

Local model of interaction for haptic manipulation of rigid virtual worlds

Daniela Constantinescu, Septimiu E. Salcudean, Elizabeth A. Croft

Abstract—This paper proposes a local model of rigid body interaction that provides users with convincing kinesthetic feedback while they manipulate a virtual tool within a rigid multibody virtual world. The virtual tool can be a rigid object or a linkage. The local model can interface a haptic device to any virtual environment simulation that provides the contact position, the contact normal direction, and the penetration depth of the virtual tool into the virtual world at the virtual tool contacts. The local model includes a proxy of the virtual tool that embeds an approximation of its geometry and of its dynamics, and that eliminates force discontinuities at model updates via proxy deformation. In addition, the model includes active and predicted virtual tool contacts. Predicted contacts are used to maintain force continuity and to better constrain users to tight virtual spaces. Experiments performed within a planar virtual world demonstrate that, compared to prior local models of rigid body interaction, the proposed model enables users to feel stiffer contacts and improves users' perception of free space in locally cluttered virtual environments.

Index Terms—haptic feedback, local model of rigid body interaction, dynamic proxy, rigid multibody virtual environment.

I. INTRODUCTION

For more than a decade, virtual reality applications have interested various prospective users, from engineers who would feel forces acting on virtual prototypes to doctors who would learn brain surgery on computer models rather than on animal or human patients. Due to the wealth of potential advantages of virtual force interactions, much research has focused on the development of control [1], [2], and simulation [3], [4], [5], [6], [7], [8], [9] algorithms for haptic manipulation of rigid virtual worlds. However, few applications are presently in use [6], [8] and consumer grade haptics still elude us.

A prerequisite for the more common use of haptics is a method that allows the coupling of a haptic device to an arbitrary virtual environment and provides stable and realistic forces to users. Such a method is challenging to develop

because it must reconcile conflicting requirements from our sense of touch, the device controller, and the simulation. To perceive virtual objects as realistic representations of real objects, our sense of touch requires forces that suitably approximate physical phenomena and are provided at frequencies of the order of hundreds of Hz. To create an adequate perception of rigidity and to maintain the stability of the interaction, the device controller requires setpoints for the force control loop at fixed frequencies of the same order of magnitude. However, typical interactive, physically based, rigid multibody simulations provide forces at variable rates of the order of tens of Hz [10]. To address the need for physically motivated forces at fixed high frequencies, two approaches have been pursued.

In a first approach, researchers have improved the real time performance of the virtual environment via new collision detection algorithms [7], [8], [9]. Those algorithms alleviate the collision detection bottleneck characteristic of simulations of complex virtual worlds. Simulations that employ such algorithms achieve haptic performance for complex contact geometries, but the stability of the interaction is not guaranteed [7]. Furthermore, force feedback within such fast virtual worlds is still not possible if the haptics server differs from the virtual environment simulation server. This could occur, for example, when the haptic device is connected to the simulation engine through the internet.

In another approach, researchers have decoupled the force control loop from the virtual environment simulation. The decoupling relieves the haptic rate demand on the simulation, thus enabling users to manipulate virtual worlds of increased complexity. In addition, the haptics and the simulation engines may be different. However, users perceive realistic interactions only if the forces provided to the control loop during one simulation step suitably approximate the forces acting on the virtual object that they manipulate in the virtual environment (hereafter called the virtual tool). Therefore, the decoupling of the force control loop from the simulation requires a local model of the interaction. This local model must run at the fixed haptic rate and must include virtual world features that are salient to the interaction over one simulation step.

The earliest local model has been proposed for point interaction within virtual environments [11]. That model comprises the position and the outward normal of the active contact. Hence, it is a geometric local model of the virtual world that shifts the computational delay of the simulation from delay in updating forces to delay in updating the local geometry. As a result of faster force computations, users feel stiffer contacts. However, force discontinuities arise at model updates due to geometry discontinuities, such as large changes in the contact

Manuscript received September 16, 2004; revised April.

Regular paper

Corresponding author: D. Constantinescu is with the Department of Mechanical Engineering, University of Victoria, Canada. Mailing address: University of Victoria, Department of Mechanical Engineering, Engineering Office Wing, Room 548, 3800 Finnerty Road, Victoria, B.C., Canada V8W 2Y2. Fax: (250)721-6051. Email: danielac@me.uvic.ca.

S. E. Salcudean is with the Robotics and Control Laboratory, University of British Columbia, Canada. Mailing address: Department of Electrical and Computer Engineering, 2356 Main Mall, UBC, Vancouver, B.C., Canada V6T 1Z4. Fax: (604) 822-5949. Email: tims@ece.ubc.ca.

E. A. Croft is with the Industrial Automation Laboratory, University of British Columbia, Canada. Mailing address: Department of Mechanical Engineering, 2054-2324 Main Mall, UBC, Vancouver, B.C., Canada V6T 1Z4. Fax: (604) 822-2403. Email: ecroft@mech.ubc.ca.

position or the contact normal. Force discontinuities introduce perceptual artifacts and may destabilize the interaction. In prior research, force discontinuities at model updates have been diminished: (i) by allowing the active contact to move in the local model with velocity equal to its velocity in the virtual environment at the moment of the update [10]; (ii) by updating the local contact position progressively, through linear interpolation between the old and the new positions reported by the simulation [12]; and (iii) by alleviating geometry discontinuities via local models that comprise both active and predicted geometry [13], [14]. Existing local models with predicted geometry consider only point interaction with the virtual environment. Furthermore, they compute expected contacts with the virtual object touched by the user based on information about the connectivity of the geometric features (i.e., faces, edges, and vertices) of this object. Thus, the model in [13] comprises all object faces neighboring the active contact, while the model in [14] includes all object faces within the user's reach over the next simulation step (selected using the user's hand velocity in addition to connectivity information). However, existing simulations embed information about the connectivity of the geometric features of the virtual objects in various data structures. Local models using prediction algorithms that require such information impose restrictions on the virtual environments that they can interface to the haptic device.

A local model of rigid body interaction (called intermediate representation) has been proposed in [15]. It includes all active contacts of the virtual tool. Because instability arises when the virtual tool is tightly constrained by the virtual environment (for example, during peg-in-hole manipulations), the intermediate representation has been used to restrict only the translation of the haptic device. Device rotation has been constrained to the virtual tool rotation through a virtual coupler [1], [2] (i.e., a proportional derivative - PD - controller that acts like a rotational spring-damper between them).

To meet the need of haptic devices for physically motivated forces provided at high fixed frequencies, this work proposes to decouple the force control loop from the simulation. As demonstrated via experiments in Section V, it develops the first reported local model of rigid body interaction that can be used to constrain both the translation and the rotation of the haptic device during user manipulations of rigid virtual tools within multibody virtual worlds. Four features distinguish the proposed model from existing models:

- The model selects local geometry without requiring information about the connectivity of the geometric features of the virtual objects to be available in the virtual environment. Therefore, it can interface a haptic device to an interactive simulation regardless of the algorithms and the data structures employed to generate the virtual world. This is shown in Section V via interfacing a haptic device to a multibody simulation generated using a commercial physics engine through the proposed model.
- The model includes a dynamic representation of the virtual tool (hereafter called dynamic proxy). The dynamic proxy allows physically-motivated virtual tool interactions to be computed locally regardless of whether users

manipulate rigid objects or linkages. Moreover, it enables the haptic controller to transmit wrenches (i.e., forces and torques) between the device and the dynamic proxy and to coordinate body positions (i.e., position and orientation) between them. The experiments in Section V-C show that, compared to directly coupling an impedance device to an admittance simulation [15], this controller renders stiffer contacts to users.

