Federico Barbagli

Stanford Robotics Laboratory
Stanford University, Stanford, CA, USA and
Information Engineering Department
University of Siena, Siena, Italy

Domenico Prattichizzo

Information Engineering Department
University of Siena, Siena, Italy
prattichizzo@dii.unisi.it

Kenneth Salisbury

Stanford Robotics Laboratory
Stanford University, Stanford, CA, USA

A Multirate Approach to Haptic Interaction with Deformable Objects Single and Multipoint Contacts

Abstract

In this paper we describe a new solution for stable haptic interaction with deformable object simulations featuring low servo rates and computational delays. The solution presented is a combination of the local model and the virtual coupling concepts proposed in the past. By varying the local model impedance depending on the local stiffness of the deformable object, the interaction between local model and simulation can always be made stable independently of low servo rates or computational delays. Moreover, by using more complex local impedances that feature an integral term, we are able to control the steady-state error between the device and the surface of the deformable object. This allows us to maximize the Z-width of the simulation, while obtaining overall stable behavior without using any added damping. The local model is always computed using the current deformable object surface, thus allowing for multipoint contact interaction, i.e., allowing multiple users to feel each other's influence on the object. The proposed solution is presented and analyzed in a multirate setting. Experimental results employing a Phantom haptic interface are presented.

KEY WORDS—haptic interfaces, deformable objects, multipoint interaction, multi-rate, stability

1. Introduction

Obtaining stable haptic interaction with deformable objects, such as those employed in force-feedback enhanced surgical simulators, is a challenging task. Deformable object algorithms can reach very high levels of computational complexity, which translates into low servo rates and computational delays. These effects, as shown by various authors in the past (Colgate and Brown 1994; Gillespie and Cutkowsky 1996; Niemeyer and Slotine 1997; Adams and Hannaford 1999), can often lead to various forms of unstable behavior.

In order to limit such effects, various approaches have been proposed in the past. Past solutions can be classified into three main groups: simplify the deformable objects algorithms in order to be less computationally heavy (Astley and Hayward 1998; Cavusoglu and Tendick 2000; Cotin, Delingette, and Ayache 2000); create a virtual coupling between haptic device and virtual tool in order to ensure passivity of the overall simulation (Colgate, Stanley, and Brown 1995; Adams and Hannaford 1999); create a model that approximates the deformable object with which haptic interaction can be computed at high servo rates (Adachi, Kumano, and Ogino 1995; Mark et al. 1996; Balaniuk 1999; James and Pai 2001; Mazzella, Montgomery, and Latombe 2002; Mahvash and Hayward 2003, 2004). Solutions that belong to the latter two classes have the advantage of being simulation method independent, i.e., can be equally used with any algorithm modeling deformable objects. We will thus focus our attention on these types of algorithms.

Algorithms belonging to the second class have the great advantage of ensuring stable haptic interaction with a wide variety of unknown virtual environments. In the most complete work on stable haptic interaction to date, Miller, Colgate, and Freeman (2000) show that haptic interaction with a very general class of virtual environments (non-passive, delayed and non-linear) can be accomplished by introducing additional damping in the system. This means that engineers who wish

The International Journal of Robotics Research
Vol. 24, No. 9, September 2005, pp. 703-715,
DOI: 10.1177/0278364905057055

to program virtual environments do not have to worry about stability. The main drawback of these algorithms is, however, that they rely on high servo rates, which often cannot be accomplished when interacting with deformable objects, and on damping, which compromises the overall transparency.

Algorithms belonging to the third class have the advantage of decoupling simulation and haptic rendering loop, thus allowing high servo rates. Their application to the case of deformable object simulations featuring low servo rates as well as computational delays however can be a very challenging task. This is due to additional stability problems that do not have to be considered in the case of rigid objects.

In this paper we present a haptic rendering algorithm that allows for stable interaction with deformable objects featuring computational delays and low servo rates, extending the results presented in Barbagli, Prattichizzo, and Salisbury (2003) to a multirate setting. The solution presented is simulation method independent and can be seen as a combination of the two types of algorithms described above. More specifically, our solution may be seen as a fast virtual coupling or local model that adapts its parameters on-line to the local impedance characteristics of the deformable object being touched. This allows for stable haptic interaction and a high level of transparency, since no additional damping is injected in the system to make it stable. Here, stability refers to the simulation loop that plays a key role to guarantee the overall stability as shown in experimental results. In other terms, this work does not focus on the passivity analysis of the haptic loop; however, extensive experimental results have shown that stable behavior is obtained by applying the solution described in this paper. A similar problem to that tackled in this paper has been recently approached by Mahvash and Hayward (2003, 2004) using a time-domain multirate passivity analysis.

It is important to note that the presented solution allows for multiple-point interaction, i.e., different users interacting with a same object can feel each other's influence. This feature cannot be obtained by applying the standard haptic rendering algorithms (Zilles and Salisbury 1995; Ruspini, Kolarov, and Khatib 1997; Balaniuk 1999; Mazzella, Montgomery, and Latombe 2002) to the case of deformable objects and sets this work apart from that of other research groups. To better explain this point, let us consider an example of two users touching a balloon filled with water in different points. Each user globally deforms the object. As a consequence of such global deformations, each user is able to feel the other user's influence. Clearly this is not possible if interaction forces are computed based on a rigid shell of the object (Ruspini, Kolarov, and Khatib 1997).

In this paper we focus our attention on admittance deformable objects,[1] whose elasto-static behavior can be computed using a single matrix inversion, which is normally precomputed off-line before the simulation starts. Moreover, we focus our attention on the case of impedance haptic devices[2] such as the Phantom.

To the best of our knowledge, the only other example of algorithms allowing multiple-point contact interaction with deformable objects were proposed in James and Pai (2001).

2. Problem Description

The state of the art on deformable object simulation normally features computational delays as well as slow servo rates ($\ll 100$ Hz). One common practice that allows for high servo rates while interacting with slowly simulated virtual environments is to decouple the haptic loop from the graphics and simulation loops. Various techniques have been proposed in the past in order to accomplish this in the case of rigid virtual objects (Adachi, Kumano, and Ogino 1995; Mark et al. 1996; Balaniuk 1999). The basic idea behind all of these solutions is to use a simple implicit function that approximates, to a good extent, a small part of the object being touched. More specifically, such intermediate representation, or local model, represents the part of the object which is closest to the current position of the haptic device. Figure 1 gives an idea of this simple concept.

Such a model can be computed in the slow simulation loop without the user noticing discontinuities, since the frequency of the human hand movement is typically lower than the simulation frequency. Haptic rendering algorithms, such as the proxy or god-object (Zilles and Salisbury 1995; Ruspini, Kolarov, and Khatib 1997), can run at high rates thanks to the simplicity of the implicit surfaces involved.

Extending the local model technique to the case of multipoint interaction with deformable object simulations featuring computational delays and slow servo rates is non-trivial. In order for different users to feel each other's influence while

1. Impedance environments accept positions and return a new interaction force. Admittance environments accept forces and return a new mesh position (Adams and Hannaford 1999).

