leave-enter or leave-enter-leave are possible during any integration step T_i.

To show this, consider again the above scenario where $x_i < y$ and the first collision happened at $T_1 \subset T_i$, that is, this T_1 satisfies (15). Let us denote the position of the virtual mass at the end

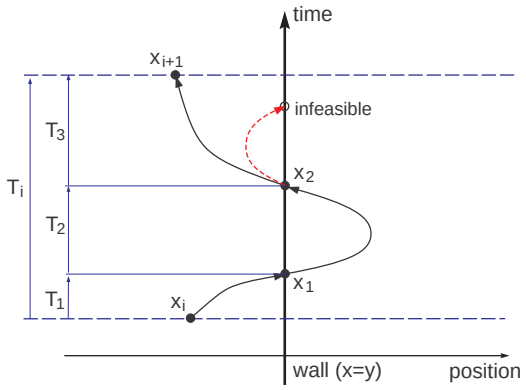

Figure 3. At most, twice collisions are possible during one integration time-step T_i

of this T_1 by x_1 (see Fig. 3). Of course, here, $x_1 = y$, since the virtual mass makes the first contact at the end of T_1.

Now, suppose that, during T_i, the virtual mass m leaves the wall. Let us denote the time-duration from x_1 to this wall-leaving by $T_2 \subset T_i$ and denote the position at the end of this T_2 by x_2. Again, we have $x_2 = y$. See Fig. 3. Then, since, during this T_2, the virtual mass m moves within the wall, there should exist $0 < T_2 \leq T_i - T_1$ s.t.

$$m\frac{v_2 - v_1}{T_2} + b\frac{v_2 + v_1}{2} + k\left(\frac{x_2 + x_1}{2} - y\right) = f_i, \quad \frac{x_2 - x_1}{T_2} = \frac{v_2 + v_1}{2}$$

where $x_2 = x_1 = y$, thus, we can simplify these two equations s.t.

$$m\frac{v_2 - v_1}{T_2} = f_i, \quad v_2 + v_1 = 0 \tag{16}$$

which is a condition necessary for this wall-leaving. Suppose further that the virtual mass m enters again the wall during T_i and denote the time-interval from x_2 to this wall-entering by $T_3 = T_i - T_1 - T_2$. Then, during this T_3, the virtual mass stays outside the wall, thus, its update equation is given by:

$$m\frac{v_3 - v_2}{T_3} = f_i, \quad \frac{x_3 - x_2}{T_3} = \frac{v_3 + v_2}{2}$$

where $x_3 = x_2 = y$, thus, we can simplify these two equations by

$$m\frac{v_3 - v_2}{T_3} = f_i, \quad v_3 + v_2 = 0 \tag{17}$$

which is necessary for this wall-reentering.

Therefore, for the virtual mass m to enter, leave, and re-enter the wall within one integration step T_i, the following two conditions should be necessarily met:

$$-2mv_1 = f_i T_2 \quad \text{and} \quad 2mv_1 = f_i T_3 \tag{18}$$

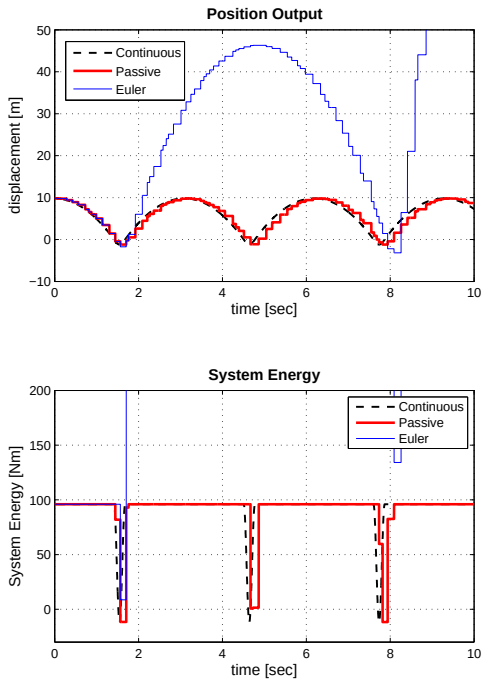

Figure 4. Bouncing ball simulation: integration-step varying randomly between 0.05 to 0.2sec.