- The model maintains force continuity at model updates via local proxy deformation. Proxy deformation converts local geometry discontinuities caused by the virtual environment delay into discontinuities in proxy geometry. At the same time, it maintains the continuity of the proxy motion and of the penetration of the local contacts. As illustrated via simulations in Sections III-B and III-C and via experiment in Section V-A, proxy deformation eliminates force spikes at model updates and, thus, enables stable haptic interaction within virtual environments with interactive performance (environments with average frequencies of 50Hz are considered in this work).
- The model incorporates expected contacts between the virtual tool and objects not adjacent to it. Expected contacts are included in the local model via virtual environment geometry within ϵ distance from the virtual tool (hereafter called ϵ -active geometry). As the simulation in Section III-C and the experiments in Section V-B show, ϵ -active geometry improves users' perception of tight virtual spaces when users operate the virtual tool through small clearances.

The proposed local model is presented in the following sections, starting with a synopsis of haptic interaction within rigid virtual worlds using this model in Section II. The local geometry is presented in Section III. The local dynamics, including the local contact model that they are based on, are overviewed in Section IV. Experimental results obtained by interfacing a planar haptic device to a virtual environment through the proposed local model of interaction are discussed in Section V. A discussion of the advantages and limitations of the proposed local model and of directions for future work concludes the paper in Section VI.

II. SYNOPSIS OF HAPTIC MANIPULATION OF RIGID VIRTUAL TOOLS USING THE LOCAL MODEL OF INTERACTION

Haptic manipulation of rigid virtual tools through the local model of interaction proposed in this paper is schematically depicted in Figure 1. As shown in this figure, the interaction between the virtual tool and the virtual world is approximated in the local model through the interaction between the virtual tool and neighboring virtual objects. The quality of the approximation is maintained by updating the local model at each step of the simulation. The virtual interactions are applied to users through a controller that coordinates both wrenches and body positions between the haptic device and the local model of interaction.

Figure 1 shows that the present local model comprises active and ϵ -active constraints imposed on the motion of

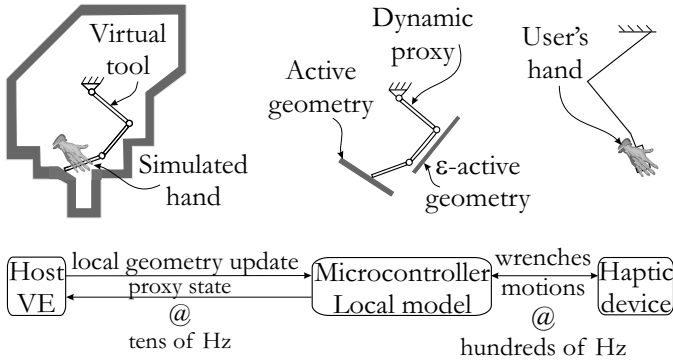


Fig. 1. Communication between the virtual environment (VE), the local model of interaction, and the haptic device.

the virtual tool by the virtual environment, as well as a dynamic proxy of the virtual tool. The ϵ -active constraints represent virtual environment geometry within ϵ distance from the virtual tool. They add contact prediction capabilities to the local model without requiring the virtual environment to implement particular data structures (like kinetic data structures [16], [17], [18] or velocity-aligned discrete oriented polytopes [19]) and algorithms (like those proposed in [20], [21], [22], [23], [24]). By taking advantage of motion coherence and of geometric locality, such dynamic collision detection data structures and algorithms significantly improve the simulation performance while guaranteeing that no collisions are missed. Therefore, many simulation packages implement some dynamic collision detection. However, the present local model aims to interface a haptic device to any commercially available simulation engine. Therefore, it predicts virtual tool contacts using ϵ -active geometry. As described in Section III-C, ϵ -active geometry imposes no restrictions on the simulation.

The dynamic proxy differs from proxies used in prior haptics research [3], [25] in two ways: (i) it provides a representation of the virtual tool rather than of the haptic device; and (ii) it approximates the geometry and the dynamics of the virtual tool rather than only the position of its centre of mass (COM). These differences arise from the dissimilar purposes served by earlier proxies (in the haptic simulation) and by the dynamic proxy (in the present local model of interaction). Earlier proxies encode the position of the user in the simulation that is closest to the haptic device and is compatible with the geometry of the rigid virtual objects. Those proxies accomplish two goals: (i) they prevent users from tunneling through thin virtual walls when the haptic device moves past such walls in a single simulation step; and (ii) they identify a unique penetration distance and a unique direction of user penetration into the virtual environment regardless of the contact geometry, thus enabling users to perceive constant environment stiffness. During rigid body interaction within rigid multibody virtual worlds, the geometry of the contact configuration and the mass distribution of the virtual tool influence the realism of the wrenches applied to users. Therefore, the dynamic proxy is designed to provide suitable approximations both of the geometry of the virtual tool contacts (described in Section III-B) and of the virtual tool dynamics (described in Section IV),

and, thus, to enable the proposed local model to compute convincing virtual interactions.

A further benefit of the dynamic proxy is that it allows the four channel teleoperation controller developed in [26] to be used for coordinating both wrenches and body positions between the local model of interaction and the haptic device. This controller feeds the user-applied wrench through to the proxy and the proxy interactions back to the user's hand via its two force coordination channels. Its position coordination channels act as one translational and one rotational spring-damper pairs between the proxy and the device. Compared to the virtual coupler [27], [2] (which coordinates only positions between the haptic device and the virtual environment), the four channel controller applies to users wrenches representing collisions and dry friction and, thus, increases the realism of the virtual interaction. Compared to the direct coupling of the haptic device to the virtual environment [15] (which only applies the hand wrenches to the proxy and the virtual interactions to the user's hand), the teleoperation controller adds damping through its position coordination channels. The experiments in Section V demonstrate that this additional damping allows stiffer contacts to be rendered to users. Hence, the dynamic proxy facilitates the use of a haptic controller that improves the compromise between transparency and stability during user manipulations of rigid virtual tools within multibody virtual worlds.

III. LOCAL GEOMETRY

The physically-motivated forces required for a convincing haptic experience arise at the contacts between the virtual tool and other virtual objects, collectively referred to as the virtual environment. Hence, the representation of contact and the geometry of the virtual world are important for realistic force feedback. Complete geometry is desirable for accurate computation of interactions, since it allows all contact transitions of the virtual tool to be resolved locally. Nevertheless, complete geometry cannot be used in the local model due to the complexity and the variability across simulation packages of both the data structures and the collision detection algorithms associated with the complete geometry.

Partial virtual environment geometry may result in perceptual artifacts or unstable interaction due to undesirable force discontinuities at model updates. The selection of salient geometry to be used in the local model, and the choice of a method for transitioning between the old and the new local models, are critical for a stable and convincing virtual kinesthetic experience. The techniques employed for selecting appropriate local geometry and for transitioning between successive local models are presented in the following section.

A. Active geometry

Virtual environment geometry in direct contact with the virtual tool at simulation updates, called active geometry, is of primary importance for realistic user manipulations of multibody virtual worlds. Therefore, the active geometry is initially chosen to approximate the virtual environment in the proposed local model of interaction.