2. Impedance haptic devices accept forces and return positions (Adams and Hannaford 1999).

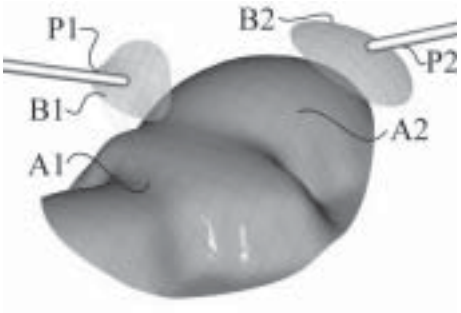


Fig. 1. Local representation of object surface.

touching a deformable object, a local model computed using the current surface representation of the object should be used. By doing so, any global deformation on the object (due to any user) can be felt by all other users. However, this complicates the overall stability of the system.

Computing a local model for rigid objects can be seen as an "open loop" problem. Given a new probe position inside the virtual environment (VE), a new local model can be computed solely based on geometric considerations. The same does not apply to deformable objects. The local model position depends on the state of the object's surface. This state, on the other hand, depends on the interaction force between user and virtual object, i.e., on the local model position. Hence a closed loop is created. Such a closed loop can become unstable, as discussed in the following, thus driving both the VE and the haptic interface in a vibrating state that completely destroys any sense of realism. In order to avoid these problems, a new local model, one not solely based on geometrical considerations, must be defined. This can be seen as an adaptive virtual coupling (Colgate, Stanley, and Brown 1995) whose rate is completely decoupled from that of the deformable object simulation.

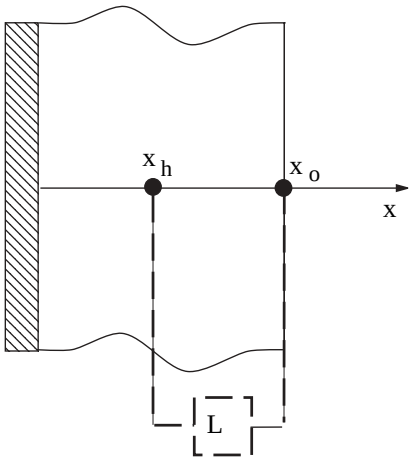


Fig. 2. The mechanical model of the haptic interface interacting with a deformable object. The probe position is represented by x_h and the proxy by x_o. The inputs of the local model L are the proxy and the probe positions while the output is the interaction force. The local model runs at a faster rate than the simulation engine of the deformable objects.

2.1. Mathematical Description of the Problem

We consider the case of a one-dimensional deformable object as shown in Figure 2 and approximate its behavior using its local mechanical impedance. This corresponds to the case of using a simple plane-based local model, which is tangent to the object being touched at the point of contact, and simulating the one-degree-of-freedom interaction with such an object along the contact normal.

Referring to Figure 2, the mechanical impedance along the x-direction can be a simple spring, a visco-elastic element or a more complex dynamic model. If the mechanical impedance is linear we can refer to it through its discrete-time transfer function. Note that a linear impedance can be thought of as a local approximation of a more involved dynamics along the same direction. In this one-dimensional example, the position of the proxy used on it is coincident with the free end of the object x_o (Figure 2).

The local model can be chosen with different dynamic behaviors that are modeled by the discrete-time transfer function $L(z)$. The system proposed in Figure 2 is multirate in nature. Every T secs a new haptic device position x_h is sampled and a corresponding interaction force equal to $f(z) = L(z)[x_h(z) - x_o(z)]$ is fed back to the user. Such a force is sampled every NT s by the simulation block returning a new deformable object surface position after NT s, i.e., after a computational period.

The time-domain representation of the closed loop mentioned above, in the case of a one-dimensional deformable object, is the one reported in Figure 3. Two closed loops exist. The haptic loop is a process that reads the new position of the haptic interface while it is being moved by the human operator, computes the new interaction force with the local model, and writes such force to the haptic interface. Such a process runs at high servo rates greater than ($1/T \gg 1$ kHz). This allows for maximization of the dynamic range of the achievable impedances of the device or Z-width as defined in Colgate and Brown (1994).

The simulation loop is a process that computes how the interaction force between haptic interface and local model influences the deformable object surface. Such a process is usually slow due to the complexity of the simulated environment ($1/NT \gg 100$ Hz). For simplicity such loops are considered synchronized. Note that if the two loops are not synchronous, it is always possible to consider an equivalent problem with synchronized rates (Khargonekar, Poolla, and Tannenbaum 1985; Francis and Georgiou 1988).

Let k (k') be the integer variable for the low (high) rate time interval. In order to model the multirate nature of this system, we use an N-fold decimator whose output $f(kN)$ of the high rate signal $f(k')$ and the N-fold expander

$$x_o(k') = \begin{cases} \hat{x}_o(kN), & \text{if } k' = kN \\ 0, & \text{otherwise} \end{cases}$$

are used as shown in Figure 3. Note that variable $\hat{x}_o$ has been used to distinguish the object surface position at low rate from x_o that is the corresponding fast rate variable obtained through the expander.

In the next section we consider the stability of the simulation as independent from the haptic loop, the one involving the

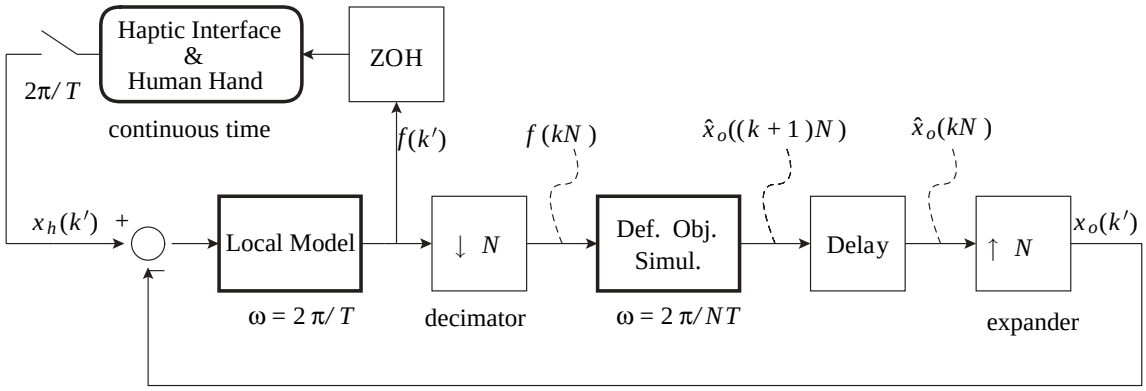


Fig. 3. All the blocks that are typically present in a deformable object simulation with haptic feedback. Two separate closed loops exist. The lower one is referred to as the simulation loop.