where we combine (16) and (17) with $v_1 = -v_2 = v_3$. However, since f_i is constant during T_i and $T_2, T_3 > 0$, these two conditions can not be satisfied simultaneously unless $f_i = 0$. Now, suppose that $f_i = 0$. Then, this is only possible with $v_1 = v_2 = v_3 = 0$ and $x_1 = x_2 = x_3 = y$ with $f_i = 0$, that is, from the end of T_1, the virtual mass m will just stay at the boundary of the wall without any further move whatsoever. Similar can also be shown for the impossibility of leaving-entering-leaving scenario, that is, if the virtual mass m start from and leaves the wall, it can re-enter the wall only once and, if it does, will stay within the wall thereafter, not leaving the wall again. Or, if $f_i = 0$, it may stay at the boundary of the wall.

The above observation implies that, during any integration step T_i, we can just terminate the passive collision handling (e.g. (15)) after two collisions are detected (e.g. x_1, x_2 in Fig. 3), and simply integrate the remaining time-step $T_3 = T_i - T_1 - T_2$, with either zero b, k if the virtual mass at x_2 is leaving the wall or non-zero b, k if it is about to enter the wall at x_2. Also, if $f_i = 0$ and $x_i = y, v_1 = 0$ is detected, we can simply proceed to the next integration step T_{i+1} with $x_{i+1} = y, v_{i+1} = 0$, since there will be no move by the virtual mass.

This also implies that, if the virtual mass is close to the virtual wall so that collisions are possible, we need to slow down the data sending from the virtual environment to the haptic device by $3T_{\text{comp}}$ where T_{comp} is the worst-case computing time to finish one sub-interval collision handling (i.e. time required to finish T_2). In most case, however, we believe that this still can be done fairly fast, since its integration during T_1, T_2, T_3 (e.g. (13) with (15) are all non-iterative with their closed-form solutions available. For instance, during our one-DOF virtual wall haptic experiment in Sec. 5, we achieve $3T_{\text{comp}} < 1ms$ using just an ordinary office PC.

We perform a simulation of a bouncing ball under the gravity and pure-spring virtual wall (no damping) with our passive collision handling algorithm. The results are given in Fig. 4, where it is clear that, even with randomly-varying integration steps, our combined passive non-iterative integration and collision handling algorithms can maintain the stable ball bouncing just as with the continuous Runge-Kutta ODE solver. This clearly shows that our algorithms preserve passivity of the combined ball and the virtual wall by enforcing conservative energy-shuffling among the gravitation, wall spring, and ball's velocity. In contrast, the Euler explicit algorithm produces energy by itself and quickly becomes unstable.

5 PASSIVE INTERFACING WITH HAPTIC DEVICE

Let us now turn to the problem of interfacing this discrete passive virtual world (e.g. virtual mass + virtual wall with passive collision handling (PCH) - see Fig. 5) with the real haptic device. For this, in this work, we utilize the recently-proposed passive set-position modulation (PSPM) framework [8], which enables us to connect the virtual mass and the haptic device via virtual-coupling like interconnection, while enforcing continuous passivity of the haptic device and discrete passivity of the virtual world individually, even with variable integration-steps and asynchronous data-communication between the real and virtual worlds. Due to the space constraint, our treatment here is necessarily not comprehensive and we refer readers to [8] for more details.

Similar to the virtual-coupling [2], as shown in Fig. 5, we consider the virtual mass m as a two port system, interacting both with the haptic device via the PSPM module and a discrete passive virtual environment (e.g. virtual wall with passive collision handling - right-most block of Fig. 5) in the virtual world. Then, similar to (13)-(14), the simulation algorithm can be written by

$$m\frac{v_{i+1}-v_i}{T_i} = \tau_i + f_i, \quad \frac{v_{i+1}+v_i}{2} = \frac{x_{i+1}-x_i}{T_i} \tag{19}$$

where f_i is the interaction force with the virtual environment, while τ_i embeds the PSPM coupling given by

$$\tau_i = -b_c\frac{v_{i+1}+v_i}{2} - k_c\left(\frac{x_{i+1}+x_i}{2} - q_i\right) \tag{20}$$

where q_i is the haptic device's (continuous) position signal $q(t)$ received at the time-index i. During T_i, this q_i is constant.