To enable the coupling of a haptic device to an arbitrary virtual world, the active geometry is restricted to information generally available in physically-based virtual environments. Typical simulation packages represent rigid bodies as collections of convex polyhedra [8] and compute basic (vertex-face and edge-edge) contacts [28] between pairs of these polyhedra. Furthermore, to solve the forward dynamics of the virtual world, they approximate rigid body contact via a finite number of basic contacts¹ [29], [30]. For each basic contact, they provide a contact point $\tilde{\mathbf{c}}_{p,VE}$, a contact normal direction \mathbf{n} , and a penetration depth. The penetration depth is equal and opposite to the separation distance s between the polyhedra features involved in the basic contact; i.e., it is positive when the virtual tool and the virtual environment overlap. The *active geometry* consists of all basic contacts of the virtual tool with the virtual environment.

Note that the active geometry provides a discrete approximation of the contact geometry of the virtual tool. This approximation suitably embeds the motion constraints imposed on the virtual tool by the virtual environment at all but the contacts with indeterminate contact normals, for example at vertex-edge contacts. At such contacts, the contact normal reported by the simulation may change frequently and abruptly due to the finite precision of the computations and the lack of memory of existing collision detection algorithms for multi-body environments. For example, vertex-edge contacts may be reported as vertex-face contacts with either face neighboring the edge. To alleviate the perceptual artifacts due to contacts with indeterminate normals, the local model of interaction uses the ϵ -active geometry, as described in Section III-C.

In the local model of interaction, the active geometry is encapsulated in the local proxy contacts. A *local proxy contact* is defined through the identifiers of the two contacting objects and its geometric, kinematic, and dynamic properties. Typically, the identifiers of the contacting objects are the addresses of their data structures in the simulation. The geometric properties of a local proxy contact embed the geometric information provided by typical simulation packages for each basic contact of the virtual tool (see Figure 2). These properties consist of: the position of the local constraint $\tilde{\mathbf{c}}_p$, the normal direction of the local constraint \mathbf{n} , and the local contact point $\tilde{\mathbf{p}}$ (see Figure 2(d)). The position of the local constraint $\tilde{\mathbf{c}}_p$ is the position of the face point closest to the vertex when the virtual tool is in vertex-face contact with the virtual environment (Figure 2(a)); it is the position of the vertex when the virtual tool is in face-vertex contact with the virtual environment (Figure 2(b)); and, it is the position of the point on the environment edge that is closest to the virtual tool edge when the virtual tool is in edge-edge contact with the virtual environment (Figure 2(c)). The normal direction of the local constraint \mathbf{n} is the contact normal direction supplied by the simulation. Furthermore, the local contact point $\tilde{\mathbf{p}}$ is the vertex itself when the virtual tool is in vertex-face contact with the

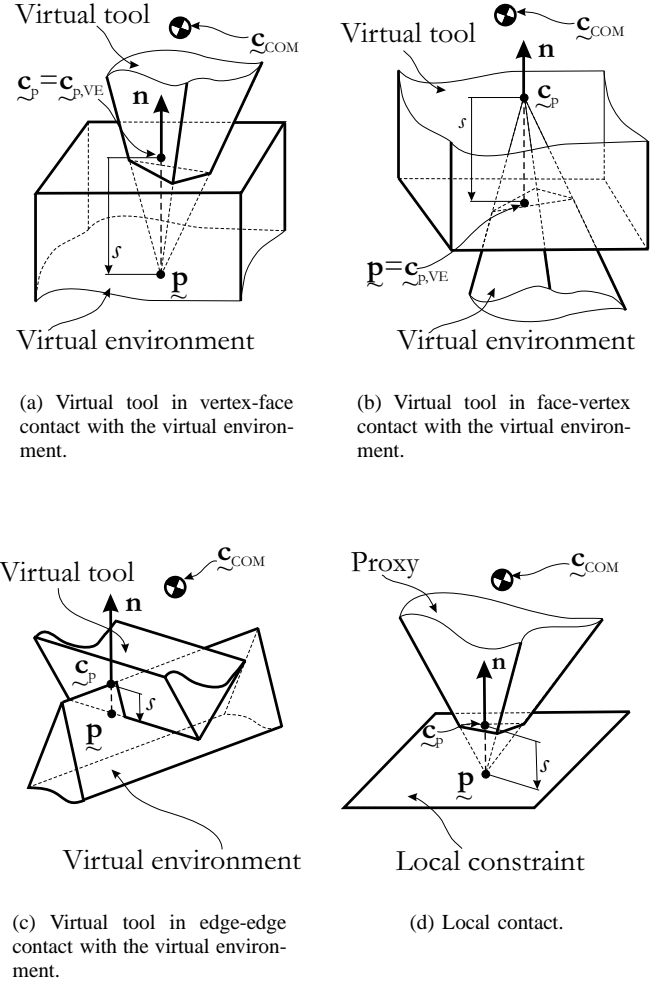


Fig. 2. Geometric information provided by typical simulation packages for each basic contact of the virtual tool (Figures 2(a), 2(b), and 2(c)) and its representation in the local model of interaction (Figure 2(d)).

virtual environment (Figure 2(a)); it is the face point closest to the vertex when the virtual tool is in face-vertex contact with the virtual environment (Figure 2(b)); and, it is the point on the virtual tool edge closest to the environment edge when the virtual tool is in edge-edge contact with the virtual environment (Figure 2(c)). Note that, constraints are considered to extend infinitely in the local model. Moreover, a proxy-centric view is adopted in describing the active geometry: constraint normals are directed from the virtual environment to the virtual tool and the contact point is taken on the proxy.

Both the position of the local constraint $\tilde{\mathbf{c}}_p$ and its normal direction \mathbf{n} are provided in world coordinates by the virtual environment simulation. This is similar to the intermediate representation of virtual environment geometry used in [15] and is a direct extension to rigid body interaction of the intermediate representation in [11]. The local contact point $\tilde{\mathbf{p}}$ is computed in proxy coordinates by the local model of interaction:

$$\tilde{\mathbf{p}} = \mathbf{R} \left(\tilde{\mathbf{c}}_p + s\mathbf{n} - \tilde{\mathbf{c}}_{COM} \right). \quad (1)$$

¹For example, typical simulation packages represent a face-face contact via several vertex-face contacts, an edge-face contact via two vertex-face contacts, and a vertex-edge contact via two vertex-face contacts [29].

In Equation (1), \mathbf{R} is the rotation from the world to the proxy coordinates and $\tilde{\mathbf{c}}_{COM}$ is the position of the centre of mass (COM) of the proxy.

The kinematic properties of a local proxy contact are provided by the local constraint velocity v_{constr} , which is the projection along the constraint normal \mathbf{n} of the velocity of the virtual environment point in contact with the virtual tool $\mathbf{v}_{\tilde{\mathbf{c}}_p}$:

$$v_{constr} = \mathbf{n}^T \mathbf{v}_{\tilde{\mathbf{c}}_p}. \quad (2)$$

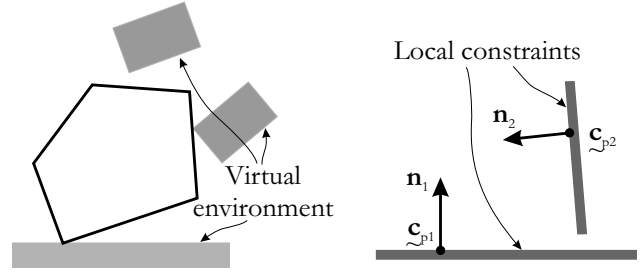
The local constraint velocity is used to predict the constraint position at the next simulation update, thus diminishing local geometry discontinuities at model updates. The dynamic properties of a local proxy contact consist of the contact stiffness k_{lc} , the contact damping b_{lc} , the coefficient of restitution e , and the coefficients of static and kinetic friction, μ_s and μ_k , respectively. In the proposed local model of interaction, all local contacts represent contacts between rigid bodies and have the same stiffness and damping. Furthermore, nonincreasing kinetic energy of the proxy during collisions is ensured by assuming that all local contacts have the same coefficient of restitution (as explained in Section IV).