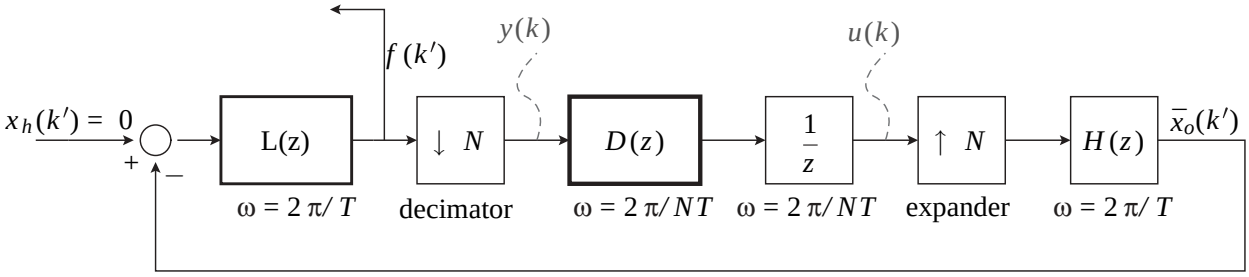


Fig. 4. Simulation loop with generic discrete time transfer functions.

human operator. This is not sufficient to ensure overall stability. However, the stability of the simulation loop plays a key role to guarantee the overall stability as shown in experimental data that supports the results presented in the following.

2.2. Simulation Loop Stability

In Figure 4 $L(z)$ is the discrete transfer function representing the local model force algorithm, and $D(z)$ is the discrete transfer function representing the deformable object surface admittance relating the deformable object position at timestep $(k+1)N$ to the force exerted on the object at k.

In Figure 4 a discrete-time zero-order hold described by

$$H(z) = 1 + z^{-1} + z^{-2} + \cdots + z^{-(N-1)} \tag{1}$$

has been added after the expander to gain a zero-order hold behavior of signals. Other types of holders can be simply taken into account in this framework.

The analysis of stability and performance of the multirate system of Figure 4 is approached in a multirate framework based on lifting techniques (Khargonekar, Poolla, and Tannenbaum 1985; Francis and Georgiou 1988). For the convenience of the reader, we briefly recall the definition of the lifted map $\widetilde{M}$ of a generic map M defined through actions on input–output time sequences:

$$\begin{aligned} M &: (u(0), u(1), \ldots, u(N), \ldots) \\ &\quad \rightarrow (y(0), y(1), \ldots, y(N), \ldots) \end{aligned} \tag{2}$$

$$\widetilde{M} : \left[\begin{pmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{pmatrix}, \begin{pmatrix} u(N) \\ u(N+1) \\ \vdots \\ u(2N-1) \end{pmatrix}, \ldots \right]$$

$$\rightarrow \left[\begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{pmatrix}, \begin{pmatrix} y(N) \\ y(N+1) \\ \vdots \\ y(2N-1) \end{pmatrix}, \ldots \right].$$

In Khargonekar, Poolla, and Tannenbaum (1985) and Francis and Georgiou (1988) it has been proved that if dynamics M is linear, time-varying and periodic with period N, it is equivalent to a linear time invariant (LTI) system in the lifted domain and that proving stability of lifted system $\widetilde{M}$ is equivalent to

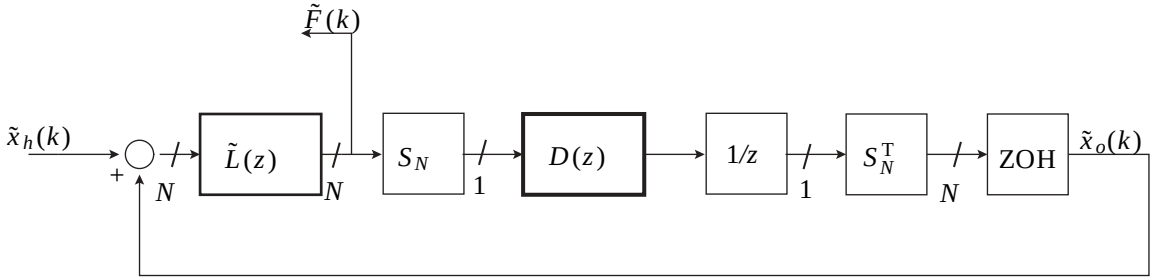


Fig. 5. LTI equivalent system. The deformable object dynamics does not change while $\widetilde{L}(z)$ is the lifted version, of the local model $L(z)$, with N inputs and N outputs.

proving the stability of M. It is worth noting that the lifted system runs at a lower rate than the non-lifted one. In particular, the time-step of dynamics $\widetilde{M}$ is N times larger than the time-step of dynamics M. Henceforth we use symbol $\sim$ for all vectors and matrices defined in the lifted domain.

The basic idea is to find a system that is equivalent to that in Figure 4 but for which only the slower servo rate (with time index k) is considered. This is done by first transforming the multirate simulation loop in an equivalent periodic system and then noticing that this is isomorphic to an LTI system with slower servo rate $(2\pi N)/T$ (Khargonekar, Poolla, and Tannenbaum 1985; Francis and Georgiou 1988). In particular, Lemmata 2.7 and 2.8 of Khargonekar, Poolla, and Tannenbaum (1985) are the key results that allow us to use the methods from LTI stability analysis applied to the lifted systems to study the multirate system in Figure 4.

It can be shown that the lifted LTI system can be represented by the block diagram reported in Figure 5. Note that the lifted LTI system equivalent to the multirate system in Figure 4 is no longer single input single output (SISO). In particular, the input and output of the lifted local model $\widetilde{L}(z)$ are N-dimensional vectors. According to the vector notation, the decimator can be modeled by the row vector

$$S_N = \overbrace{[\ 1 \quad 0 \quad \cdots \quad 0\ ]}^{N}$$

and the expander by the vector S_N^{T}. Let $[A_L, B_L, C_L, D_L]$ be a state space realization of $L(z)$ and let $\lambda(k')$ be its state vector at fast rate. The low rate (time index k) lifted dynamics $\widetilde{L}(z)$ of the local model have state space realization, represented by the quadruple $[\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}]$ and state vector $\widetilde{\lambda}(k)$,

$$\begin{aligned} \widetilde{\lambda}(k+1) &= \widetilde{A}\widetilde{\lambda}(k) + \widetilde{B}\widetilde{u}(k) \\ \widetilde{F}(k) &= \widetilde{C}\widetilde{\lambda}(k) + \widetilde{D}\widetilde{u}(k) \end{aligned} \tag{3}$$

with

$$\widetilde{\lambda}(k) = \lambda(Nk), \quad \widetilde{u}(k) = \begin{pmatrix} u(Nk) \\ u(Nk+1) \\ \vdots \\ u(Nk+N-1) \end{pmatrix},$$

$$\widetilde{F}(k) = \begin{pmatrix} F(Nk) \\ F(Nk+1) \\ \vdots \\ F(Nk+N-1) \end{pmatrix}, \tag{4}$$

and

$$\widetilde{A} = A_L^N, \quad \widetilde{B} = [A_L^{N-1}B_L A_L^{N-2}B_L \ \cdots \ B_L]$$

$$\widetilde{C} = \begin{bmatrix} C_L \\ C_L A_L \\ \vdots \\ C_L A_L^{N-1} \end{bmatrix} \tag{5}$$

$$\widetilde{D} = \begin{bmatrix} D_L & 0 & \cdots & \cdots & 0 \\ C_L B_L & D_L & 0 & \cdots & \vdots \\ C_L A_L B_L & C_L B_L & D_L & \cdots & \vdots \\ \vdots & \vdots & \vdots & & 0 \\ C_L A_L^{N-2} B_L & C_L A_L^{N-1} B_L & \cdots & \cdots & D_L \end{bmatrix}.$$

Note that in the lifted dynamics only dimensions of input and output vectors change while the state space dimension does not.