Now, suppose that the virtual environment in Fig. 5 is discrete passive (e.g. virtual wall + PCH). Then, as long as we can render the PSPM block to be also discrete passive, the total interconnection of the virtual world in Fig. 5 will be passive, thus, stable to interact with. Recall from Sec. 3 that, if q_i is fixed across different i, this PSPM coupling (i.e. with $(v_{i+1}+v_i)/2$

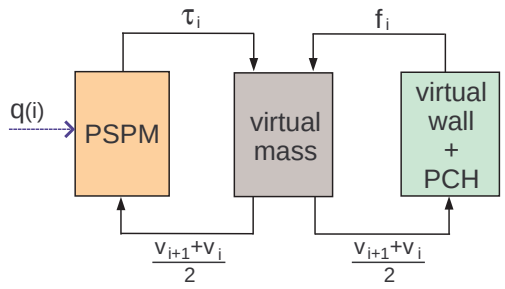

Figure 5. Illustration of the virtual world consisting of virtual-mass, passive set-position modulation (PSPM), and virtual environment (e.g. virtual-wall + passive collision handling (PCH))

and τ_i as input and output) will be guaranteed to be discrete passive. However, due to the switchings of q_i, the energy in the PSPM spring k_c may jump and, thereby, can violate passivity. This can be observed from (11): if $q_i \neq q_{i+1}$, by combining (11) for T_i, T_{i+1}, we have

$$\frac{1}{2}mv_{i+2}^2 - \frac{1}{2}mv_i^2 + \frac{1}{2}k_c(x_{i+2}-q_{i+1})^2 - \frac{1}{2}k_c(x_i-q_i)^2 = \Delta P_{i+1} - \sum_{k=i}^{i+1} b_c \frac{v_{k+1}+v_k}{2}T_k + \sum_{k=i}^{i+1} f_k \frac{v_{k+1}+v_k}{2}T_k$$

where $\Delta P_{i+1} = k_c(x_{i+1}-q_{i+1})^2/2 - k_c(x_{i+1}-q_i)^2/2$ is the (possibly non-passive) energy jumping in the spring k_c. Note that, if $\Delta P_{i+1} = 0$, this recovers to the passive relation of (11).

To avoid this potentially passivity-breaking energy jumps in k_c, we apply the PSPM framework [8] here. First, at the i-th time-index, we compute the damping dissipation in the previous step T_{i-1}, i.e. with (14), $D_{i-1} = b(x_i - x_{i-1})^2/T_{i-1}$. Then, given q_i, as long as $\Delta P_i \leq D_{i-1}$, discrete passivity will be granted, since, by doing so, we can bound the jumping ΔP_i below than the passivity-enforcing D_{i-1}. However, if we start with stationary initial condition so that $D_0 = 0$, no further jumping will be possible, thus, the virtual mass will just stay where it was, even if the set-position signals q_i are varying. To avoid this "start-off" problem, we augment the PSPM with a virtual energy reservoir E_2 with a non-zero initial energy. Then, following [8], we solve, at each i-th time-index, $\bar{q}_i$, the modulated version of q_i, s.t.:

$$\min_{\bar{q}_i} \quad ||q_i - \bar{q}_i||$$
$$\text{subj. } E_2(i) = E_2(i-1) + D_{i-1} - \Delta\bar{P}_i \geq 0$$

where $\Delta\bar{P}_i$ is ΔP_i computed with q_i replaced by $\bar{q}_i$. Then, we use this $\bar{q}_i$ instead of q_i in (20). Note that the second line inequality here enforces discrete passivity by limiting $\Delta\bar{P}_i$ below than available energy in the system (i.e. $E_2(i-1) + D_{i-1}$).