The local proxy contacts are the only geometry that the local model is aware of. Hence, only partial virtual environment geometry is available locally. This is illustrated in Figure 3, where the local representation of an example virtual world geometry is depicted. Partial geometry makes the proposed approach compatible with simulation packages regardless of the data structures that they use for representing the virtual objects. Moreover, it simplifies local collision detection, which becomes an iteration through all local contacts in order to compute separation distances according to:

$$s = \mathbf{n}^T \left(\tilde{\mathbf{c}}_{COM} + \mathbf{R}^{-1} \tilde{\mathbf{p}} - \tilde{\mathbf{c}}_p \right). \quad (3)$$

In Equation (3), s is the separation between the proxy and the local constraint at the contact of interest and \mathbf{R}^{-1} is the rotation from proxy to world coordinates. In other words, local collision detection only checks whether a contact reported by the virtual environment is made or broken between updates. Transitions of the proxy to contact states involving other features of the virtual tool and the virtual environment than those reported by the simulation are not possible between updates. This is a direct consequence of the fact that no geometric connectivity information is used in the local model.

The active geometry shifts the computational delay of the simulation from delay in computing interaction forces to delay in updating the local geometry. As a result of faster force computations, the users' haptic experience is improved in two ways: (i) they can manipulate much stiffer objects, due to the high control rate that can be achieved; and (ii), they can feel physical phenomena that rely on fast force transitions, such as collisions and stick-slip friction. However, at model updates, undesirable discontinuities may arise in the locally computed forces that may destabilize the interaction. Two techniques are proposed to avoid such discontinuities: local



(a) Example contact geometry in the virtual environment simulation.

(b) Local representation of the virtual environment geometry shown in Figure 3(a).

Fig. 3. Partial virtual environment geometry available in the proposed local model of interaction.

proxy deformation and prediction of the local constraints. As described in the following Sections III-B and III-C, local proxy deformation smoothes interaction forces at simulation updates via maintaining the penetration of the local contacts continuous, while constraint prediction alleviates the effect of the computational delay of the virtual environment simulation via sending constraints to the local model before they become active.

B. Local proxy deformation

An update of the local model may result in: (i) new local contacts with significant penetration (this may happen when the user quickly moves the virtual tool into a constraint); and (ii) existing local contacts with significant discontinuities in the penetration (this may happen when the virtual environment is generated using a penalty-based simulation, because the proxy interactions are approximations of the virtual tool interactions and the proxy penetration into the local constraints may differ from the virtual tool penetration into the virtual environment). Discontinuities in the local contact penetration lead to discontinuities in the setpoint of the force control loop and may produce unacceptable force spikes or may destabilize the interaction. In the present local model, this difficulty is addressed by maintaining the penetration of the local contacts continuous through proxy deformation. When the virtual environment sends a new contact to the local model, the proxy is deformed such that it only touches the new constraint, i.e., the new local contact point is computed by (see Figure 4):

$$\tilde{\mathbf{p}} = \mathbf{R} \left(\tilde{\mathbf{c}}_p - \tilde{\mathbf{c}}_{COM} \right). \quad (4)$$

When the virtual environment sends an existing contact to the local model, the local contact point is re-computed by:

$$\tilde{\mathbf{p}} = \mathbf{R} \left(\tilde{\mathbf{c}}_p + s\mathbf{n} - \tilde{\mathbf{c}}_{COM} \right). \quad (5)$$

The proxy is expanded back towards the actual geometry of the virtual tool whenever a local contact is lost.

To maintain continuous penetration of the local contacts, proxy deformation introduces suitable adjustments in proxy

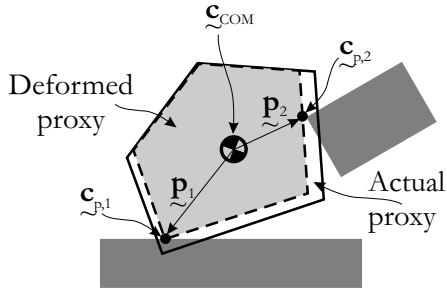


Fig. 4. Local deformation of the proxy due to violation of new constraints.

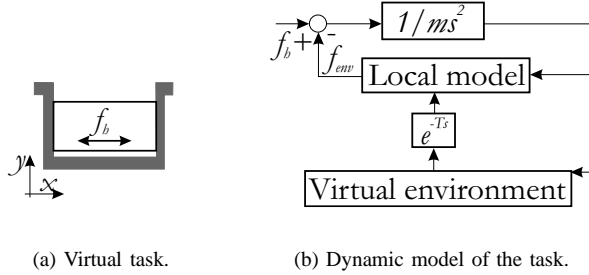
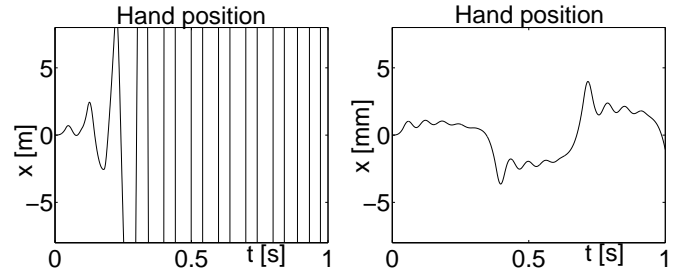


Fig. 5. One dimensional peg-in-hole task used to illustrate the influence of local proxy deformation on the stability of the interaction in Figure 6, and used to illustrate the influence of local geometry on user's perception of locally cluttered virtual environments in Figure 9.

geometry (see Figure 4). Adjusted (deformed) proxy geometry eliminates force discontinuities at model updates and maintains continuity of proxy motion. Hence, it alleviates the destabilizing effect of the virtual environment delay. This advantage of the proxy deformation technique is illustrated through a simulated one degree of freedom peg-in-hole manipulation. In the simulation, the user shakes a 1kg peg against the exactly fitting hole walls by applying a force $f_h = 5 \sin\left(\frac{2\pi}{10}\right)$ N (see Figure 5(a)). The simulation delay is 20ms and the stiffness and damping of the local contacts are $K_{contact} = 10000$ N/m and $B_{contact} = 15$ N/(m/s), respectively. For simplicity, the local model is directly coupled to the haptic device (i.e., the environment forces are directly applied to the user's hand), as depicted in Figure 5(b). Moreover, the origin of the x -axis corresponds to the middle of the virtual hole. The trajectory of the user's hand is plotted in Figure 6(a) for the case when contact penetration may be discontinuous at model updates, and in Figure 6(b) for the case when contact penetration continuity is maintained via proxy deformation. Note that local penetration discontinuities destabilize the example interaction, while proxy deformation maintains the interaction stable.

Proxy deformation allows the user to drift relative to the virtual environment. This drift is noticeable during fast motions in locally cluttered virtual environments, when significant proxy deformation may occur in order to maintain the continuity of the penetration of the local contacts (see the simulation example in Section III-C). Therefore, the haptically perceived free space may be much larger than the free space perceived visually. To alleviate this problem, ϵ -active geometry is used in the local model of interaction.



(a) Local model with active geometry (no proxy deformation).