The block diagram reported in Figure 5 can be further simplified to obtain the block diagram in Figure 6 by noting the following.

- The structure of the decimator S_N implies that only the first output of $\widetilde{L}(z)$ influences the deformable object dynamics $D(z)$, i.e., $\widetilde{L} = [A_L^N,\ \widetilde{B},\ C_L,\ D_L]$.

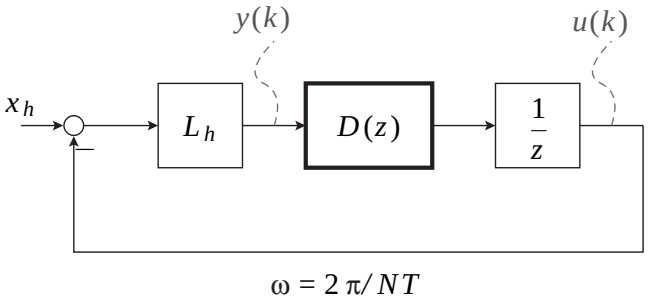


Fig. 6. SISO equivalent LTI system describing slow-rate dynamics.

- For $x_h(k') = 0$ the input to $\widetilde{L}$ is a vector with N equal elements, and thus the LTI system in Figure 5 is equivalent to that reported in Figure 6 with $L_h = [A_L^N,\ \sum_{i=0}^{N-1} A_L^i B_L,\ C_L,\ D_L]$. Note that this holds also for N-steps piecewise constant $x_h(k')$ and not only for zero values of $x_h(k')$.

In the block scheme of Figure 6, both the local model and the deformable object dynamics are SISO systems. Such a system describes the behavior of the multirate simulation loop at the slow rate and it can be used to conveniently analyze the stability of the simulation loop. In fact, using the results in Khargonekar, Poolla, and Tannenbaum (1985) it can be shown that studying stability of the multirate loop in Figure 4 is equivalent to studying the stability of the LTI system in Figure 6.

3. Proportional Local Model

Consider a one-dimensional deformable object whose local mechanical impedance is a simple spring

$$Z_o = K_o \quad \text{and} \quad D(z) = K_o^{-1}. \tag{6}$$

Let the local model be proportional

$$L(z) = K_h.$$

The simulation loop analysis reduces to the study of the closed-loop transfer function (Figure 6)

$$G(z) = \frac{1}{1 + (K_o/K_h)z} \tag{7}$$

at the lower rate $\omega = (2\pi/NT)$. The asymptotic stability is guaranteed if and only if

$$K_h < K_o. \tag{8}$$

Note that using a visco-elastic approximation of the deformable object generally results in more stable behavior than in the case of a simple elastic approximation. In such a case, in fact, we have

$$D(z) = \left(K_o + B_o\frac{z-1}{z}\right)^{-1} \tag{9}$$

where B_o is equal to the local damping of the deformable object divided by the time interval NT. The simulation loop analysis reduces to the study of the closed-loop transfer function

$$G(z) = \frac{(K_0 + B_0)z - B_o}{(K_0 + B_0)z + K_h - B_0}$$

for which the asymptotic stability is guaranteed if and only if

$$K_h < K_o + 2B_o, \tag{10}$$

which is always satisfied if stability is guaranteed in the absence of damping (eq. 8).

4. Problems With This Approach

Simply picking K_h and K_o to satisfy eq. (8) in order to obtain a stable behavior for the deformable object is not enough to ensure a realistic haptic experience. Stability is in fact only one aspect of a satisfactory system response. In the following we closely examine both transient and steady-state response for our system as well as the trade-offs between such behaviors.

4.1. Transient Response

Factors characterizing the transient response of a dynamic system, such as settling time and overshoot, play a key role in the case of realistic haptic interaction. In reality, a purely elastic deformable object will assume a new surface configuration instantaneously and without vibrating. In our case, however, this might not always be true and certain sets of parameters might lead to noticeable oscillations (see Figure 7). Considering the transfer function (7), it appears obvious that settling time (and overshoot) grows with K_h/K_o, and thus to limit such effect K_h should be chosen such that $K_o \gg K_h$ whenever possible.

4.2. Steady-state Response

Factors characterizing the steady-state response of our system are equally important in order to obtain an overall sense of realism. In general, the stiffness perceived at steady state by the user, calculated as the steady-state force divided by the deformable object surface position change, is always equal to K_o. However, due to the nature of impedance devices (Yoshikawa et al. 1995), a position error, defined as the distance between proxy and haptic device position, is always present. While such an error may go unnoticed at times due to the limitations of the human position system, certain thresholds exist above which the sense of realism is lost.

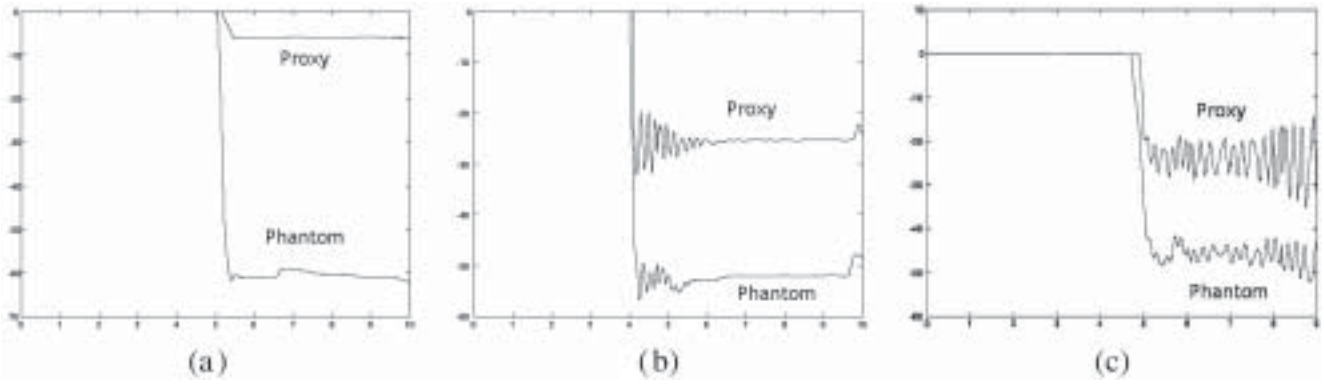


Fig. 7. Linear spring response to a step input for (a) $K_o = 0.9$ N mm^{-1} and $K_h = 0.1$ N mm^{-1}, (b) $K_o = 0.2$ N mm^{-1} and $K_h = 0.19$ N mm^{-1} and (c) $K_o = 0.2$ N mm^{-1} and $K_h = 0.22$ N mm^{-1}. The time-scale, x-axis, is in seconds while the proxy and probe positions scales, y-axis, are in millimeters.