Similarly, we use the following PSPM coupling control for the haptic device: $\tau(t) = -b_1\dot{q}(t) - k_1(q(t) - x_k)$, where x_k is received discrete virtual-mass position at a certain time t_k. Then, switchings of x_k will also induce energy jumps in the spring k_1, and, to passify this, we can just use the standard PSPM with virtual energy reservoir E_1 [8]. Note that this PSPM coupling converges to the virtual coupling [2], when we have enough energy in E_1, E_2, although, here, this PSPM coupling may be asymmetric (i.e. $b_1 \neq b_c, k_1 \neq k_c$) and can be tuned locally.

Using this PSPM coupling and the (linear) virtual wall (+ passive collision handling) as the virtual environment of Fig. 5, we perform a virtual-wall experiment. Note that, with the PSPM coupling in (20), 1) the numerical integration algorithm of the virtual world (i.e. virtual mass + virtual environment + PSPM coupling) is still non-iterative; and 2) the results of the passive collision handling also still holds. This is because the structure of the dynamics equation (13) is preserved even with the PSPM coupling (20). In fact, for the collision handling of Sec. 4, we can show that, with (20), the contradiction condition (18) becomes:

$$-2mv_1 = k_c(y - \bar{q}_i)T_2 \quad \text{and} \quad 2mv_1 = k_c(y - \bar{q}_i)T_3$$

which is only feasible with $y = \bar{q}_i$ and $v_1 = v_2 = 0$, that is, the desired set-position $\bar{q}_i$ is at the boundary of the virtual wall, and the virtual mass m should stay there during the remaining of T_i. We also implement energy shuffling/ceiling, that is, if E_1 becomes larger than a certain value $\bar{E}_1$, we send $\Delta E_1 = E_1 - \bar{E}_1$ to the virtual mass side, and vice versa (with some scalings). This is especially useful when the task requires the human to keep injecting energy into the system.

Experimental results are given in Fig. 6, where the plotted master force is scaled by ten. This is because, since we set $(k_1, k_c) = (500, 5000)$N/m (with $(b_1, b_c) = (10, 10)$Ns/m), human contact force will be ten times less than the contact force in the virtual world. Of course, our PSPM framework enforces passivity even with this force scalings. See [8]. We also set the virtual wall stiffness/damping to be $k = 30$kN/m and $b = 28.3$Ns/m, and the virtual mass to be 0.001kg. We use a single PC with 2.13GHz CPU and 2GB RAM to run all the required numerical algorithms, simple 3D OpenGL graphics, and servo-loop for Novint Falcon haptic device. The data-communication between the virtual world and the haptic device is asymmetric, asynchronous, and randomly varying s.t. during about 80% of running time, the update rate is 2ms, while about 20%, it is randomly varying from 2ms to 5ms. As clearly seen in Fig. 6, even with this asymmetric and variable data update, haptic interaction is very stable, and the human can perceive the correct virtual contact force, in part due to the very light virtual mass.

Here, probably the most interesting aspect of our framework is that the parameters of the virtual world (e.g. virtual mass, stiffness/damping of the virtual wall, variable integration-steps) can be chosen totally independently from the continuous haptic device part (e.g. device servo-rate/damping). For instance, we can set the virtual mass arbitrary small and the stiffness of the virtual wall arbitrary large, without being restricted by the device-damping/sampling-rate related constraints (e.g. [1, 7]). For instance, roughly-computed minimum mass according to [1] is 0.2kg, yet, our virtual mass is 0.001kg. This can be possible here, since 1) the virtual world is guaranteed to be discrete passive in the sense of (8); and 2) the PSPM framework enforces

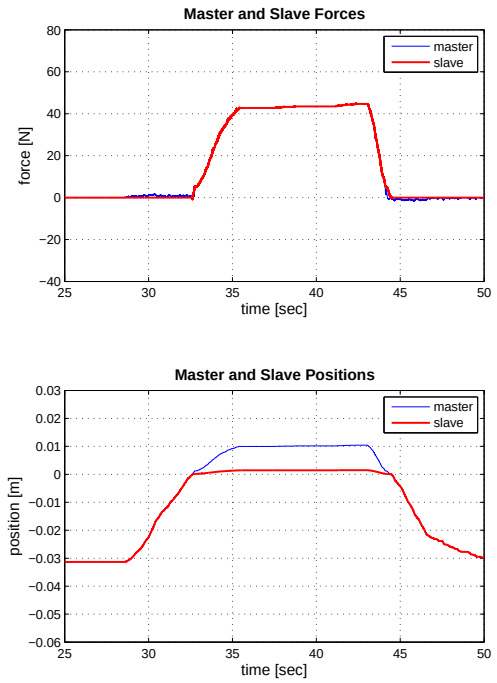

Figure 6. Haptics experiment with virtual wall.