(b) Local model with active geometry and proxy deformation.

Fig. 6. Position of the user's hand during the simulated one dimensional peg-in-hole task depicted in Figure 5 for haptic interaction via a local model with penetration discontinuities at model updates (Figure 6(a)) and via a local model with proxy deformation (Figure 6(b)).

C. ϵ -active geometry

To diminish drift and allow realistic haptic rendering of small clearances (in virtual environments where rigid constraints are enforced exactly) and of tight constraints (in penalty-based virtual worlds), the local model is augmented by including constraints within ϵ distance from the virtual tool. These constraints are computed in the virtual environment simulation via performing collision detection between the virtual environment and an ϵ -active virtual tool. If the virtual tool is convex, the ϵ -active virtual tool is obtained by sweeping a sphere of radius ϵ over the volume of the virtual tool, as described in [31] and depicted in Figure 7(a). If the virtual tool has concave surfaces, then typical simulation packages represent it as a collection of convex polyhedra [7], [8] and the ϵ -active virtual tool is obtained by sweeping a sphere of radius ϵ over the volume of each component polyhedron, as illustrated in Figure 7(b). Furthermore, the contacts computed by the simulation are clustered as proposed in [8] before sending them to the local model of interaction. Contact clustering maintains the stability of the haptic interaction in concave configurations, where the simulation may compute a large number of contacts whose combined stiffness may exceed the stiffness that the haptic device can stably render to users [32]. An example contact configuration involving a concave ϵ -active virtual tool and the corresponding contacts computed by the simulation are depicted in Figures 7(b) and 7(c), respectively (for clarity, only the contact normals are shown in Figure 7(c)). Its local model representation is shown in Figure 7(d).

ϵ -active geometry reduces drift by adding prediction capabilities to the proposed model: constraints are sent to the local model before they become active. Moreover, the approach is simple to implement and general enough to be applicable to any virtual environment. In addition to diminishing drift, the ϵ -active virtual tool selects a unique constraint normal at degenerate contacts, as illustrated in Figure 8 for a vertex-vertex contact of the virtual tool with the virtual environment. Effectively, the ϵ -active virtual tool provides a method for interpolating the constraint normal at degenerate contacts and, thus, alleviates the singularity in the constraint normal

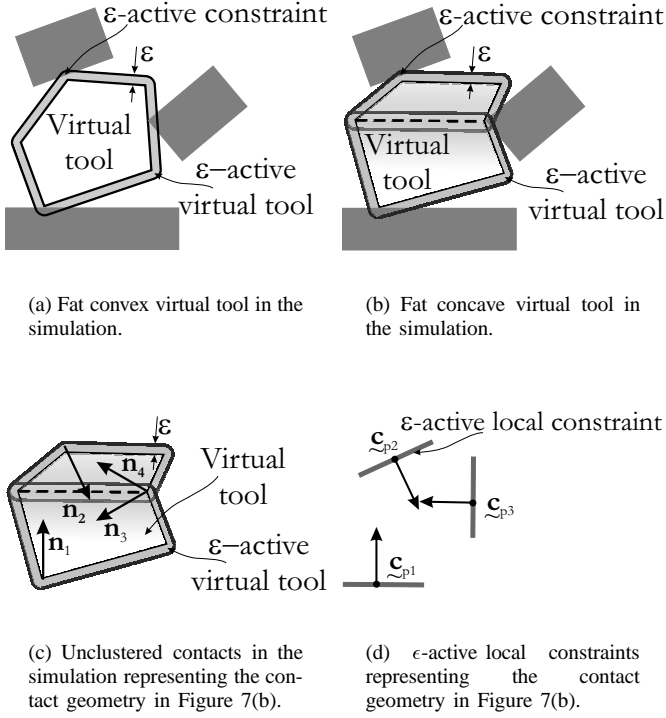


Fig. 7. ϵ -active constraints included in the local model through the use of an ϵ -active virtual tool, obtained by sweeping a sphere of radius ϵ over the volume of each convex polyhedron composing the virtual tool.

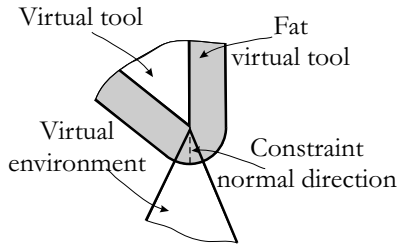


Fig. 8. The ϵ -active virtual tool eliminates the singularity in the constraint normal computation at a vertex-vertex contact between the virtual tool and the virtual environment by selecting a unique constraint normal direction.

computation at degenerate contacts.

At simulation updates, ϵ -active geometry is included in the local model by computing the local contact position according to:

$$\tilde{\mathbf{p}} = \mathbf{R} \left(\tilde{\mathbf{c}}_p + (s - \epsilon) \mathbf{n} - \tilde{\mathbf{c}}_{COM} \right), \quad (6)$$

for new local contacts and existing local contacts with separation s_p obeying $s_p < s - \epsilon < 0$ or $0 < s - \epsilon$, according to Equation (5) for existing local contacts with separation s_p obeying $s - \epsilon < s_p < 0$, and according to Equation (4) for existing local contacts with separation s_p obeying $s - \epsilon < 0 < s_p$. In Equation (6), s is the separation between the ϵ -active virtual tool and the virtual environment as reported by the simulation.

The ability of the ϵ -active geometry to improve user's perception of a tightly constrained virtual tool generated using a penalty-based simulation is illustrated through the simulated

one degree of freedom peg-in-hole manipulation depicted in Figure 5(a). The local constraints, the trajectory of the user's hand, and the environment forces acting on the virtual peg are plotted in Figures 9(a), 9(c), and 9(e), respectively, for the case when active geometry is included in the local model. They are plotted in Figures 9(b), 9(d), and 9(f), respectively, for the case when ϵ -active geometry within 5mm of the virtual tool is sent to the local model.