Such error is equivalent, in our specific case, to the steady-state tracking error for a step input applied to system (7), which is equal to

$$\bar{x}_o = \frac{K_h}{K_h + K_o} x_h. \tag{11}$$

In order to limit the extent of such error to be under the human position system perception threshold, it should be $K_o \ll K_h$. This would however drive the system to instability and thus K_h should be chosen to be the closest possible to K_o.

4.3. Trade-off Between Steady State and Transient Response

A trade-off between transient and steady-state response exists, as mentioned above, and has been tested experimentally. In the case of $K_o \gg K_h$, reported in Figure 7(a) where $K_o = 9K_h$, the system response has virtually null settling time but the steady-state error is very large since the surface approximately moves only 1 cm for a physical movement of the haptic device approximately equal to 6 cm.

In the case of $K_o \simeq K_h$, reported in Figure 7(b) where $K_h = 0.19$ N mm^{-1} and $K_o = 0.2$ N mm^{-1}, the system response has a settling time of about 2 s and it is thus clearly perceived as an oscillatory effect. The steady-state distance between proxy and Phantom position is however better, being approximately equal to 25 mm for a surface deformation of 25 mm. In Figure 7(c), $K_h > K_o$ and the system is clearly unstable.

5. A Solution for Steady-state Impedance: Using an Integral Term

The main problem of purely elastic local models is that, in order to obtain a non-oscillatory force feedback, the steady-state position of the device cannot be controlled and can often be larger than desired. In order to solve this problem, a new local model, enhanced with an integral term in parallel to the proportional term used before, has been implemented. A similar idea has been proposed and implemented in Massie (1996) for the case of rigid virtual walls.

Referring to Figure 4, let us consider the Z-transform of the local model to be

$$L(z) = K_h + \frac{TI}{z-1} = K_h + \frac{TIz^{-1}}{1-z^{-1}} \tag{12}$$

where coefficient I weighs the integral action of the local model. Recall that both the proportional term K_h and the discrete integrator term $(IT/z-1)$, run at a high servo rate $(2\pi/T)$ while the deformable object simulation runs slowly $(2\pi/NT)$. The basic idea of such a scheme is that the computed force that is fed to the user and to the deformable object simulation still depends on the haptic device penetration with respect to the local model. However, the integral term eliminates such error through time thus reducing the distance between proxy and haptic device position. This is accomplished through a dual effect. On one side, the force fed back to the user by the haptic device grows, pushing the user's hand towards the proxy position; on the other side, a larger force is fed to the deformable object algorithm and thus tends to further indent the simulated surface, i.e., move proxy and hand position closer to each other.

Clearly the stability condition (8) will no longer hold true since the discrete transfer function describing the local model is no longer a simple proportional term. As for the case of the simple proportional local model, it is possible to obtain the closed-loop LTI system in Figure 6 for the stability analysis as

$$G(z) = \left(\frac{NTI}{z-1} + K_h\right) \frac{1}{zK_o} \tag{13}$$

and

$$F(z) = \frac{G(z)}{1+G(z)}. \tag{14}$$

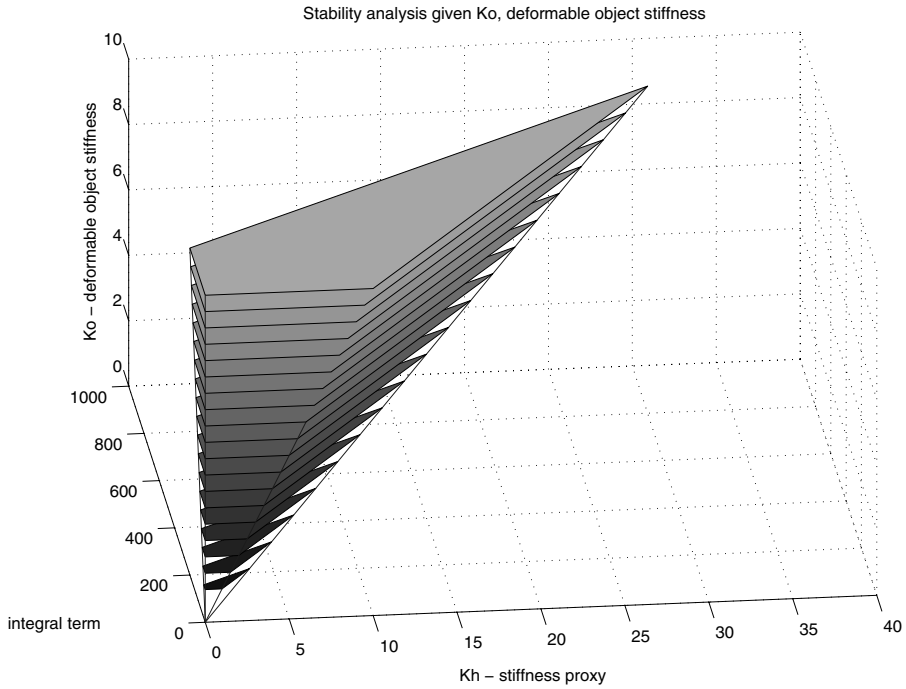


Fig. 8. The stability area for parameters K_h and *INT* given K_o of the object being touched.

In order to study how the stability of the simulation loop depends on the parameters K_o, K_h, and I, the method of Jury (1964) has been applied. In order for the simulation loop to be stable, the following conditions must hold ($K_o > 0$, $K_h > 0$, and $I > 0$)

$$\begin{cases} |INT - K_h| < K_o, \\ INT - 2K_h > -2K_o. \end{cases} \tag{15}$$

These relationships lead to a stability area for parameters K_h and *INT* for a given K_o, as those depicted in Figure 8.

Picking K_h and I inside such an area will lead to an asymptotically stable behavior for the simulation loop, unstable otherwise. It is important to note that such an area has the same shape for different values of K_o. Larger values of K_o imply areas scaled upward. A stability volume such as that depicted in Figure 8 can be defined.[3] Interestingly, the simulation loop tends to have larger slices for higher values of K_o, that is, it is easier to obtain stability when dealing with stiffer objects (i.e., the simulation loop is easier to stabilize when stiffer objects are simulated). This however does not mean that infinitely stiff walls can be rendered due to the limits on the P term of the local model imposed by the stability of the haptic loop (Colgate and Brown 1994).

3. Note that only some slices of the solid are shown to better give the idea of the volume.

The scalability of such areas is a consequence of the fact that in eq. (13) $G(z)$ can be expressed with K_h and I normalized with respect to K_o. This makes the stability of the simulation loop robust to errors due to the approximation of the deformable object dynamics as its local stiffness K_o. In fact, the values of K_h and I can always be picked based on values of K_o lower than those returned by a deformable object simulator in order to have a certain stability margin.