continuous passivity and discrete passivity separately. This complete separation between the device servo-loop and the virtual world, we believe, will be very useful in practice, since these two aspects can be completely independently constructed and developed, as originally aimed for in [2]. Of course, here, the system performance will be limited by the haptic device hardware (e.g. larger k_1 only possible with faster servo-rate, which, in our case, is limited by the hardware to be 2ms), although this again will not affect the construction of the virtual world.

6 Conclusion and Future Works

In this paper, we propose a novel discrete-passive, non-iterative, and variable-step numerical integration algorithm for haptic rendering of scalar mechanical systems, consisting of constant mass/damping and a certain class of nonlinear spring. We also provide a passive collision handling algorithm between the virtual-mass and spring-damper type virtual-wall and a way of how to interface this discrete passive virtual environment to continuous haptic device while enforcing passivity even with variable numerical integration steps.

There are many possible directions for future research, and among them, of particular interests to us is how to extend this proposed framework to a general multi-dimensional nonlinear discrete mechanical systems (e.g. serial-chain robots/attitude dynamics), where we need to incorporate configuration-dependent inertia/Coriolis matrices, as well as higher-dimensionality of collision detection/handling (e.g. complicated collision surface).

REFERENCES

[1] Brown, J. M., and Colgate, J. E., 1998. "Minimum mass for haptic display simulations". In Proceedings of ASME International Mechanical Engineering Congress and Exposition, pp. 249–256.

[2] Colgate, J. E., Stanley, M., and Brown, J. M., 1995. "Issues in the haptic display of tool use". In Proceedings of IEEE/RSJ International Conf. on Intelligent Robots and Systems, Vol. 3, pp. 140–145.

[3] Adams, R. J., and Hannaford, B., 1999. "Stable haptic interaction with virtual environments". *IEEE Transactions on Robotics and Automation,* **15**(3), June, pp. 465–474.

[4] Otaduy, M. A., and Lin, M. C., 2006. "A modular haptic rendering algorithm for stable and transparent 6-dof manipulation". *IEEE Transactions on Robotics,* **22**(4), pp. 751–762.

[5] Gillespie, R. B., and Cutkosky, M. R., 1996. "Stable user-specific haptic rendering of the virtual wall". In Proceedings of ASME International Mechanical Engineering Congress and Exposition, pp. 397–406.

[6] Constantinescu, D., Salcudean, S. E., and Croft, E. A., 2005. "Haptic rendering of rigid contats using impulsive and penalty forces". *IEEE Transactions on Robotics,* **21**(3), pp. 309–323.

[7] Colgate, J. E., and Schenkel, G., 1997. "Passivity of a class of sampled-data systems: application to haptic interfaces". *Journal of Robotic Systems,* **14**(1), pp. 37–47.

[8] Lee, D. J., and Huang, K., 2008. "Passive position feedback over packet-switching communication network with varying-delay and packet-loss". In Symposium of Haptic Interfaces for Virtual Environments & Teleoperator Systems. Also available at http://web.utk.edu/~djlee/papers/Haptics08.pdf.

[9] Hannaford, B., and Ryu, J.-H., 2002. "Time domain passivity control of haptic interfaces". *IEEE Transactions on Robotics and Automation,* **18**(1), pp. 1–10.

[10] Wendlandt, J. M., and Marsden, J. E., 1997. "Mechanical integrators derived from a discrete variational principle". *Physica D,* **106**(3-4), pp. 223–246.

[11] Simo, J. C., Tarnow, N., and Wong, K. K., 1992. "Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics". *Computer Methods in Applied Mechanics & Engineering,* **100**, pp. 63–116.