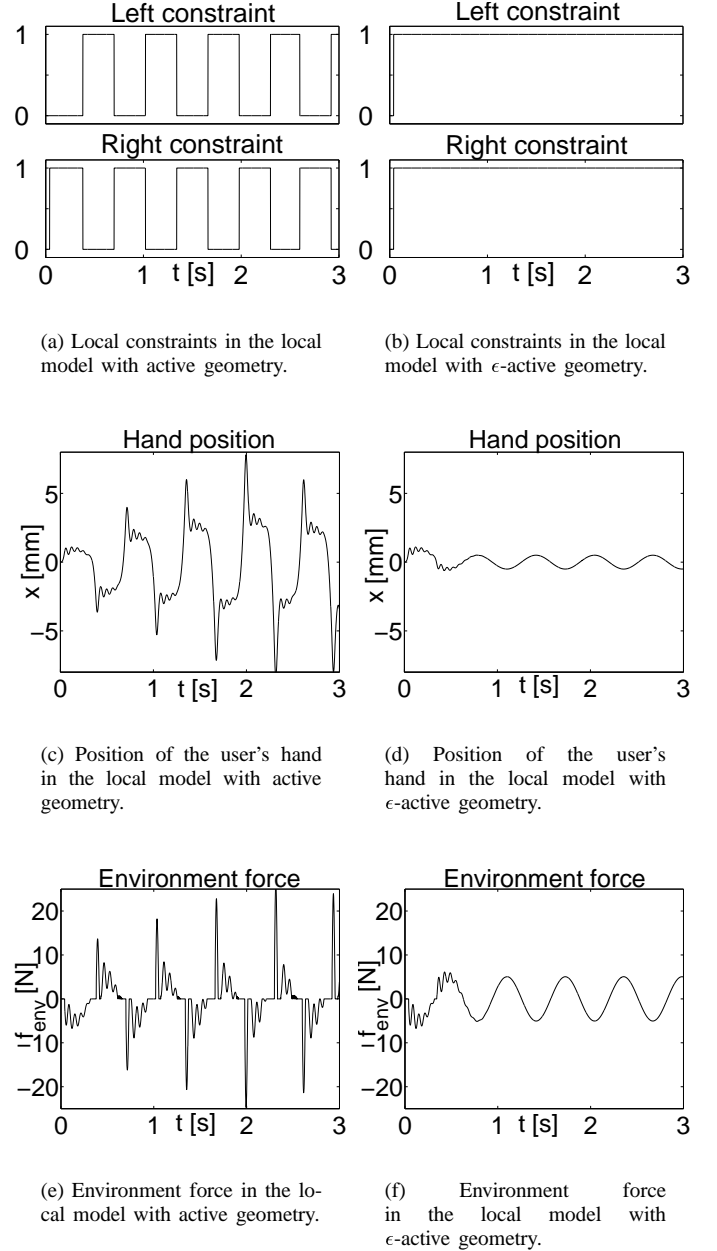


Fig. 9. Simulation results for the one dimensional peg-in-hole task depicted in Figure 5 for a local model with active geometry (Figures 9(a), 9(c), and 9(e)), and for a local model with ϵ -active geometry within 5mm of the virtual tool (Figures 9(b), 9(d), and 9(f)).

Note that the virtual peg exactly fits the virtual hole. Therefore, in the virtual environment, the peg is in contact with both hole walls when its centre of mass is at the origin. The peg penetrates the right virtual wall when the coordinate of its

centre of mass is positive, and it penetrates the left wall when the coordinate of its centre of mass is negative. Figures 9(a) and 9(c) show that, due to virtual environment delay, the active constraints arrive late to the local model, when the penetration between the virtual tool and the virtual environment is significant (several mm in this example). Figure 9(e) shows that local proxy deformation ensures force continuity at model updates and, thus, the stability of the interaction. However, user's perception of the virtual hole is inadequate. Note in Figure 9(e) that environment forces act on the virtual peg intermittently. In other words, the user feels the virtual hole only occasionally despite the fact that the hole exactly fits the peg. Hence, active geometry is not sufficient for realistic haptic rendering of tight constraints in penalty-based simulations.

Figures 9(b), 9(d), and 9(f) demonstrate that ϵ -active geometry within 5mm of the virtual tool enables users to realistically perceive the virtual hole. Figure 9(b) shows that constraints representing both virtual walls are sent to the local model throughout the interaction. This is because both constraints are always within a 5mm neighborhood of the virtual peg. Therefore, environment forces act on the virtual peg continuously and the user feels that the peg is constrained by a tight hole. Note that user's deviation from the nominal position in Figure 9(d) is due only to the limited stiffness of the virtual walls. Hence, ϵ -active geometry within 5mm of the virtual tool eliminates the effect of the virtual environment delay on the user's perception of the virtual interaction.

IV. LOCAL DYNAMICS

The realism of the kinesthetic feedback applied to users through the present local model of interaction depends on the local proxy dynamics in addition to the local geometry. The local proxy dynamics are designed to achieve two objectives: (i) to improve the perceived rigidity of the virtual contacts; and (ii) to allow users to operate both virtual objects and virtual linkages. The perceived rigidity of the virtual environment is enhanced by incorporating the impulse-augmented penalty simulation approach proposed in [33] in the local model of interaction. In this approach, contacts are perfectly rigid when they arise and have limited stiffness afterwards [33]. Therefore, users perceive large, impulsive forces upon contact and penalty and friction forces during contact. The impulsive forces improve the perceived rigidity [34], [35] and the stability [33] of the virtual contacts. In the local model of interaction, the impulse-augmented penalty approach is incorporated via the model of proxy contact described in Section IV-A and the impulsive and penalty interaction forces are computed as detailed in Section IV-B. Haptic manipulation of linkages is enabled by defining the virtual tool to be the entire articulated structure when the user holds one of its links (see Figure 1), as described in Section IV-B.

A. The proxy contact model

The proxy contact model is defined based on the contact model of one local contact. In turn, this model incorporates predicted geometry in the impulse-augmented penalty model

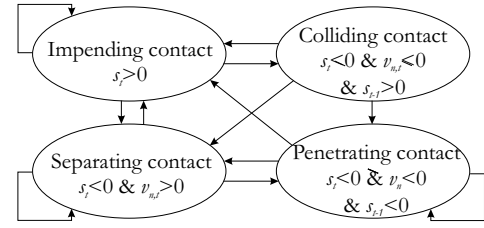


Fig. 10. The states of one local contact. t denotes the time step.

of contact [33]. In particular, the local contact model has four states (see Figure 10): impending contact, colliding contact, penetrating contact, and separating contact. At the t -th time step of the local model, a local contact is said to be an *impending contact* if it has positive separation distance $s_t > 0$. A local contact is said to be a *colliding contact* if it has nonpositive separation distance $s_t \leq 0$, nonpositive normal contact velocity $v_{n,t} \leq 0$ (i.e., nonpositive relative velocity of the proxy with respect to the constraint along the local constraint normal), and was an impending contact at the previous time step $s_{t-1} > 0$. A local contact is said to be a *penetrating contact* if it has nonpositive separation distance $s_t \leq 0$, nonpositive normal contact velocity $v_{n,t} \leq 0$, and was not an impending contact at the previous time step $s_{t-1} \leq 0$. Lastly, a local contact is said to be a *separating contact* if it has nonpositive separation distance $s_t \leq 0$ and positive normal contact velocity $v_{n,t} > 0$. Note that the impending contact state accounts for predicted geometry.

The proxy contact model has three states: free motion, colliding contact, and resting contact. The proxy is said to be in *free motion* if all local contacts are impending contacts. The proxy is said to be in *colliding contact* if at least one local contact is a colliding contact. Finally, the proxy is said to be in *resting contact* if it is neither in free motion nor in colliding contact. This proxy contact model is employed to impose local geometry on the dynamic proxy according to the approach in [33]. The following section briefly overviews the proxy dynamics, with emphasis on the additional assumptions introduced in the local model of interaction.

B. The proxy dynamics

Linkage manipulation is allowed by defining the virtual tool to be the entire linkage when the user holds one of its links, as illustrated in Figure 1. By this definition, motion constraints imposed on users by the linkage topology are encapsulated in the local proxy dynamics and nearby contacts of all links must be sent to the local model. Bilateral constraints are automatically satisfied by computing the proxy dynamics in configuration space; i.e., the bilateral constraints are embedded in the coordinate representation. If the virtual tool is a rigid object, then its Cartesian space dynamics are its configuration space dynamics.