Note that, as for the case of proportional local model, also in this case the stability condition can only be improved by the presence of damping in the deformable object dynamics. In fact, for visco-elastic local dynamics of the deformable object (9), the stability loop analysis (Figure 6) reduces to the study of the closed-loop transfer function

$$G(z) = \frac{(K_0 + B_0)z^2 + (-K_0 - 2B_0)z + B_0}{(K_0 + B_0)z^2 + (K_h - K_0 - 2B_0)z + B + NI - K_h}$$

and according to Jury's method stability is met if and only if the following conditions hold

$$\begin{cases} |INT - K_h + B_o| < K_o + B_o, \\ INT - 2K_h > -2K_o - 4B_o. \end{cases} \tag{16}$$

Conditions (16) are always satisfied if inequalities (15) hold, i.e., if stability is guaranteed in the absence of damping in object deformations, $B_o = 0$.

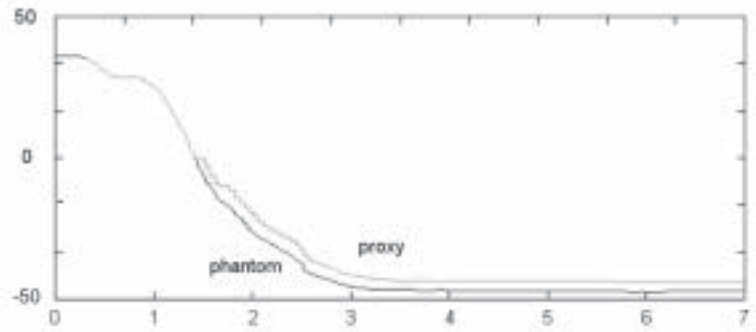


Fig. 9. Probe and proxy positions for proportional, $K_h = 0.1$ N mm^{-1}, and integral, $I = 0.003$ N mm^{-1} s^{-1} (with offset), local model. The object is modeled with a spring with elasticity $K_0 = 0.3$ N mm^{-1}.

The effect of the proportional and integral local model has been tested experimentally in the case of $K_h = 0.1$ N mm^{-1}, $I = 0.003$ N mm^{-1} s^{-1}, $K_0 = 0.3$ N mm^{-1}, $B_0 = 0.0$ N s mm^{-1}. In Figure 9 the proxy and the haptic probe position have been reported as a function of the time. The advantages of using a proportional and integral local model are clear: the steady-state error reduces to a user-defined offset introduced to avoid the limit condition where the proxy and the haptic probe position coincide. Note that such a limit condition would generate critical behavior to any collision detection algorithms. Also the settling time improves with the integral term.

5.1. Some Notes on Proportional Integral Local Models

While in theory the scheme proposed above accomplishes the proposed goals, in practice some changes must be made in order to adapt it to the real case of haptic interaction.

In order to prevent the surface of the deformable object feeling unrealistically active, the feedback force to the user should be null when the user is not in contact with the object, i.e., when the proxy and haptic device positions coincide. While this is implicit in the case of proportional local models, the same does not hold for proportional integral ones. The force due to the integral term should be eliminated. However, this should not occur instantaneously when pulling out of the object but should be a somewhat linear process. One possible approach, successfully tested in this work, is based on driving the integral term to zero linearly over an a priori fixed time interval. If such an interval is picked to be comparable to *NT*, the visual delay introduced by this solution will not be noticeable. Moreover, the object surface will not pop up to its original configuration but will do so in a more gradual way. Typical time intervals can be small multiples of *NT*. Longer periods introduce higher delays on the surface movement but lead to a nicer, more gradual force feedback effect.

The error e integrated by the I term must be chosen very carefully. Choosing such an error as the distance between proxy and haptic devices, i.e., $e = x_h - x_o$, may lead to unrealistic effects. In such a case, in fact, the I term will tend to force proxy and haptic device positions to be as close as possible, ultimately leading to an "object exiting" situation such as that described above. Feeding the I term with $e = x_h - x_o - \theta$, where θ is given by

$$\theta = \begin{cases} 0 & \text{if } e < 0, \text{ i.e., no contact} \\ \bar{\theta} & \text{if } e > 0, \text{ i.e., when there is contact} \end{cases} \tag{17}$$

and $\bar{\theta}$ is the maximum error between proxy and device positions that we decide to tolerate at steady state, will ensure "enough" distance between proxy and haptic devices.

The side effect of using an "integration threshold" is that a steady-state error will always be present. However, the extent of such error is always controllable and can be made very small. In the end, every force will be affected, at steady state, by a small offset error, which is an acceptable compromise on the overall performances of the system.

In practice, a variable threshold θ has been used. More specifically, θ can be made a function of the HI device penetration speed, i.e., $\dot{e} = \dot{x_o} - \dot{x_h}$, or/and on the current total value of the integral term. In the current implementation θ is given by

$$\theta = \begin{cases} 0 & \text{if } e < 0, \text{ i.e. no contact} \\ \bar{\theta}_1 & \text{if } e > 0 \text{ and } (\dot{x_o} - \dot{x_h}) < 0 \\ \frac{\bar{\theta}_2}{I_{tot}} & \text{if } e > 0 \text{ and } (\dot{x_o} - \dot{x_h}) > 0 \text{ and } I_{tot} < \bar{I}_{tot} \\ \bar{\theta}_2 & \text{if } e > 0 \text{ and } (\dot{x_o} - \dot{x_h}) > 0 \text{ and } I_{tot} > \bar{I}_{tot}. \end{cases} \tag{18}$$

Hence, θ is zero when there is no contact with the object; it

grows linearly when penetrating the object up to a certain limit and then is fixed to such limit; it is set to different values when entering an object and when exiting an object. The value of θ should linearly grow to its set value $\bar{\theta}_2$ while penetrating the object, because otherwise the integral term would not be active for penetrations smaller than $\bar{\theta}_2$. Moreover, it is very useful to have different values of θ depending on the penetration speed and sign. In fact, it is usually desirable to have small thresholds while penetrating an object, in order to limit the steady-state force offset, and large thresholds while exiting the object, in order to limit the cases when the penetration error drops to zero and the surface is brought back to its original configuration.

6. Local Model Algorithm

The algorithm we have used in conjunction with slowly simulated deformable objects can be summarized as in the following.

Algorithm

Step 1. The local model is calculated inside the slow simulation loop using the current surface configuration of the deformable object, as if it was static. In order to simulate the local geometry of the deformable object at the contact point we employ simple geometric primitives, such as planes and spheres, as proposed in Adachi, Kumano, and Ogino (1995) and Balaniuk (1999).

Step 2. K_h (and eventually I) are chosen inside the volume described in Figure 8. We assume that the value of K_o is returned by the algorithm used to simulate the deformable object, i.e., this paper does not deal with methods to estimate local stiffness (real or apparent) of deformable objects. It is important to note, however, that the overall stability of the local model is robust to errors on K_o if the local model's K_h and I terms are computed using a value of K_o lower than that returned by the deformable object simulation algorithm.

Step 3. The force that the local model returns to deformable object simulation, for it to compute its next configuration, is evaluated as the last force returned to the haptic device.