In this approach, only contact and user applied wrenches need to be included in the local proxy dynamics. Therefore, if a virtual tool with d degrees of freedom and c contacts is considered, the proxy dynamics during free motion and resting

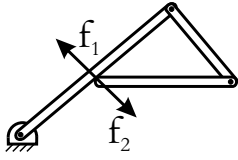


Fig. 11. One self-contact of the virtual tool is represented as two contacts in the local proxy dynamics, i.e., one contact on each contacting link.

contact are computed by:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \sum_{i=1}^c \mathbf{J}_i^T(\mathbf{q})\mathbf{f}_i + \mathbf{J}_h^T(\mathbf{q})\mathbf{F}_h, \quad (7)$$

with $c = 0$ during free motion. In Equation (7), $\mathbf{D}(\mathbf{q})_{d \times d}$ is the proxy's configuration-space inertia matrix, $\mathbf{G}(\mathbf{q})_{d \times 1}$ are the gravitational terms, $\mathbf{J}_i(\mathbf{q})_{d \times 3}$ is the proxy's Jacobian computed at the i -th contact, $\mathbf{f}_{i3 \times 1}$ is the contact force at the i -th contact, $\mathbf{J}_{h6 \times 6}(\mathbf{q})$ is the proxy's Jacobian computed at the user's hand (the hand Jacobian), $\mathbf{F}_{h6 \times 1}$ is the user-applied wrench, and $\mathbf{q}_{d \times 1}$, $\dot{\mathbf{q}}_{d \times 1}$, and $\ddot{\mathbf{q}}_{d \times 1}$ are the configuration space position, velocity, and acceleration of the proxy, respectively. Furthermore, a contact between two links of the virtual tool, called a self-contact, is represented once on each link involved in the contact (see Figure 11). Note that, similar to work in [36], [37], Coriolis and centripetal effects are neglected². In subsequent derivations, the dependence on the instantaneous proxy state of all terms is implied.

During resting contact, the proxy contact model described in Section IV-A enforces rigid constraints only approximately, through penalizing constraint violation. Therefore, the contact forces \mathbf{f}_i have a component along the local constraint normal, that models the limited rigidity of the local contact, and a component orthogonal to the local constraint normal, that models dry friction³. Hence, Equation (7) is solved for proxy acceleration and the proxy state is advanced using a fixed step integrator compatible with the requirements of the haptic control loop. Furthermore, the constraints imposed on the proxy by the local geometry are represented to the user through the dynamically consistent inverse of the proxy Jacobian computed at the user's hand [40], \mathbf{J}_h^\dagger :

$$\mathbf{F}_{env} = \mathbf{J}_h^\dagger \sum_{i=1}^c \mathbf{J}_i^T \mathbf{f}_i. \quad (8)$$

When new proxy contacts arise, i.e., at least one local contact is a colliding contact, the proxy is in colliding contact. Then, constraints are exactly enforced through impulses applied at all colliding and penetrating local contacts. Note that impulses must be applied at colliding and penetrating local contacts in order to enforce the rigidity of all proxy contacts during a proxy collision. In the present local model of interaction, these impulses are computed by applying the multiple collision resolution algorithm proposed in [33] to a

²Typically, users do not manipulate virtual linkages using high speeds. Therefore, neglecting the Coriolis and centripetal effects is likely to not introduce large errors.

³Dry friction can be implemented using any friction model that employs only local contact information, such as the models proposed in [38] or [39].

single proxy. The algorithm in [33] uses three assumptions to ensure nonincreasing kinetic energy of the virtual environment during collisions: (i) that collisions are frictionless; (ii) that all local contacts have the same coefficient of restitution $e_1 = \dots = e_c = e$; and (iii) that Newton's restitution hypothesis applies at all colliding and penetrating local contacts:

$$v_{n1} = -e v_{n0}. \quad (9)$$

In Equation (9), the indices 0 and 1 are used for pre-collision and post-collision quantities, respectively, and $e \in [0, 1]$ is the coefficient of restitution. $e = 1$ corresponds to perfectly elastic collisions, during which the proxy loses no energy. $e = 0$ corresponds to perfectly plastic collisions, whereby the proxy does not separate from the constraints at the end of the collision state.

To ensure non-increasing kinetic energy of the proxy during collisions, one additional assumption is used in the proposed local model of interaction: that the proxy collides with static local constraints. Although this assumption precludes the transfer of energy from the virtual environment to the user upon contact, it permits the virtual world to transfer energy to the user during contact. Hence, it allows contact interaction between the proxy and moving virtual environments while ensuring that local collision resolution does not affect the coupled stability of the haptic interaction [41]. This assumption is enforced by taking the local constraint velocity to be momentarily zero during proxy collisions, i.e., $v_n = \mathbf{n}^T \mathbf{J} \dot{\mathbf{q}}$, where \mathbf{J} is the proxy Jacobian at the point where v_n is computed.

Using the four assumptions together with the first order proxy dynamics, obtained through time integration of Equation (7):

$$\mathbf{D} \dot{\mathbf{q}} = \mathbf{D} \dot{\mathbf{q}}_0 + \sum_{i=1}^m \mathbf{J}_i^T \int_{t_0}^t \mathbf{f}_i dt = \mathbf{D} \dot{\mathbf{q}}_0 + \mathcal{J}_c^T \mathbf{p}, \quad (10)$$

the configuration-space impulse is computed by [33]:

$$\mathbf{p} = -(1 + e) (\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T)^\dagger \mathcal{J}_c \dot{\mathbf{q}}_0. \quad (11)$$

In Equation (10), m is the number of colliding and penetrating local contacts (out of the c colliding, penetrating, and non-penetrating local contacts). In Equation (11), $\mathcal{J}_c = [\mathbf{J}_1^T \mathbf{n}_1 \dots \mathbf{J}_i^T \mathbf{n}_i \dots \mathbf{J}_m^T \mathbf{n}_m]^T$, the \dagger sign denotes matrix pseudo-inversion, and $\mathbf{p} = \left(\int_{t_0}^t \mathbf{f}_1 dt \dots \int_{t_0}^t \mathbf{f}_m dt \right)^T$ is the vector of frictionless collision impulses. Note that collisions are modeled as instantaneous events (i.e., $t \rightarrow t_0$). As a result, the hand and gravitational forces contribute no impulses to the first order dynamics of the system.

The frictionless collision impulses are rendered to users as an impulsive wrench that, over one haptic step Δt , changes the momentum of the proxy by the same amount as the collision impulses:

$$\mathbf{F}_{env} = \frac{\mathcal{J}_c^T \mathbf{p}}{\Delta t}. \quad (12)$$

The impulsive wrench enhances the perceived rigidity of the virtual contacts [34], [35]. In essence, it generates large hand accelerations without requiring increased contact stiffness and

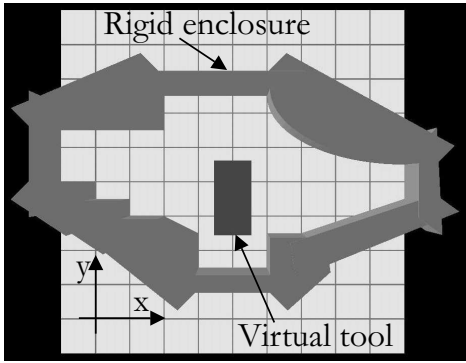


Fig. 12. The testbed virtual world used for the experimental validation of the local model of interaction (coordinates shown are the world coordinates).

damping. Depending on the coefficient of restitution, the impulsive wrench may also eliminate the bouncing upon contact, as demonstrated in Section V.

V. EXPERIMENTS

In this section, the performance of the proposed local model of interaction is compared to the performance of the intermediate representation proposed in [15] through experiments performed by coupling a planar haptic device to a testbed virtual environment simulated using VortexTM, a physics engine developed by CMLabs Simulations Inc. (www.criticalmasslabs.com). In the implementation, the local model and the device control run at 512Hz on the haptics server, a 700MHz Pentium III personal computer running VxWorksTM. A 2.4GHz Pentium IV personal computer running Windows2000 displays the virtual environment at rates ranging from 30Hz to 60Hz. The two computers communicate via a UDP socket, with the local model acting as the server and the virtual environment being the client. The local model polls the socket for new data at the beginning of each control step. When new data is available, the local model updates its state and acknowledges the receipt of the packet by sending back the proxy state. The virtual environment sends packets asynchronously, each time it has completed a simulation step. The initial configuration of the testbed virtual environment is shown in Figure 12.