In the case of very large computational delays, the user still has the impression of touching a deformable object. In such cases, the local model results as an even more fundamental block since it allows for fast collision detection, given its simple shape (plane or sphere), and stable interaction.[4]

7. Experiment

The proposed proxy algorithm has been tested using a Phantom Desktop device. A simple experimental testbed has been set up for one-dimensional experiments. In particular, the undeformed object surface is at 0 mm and collision is detected when the Phantom probe is lower or equal to 0 mm. After collision is detected, the proxy algorithm starts and a displacement between the Phantom probe and the proxy is processed by the local model $L(z)$ at 1 kHz generating the force fed back to the user and to the deformable object running at 20 Hz. Plots in Figures 7 and 9 refer to the case of proportional local model and proportional and integral local model, respectively. The time-scale, x-axis, is in seconds while the proxy and probe positions scales, y-axis, are in millimeters.

As pointed out in the previous sections, Figure 7 shows how low steady-state errors can be gained to the detriment of settling time and stability. In fact, in Figure 7(b) K_o is closer to K_h and the settling time is larger than the case of Figure 7(a) where $K_h \ll K_o$ or lower than the case of Figure 7(c) where the stability condition (8) is not satisfied.

In Figure 9 plots of proxy and probe position over time have been reported for the proportional and integral local model. The integral action yields both low steady-state error and settling time. An offset equal to 4 mm has been added to the error signal integrated over time by the local model. Consequently, the steady-state error in Figure 9 is equal to 4 mm. Recall that the integral input offset is required to avoid collision detection algorithm problems.

The proposed algorithm has been also used to control the contact interaction with more complex deformable objects as those reported in Figure 10. Deformations and forces along normal directions to the undeformed object are managed according to the proposed algorithm. In particular, a simple proportional local model is used. In Figure 10(a) the proportional gain of the local model is set to 0.6 N mm^{-1} while in Figure 10(b) it is set to 0.7 N mm^{-1}. The result is that in the first case the difference between the proxy and the Phantom probe is lower than the second case, which exhibits a lower steady-state error.

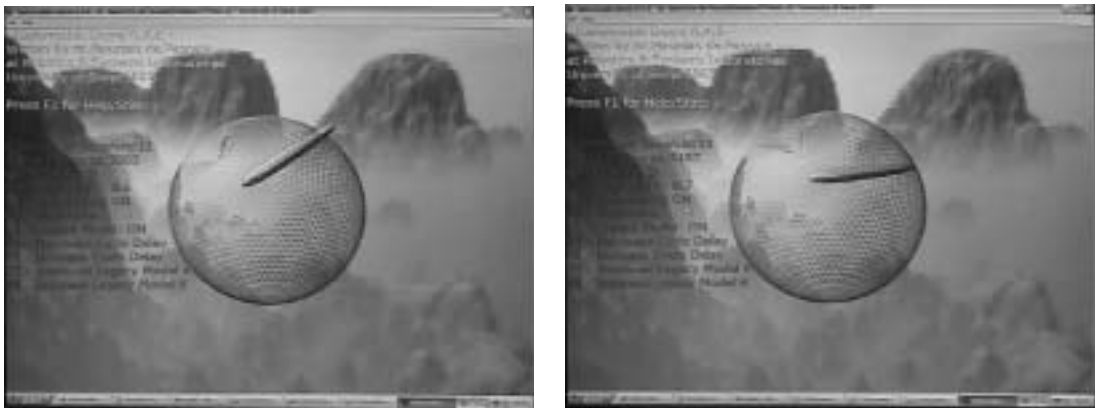
Figure 11 shows some frames of an unstable behavior video sequence. This is obtained by increasing K_h to 1.2 N mm^{-1}, which does not satisfy the stability condition (8) $K_o = 1.0$ N mm^{-1} (Extension 1).

In Figures 10 and 11 the propagation of the deformation from the contact point to the neighborhoods is obtained through suitable shape functions. The proposed proxy algorithm for deformable objects can be easily rearranged in a multipoint contact framework (Extension 2). Details on force and displacement propagation and on multipoint contact can be found in de Pascale et al. (2004).

8. Conclusions

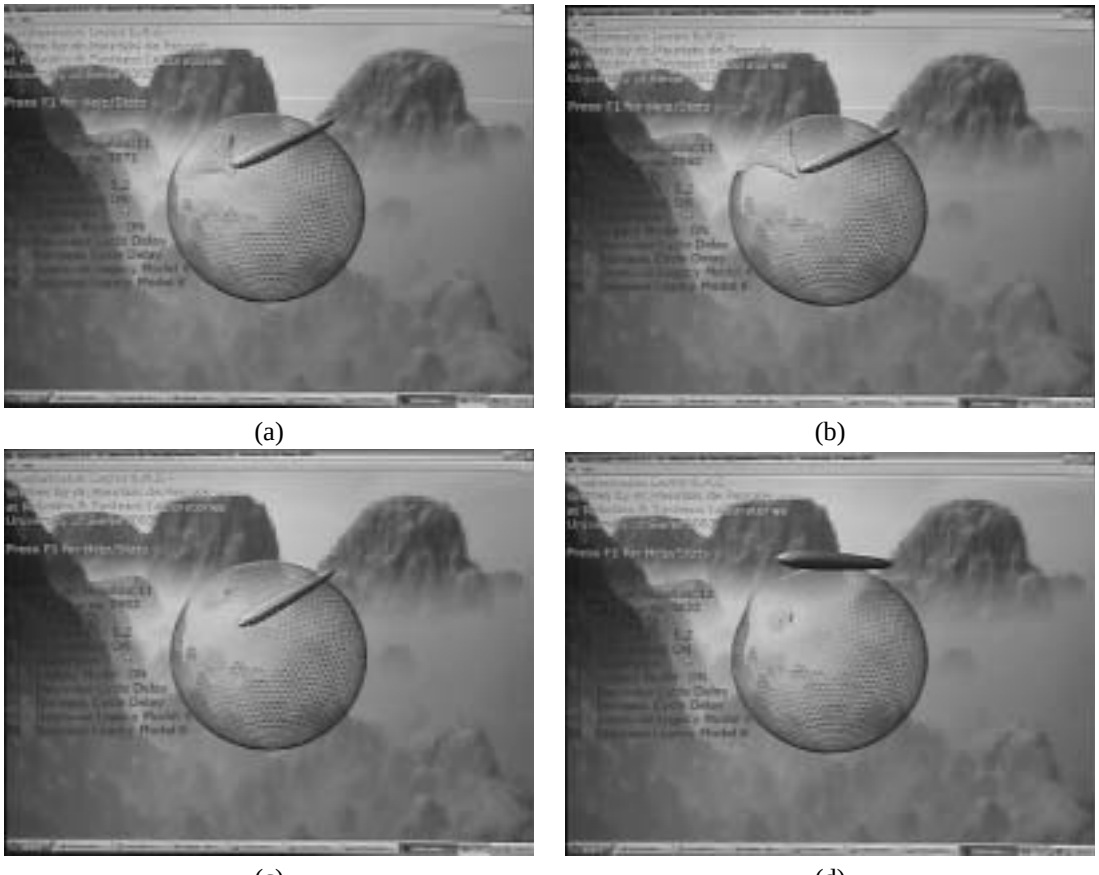
In this paper we have described new techniques allowing users to haptically interact with a deformable object simulation featuring computational delays and low servo rates. In order to have stable interaction, we propose an adaptive local model (or

4. Using a normal virtual coupling with a delay of 10 ms, for instance, would result in the user penetrating the virtual object too much and obtaining an unrealistic force response.



(a) (b)

Fig. 10. A deformable object represented by triangular meshes. The local model gain K_h is set to 0.7 N mm^{-1} in (a) and to 0.6 N mm^{-1} in (b).



(a) (b)

(c) (d)

Fig. 11. Unstable sequence of haptic interaction with deformable objects. In this case instability is due to the high level of the local model proportional gain, $K_h = 1.2$ N mm^{-1}.

virtual coupling) approach that allows for fast collision detection, large Z-widths without any need for additional damping in the system. The method is suitable to be embedded in multipoint contact interaction algorithms. The stability analysis has been investigated for the simulation loop in a multirate framework. In particular, the lifting technique has been used to study system stability of multirate systems. The main result is that when the local model is a static gain, then in order to guarantee stability the local model must be less stiff than the object at the contact point as shown in eq. (8). Proportional integral based local models have been proposed to avoid steady-state errors and the stability condition of the simulation loop here involves also the computational delay NT as shown in eq. (16). Although the main results deal with a single degree of freedom and with simple deformable object dynamics, the algorithm proposed in Section 6 shows a possible way to extend our algorithm to more involved object geometry and deformable dynamics exploiting previous results existing in the literature.

This work deals with admittance environments in the sense that the deformable object accept forces and return positions. Work is in progress to extend the paper results to impedance environments. This work does not take into account the passivity analysis of the haptic loop including the human operator and does not considered the perception studies to investigate the effect of the local model on the realism of the tactile perception. Work is in progress to extend the result to the passivity analysis and the perception study. However, extensive experimental results have shown that stable and realistic behaviors are obtained by applying the algorithm described in this paper.

Finally, it is important to note that this work does not propose any new advancement in how to geometrically approximate the surface of a deformable object, but simply deals with how to choose the local stiffness of local models based on pre-existing techniques in order to obtain stable interaction. Work is in progress to apply the proposed techniques to the proxy algorithm previously proposed by Ruspini and Khatib. This would allow us to avoid the step of computing the local model, although it still would require us to perform collision detection with a deformable object, which is normally a computationally heavy process.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Stable and unstable behaviours of contact interaction
2	Video	Two-point interaction (feeling each-other influence)

Acknowledgments

The authors wish to thank the reviewers for their valuable comments, which have improved the quality of the paper, and Maurizio de Pascale for the development of the deformable object demonstration. This research is supported by University of Siena (progetto di ricerca PAR-2004) and by Fondazione Monte dei Paschi, Italy (grant No. 21624-308-1).

References

Adachi, Y., Kumano, T., and Ogino, K. 1995. Intermediate representation for stiff virtual objects. *IEEE Virtual Reality Annual International Symposium*, Research Triangle Park, NC, March, pp. 203–210.

Adams, R. and Hannaford, B. 1999. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation* 15(3):465–474.

Astley, O. and Hayward, V. 1998. Multirate haptic simulation achieved by coupling finite element meshes through Norton equivalents. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA98)*, Leuven, Belgium, pp. 989–994.

Balaniuk, R. 1999. Using fast local modeling to buffer haptic data. *Proceedings of Fourth Phantom Users Group Workshop (PUG99)*.

Barbagli, F., Prattichizzo, D., and Salisbury, J. 2003. Dynamic local models for stable multicontact haptic interaction with deformable objects. *Haptics Symposium 2003*, Los Angeles, CA, pp. 109–116.

Brady, K. and Tarn, T. 1998. Internet-based remote teleoperation. *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May.

Cavusoglu, M. and Tendick, F. 2000. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 24–28, pp. 2458–2465.

Colgate, E. and Brown, J. 1994. Factors affecting the z-width of a haptic display. *Proceedings of the IEEE International Conference on Robotics and Automation*, Los Alamitos, CA, pp. 3205–3210.

Colgate, E., Stanley, M., and Brown, J. 1995. Issues in the haptic display of tool use. *Proceedings of the International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, August.

Cotin, S., Delingette, H., and Ayache, N. 2000. A hybrid elastic model allowing real-time cutting, deformation and force feedback for surgery training and simulation. *The Visual Computer* 16(8):437–452.

de Pascale, M., de Pascale, G., Prattichizzo, D., and Barbagli, F. 2004. A new method for haptic GPU rendering of deformable objects. *Proceedings of EuroHaptics 2004*, Munich, Germany, June.

Francis, B. and Georgiou, T. T. 1988. Stability theory for linear time-invariant plants with periodic digital controllers. *IEEE Transactions on Automatic Control* 33(9):820–832.

Gillespie, B. and Cutkosky, M. 1996. Rendering of the virtual wall. *Proceedings of the ASME International Mechanical Engineering Conference and Exposition*, Atlanta, GA, November, Vol. 58, pp. 397–406.

James, D. and Pai, D. 2001. A unified treatment of elastostatic contact simulation for real time haptics. *haptics-e, The Electronic Journal of Haptics Research* 2(1).

Jury, E. I. 1964. *Theory and Applications of the z-Transform Method*, Wiley, New York.

Khargonekar, P. P., Poolla, K., and Tannenbaum, A. 1985. Robust control of linear time-invariant plants using periodic compensation. *IEEE Transactions on Automatic Control* 30(11):1088–1096.

Mahvash, M. and Hayward, V. 2003. Passivity-based high-fidelity haptic rendering of contact. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*, Taipei, Taiwan, September 14–19.

Mahvash, M. and Hayward, V. 2004. High fidelity haptic synthesis of contact with deformable bodies. *IEEE Computer Graphics and Applications*, special issue on haptic rendering, 48–55.

Mark, W., Randolph, S., Finch, M., Verth, J. V., and Taylor, R. 1996. Adding force feedback to graphics systems: issues and solutions. *Computer Graphics: Proceedings of SIGGRAPH'96*, August, pp. 447–452.

Massie, T. 1996. Taking the mush out of haptics with infinitely stiff walls. *Proceedings of the First Phantom Users Group Workshop*, Dedham, MA, September.

Mazzella, F., Montgomery, K., and Latombe, J. 2002. The force grid: a buffer structure for haptic interaction with virtual elastic objects. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA02)*, Seoul, Korea, May 21–26.

Miller, B., Colgate, J., and Freeman, R. A. 2000. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation* 16(6):712–719.

Niemeyer, G. and Slotine, J. 1997. Designing force reflecting teleoperators with large time delays to appear as virtual tools. *Proceedings of the IEEE Conference on Robotics and Automation*, Albuquerque, NM, pp. 2212–2218.

Ruspini, D., Kolarov, K., and Khatib, O. 1997. The haptic display of complex graphical environments. *Computer Graphics* 31:345–352.

Yoshikawa, T., Yokokohji, Y., Matsumoto, T., and Zheng, X. 1995. Display of feel for the manipulation of dynamic virtual objects. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control* 117(4):554–558.

Zilles, C. and Salisbury, J. 1995. A constraint based god-object method for haptic display. *Proceedings of the IEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, Pittsburgh, PA, Vol. 3, pp. 146–151.