Three experiments are performed. The first two experiments investigate the realism of haptic rendering of static contact and the user's perception of tight virtual spaces, respectively. The third experiment demonstrates that much larger contact stiffness can be rendered to users via the local model than via the intermediate representation. In all experiments, the virtual tool is initially at rest. Since users are not able to apply the exact same wrenches during successive trials, the performance of the proposed local model of interaction cannot be compared to the performance of the intermediate representation through experiments involving manipulations performed by humans. To enable such comparison, the experiments use controlled user-applied wrenches. The same initial conditions and the same "user" are ensured during one experiment through replacing the user's hand by controlled wrenches. Since the haptic device is an impedance-type interface, elimination of

the adaptive damping associated with the user manipulation of the device represents a worst-case scenario for stability [42]. In the experiments, the stiffness and damping of the local contacts are $k_{lc} = 4000\text{N/m}$ and $b_{lc} = 30\text{N/(m/s)}$, respectively. These values represent the maximum contact impedance for which the interaction is stable in the intermediate representation (i.e., the virtual tool can be inserted in the tight-fitting hole at the bottom of the virtual world and remains stable upon being left there). Furthermore, collisions are considered perfectly plastic, i.e., $e = 0$. Interaction with virtual environments with other coefficients of restitution can be implemented via a different e value and was demonstrated elsewhere [43].

A. Haptic rendering of static contact

The first experiment investigates the haptic rendering of static contact via the proposed local model of interaction. In the experiment, the user's hand is represented by a wrench $\mathbf{F} = (0\text{N } 0.5\text{N } 0\text{Nm})^T$ that pushes the virtual tool in Figure 12 towards the bottom-most horizontal wall of the rigid enclosure (located at $y_{COM} = 124\text{mm}$), as schematically depicted in Figure 13. The applied wrench is limited by the virtual environment performance (i.e., the virtual tool pops through the wall for larger hand forces). The experimental device trajectories are plotted in Figure 14. They are obtained by interfacing the haptic device to the virtual world through: (i) the intermediate representation ("IR"); (ii) the proposed local model including active geometry ("LM₀"); and (iii) the proposed local model including predicted geometry within 5mm from the virtual tool ("LM₅").

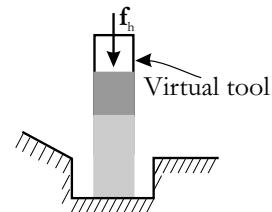
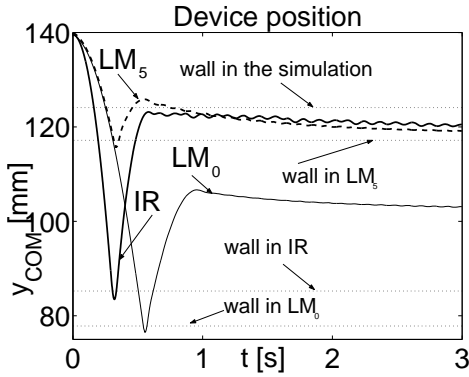
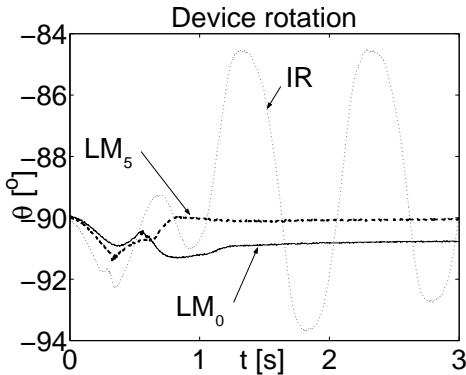


Fig. 13. Schematic of the experiment investigating the haptic rendering of static contact. Initial virtual tool position is shown in black. Later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

The trajectories in Figure 14(a) show the users' motion toward the virtual wall and the users' bounce once they hit the wall. Note that users perceive the virtual wall at various locations when the intermediate representation, the local model with active geometry, and the local model with ϵ -active geometry within 5mm of the virtual tool interface the device to the virtual world. In particular, users feel the virtual wall at $y_{COM} = 118\text{mm}$ in the local model with ϵ -active geometry, at $y_{COM} = 78\text{mm}$ in the local model with active geometry, and at $y_{COM} = 84\text{mm}$ in the in-



(a) Due to ϵ -active geometry, users perceive the virtual wall much closer to its position in the virtual environment ($y_{COM} = 124\text{mm}$). They perceive the wall at $y_{COM} = 118\text{mm}$ in the LM_5 , at $y_{COM} = 78\text{mm}$ in the LM_0 , and at $y_{COM} = 84\text{mm}$ in the IR.



(b) Proxy deformation allows the device to settle upon the wall in the LM_5 and LM_0 . Penetration discontinuities cause a limit cycle in the IR.

Fig. 14. Experimental device trajectories obtained when the virtual tool is pushed with a constant force towards the bottom-most horizontal wall. The intermediate representation (IR), the local model with active geometry (LM_0), and the local model with ϵ -active geometry within 5mm from the virtual tool (LM_5) interface the device to the simulation. The contact stiffness is $k_{l,c} = 4000\text{N/m}$, which is the maximum value for stable interaction in the IR, LM_0 , and LM_5 .

intermediate representation⁴. Hence, users perceive the virtual wall closest to its position in the virtual environment when ϵ -active geometry is included in the local model of interaction. The experiment demonstrates that ϵ -active geometry diminishes the difference between the position of the constraint in the virtual environment and its position as perceived by users. While such difference depends on the user-applied forces and on the virtual world configuration, the experiment illustrates that ϵ -active geometry within a relatively small neighborhood of the virtual tool (5mm in this experiment) can significantly reduce it (by approximately 3.5cm in this case).

⁴Note that users perceive the wall at different locations in the intermediate representation and in the local model with active geometry because the dissimilar controllers apply different forces on their hand. Before hitting the wall, users move freely in the intermediate representation, and they are position coordinated with the proxy in the local model.

The experiment also illustrates the role of proxy deformation in alleviating perceptual artifacts caused by discontinuities at simulation updates. Note in Figure 14(b) that the device settles upon the wall in the local model with active and ϵ -active geometry, and enters a limit cycle in the intermediate representation. The limit cycle is due to penetration discontinuities at updates. In the experiment, these discontinuities are too small to destabilize the interaction. However, they are sufficiently large to cause the device to rotate and loose one contact with the wall. As a result, one constraint arrives late to the intermediate representation. Its discontinuous penetration produces opposite device rotation and a new loss of contact, and, thus, perpetuates the rotational bouncing of the device. In the local model, proxy deformation maintains the penetration of the local contacts continuous, thereby allowing the wall damping to quickly bring the device to rest.

B. Haptic rendering of tight virtual spaces

The second experiment demonstrates the role of ϵ -active geometry in improving users' perception of tight virtual spaces. In this experiment, the controlled interaction represents a peg-in-hole manipulation. The user rotates the rectangular virtual tool shown in Figure 12 by 90° and inserts it into the hole at the bottom of the rigid enclosure that exactly fits the peg. The user then releases the peg and their hand is replaced by the wrench $\mathbf{F}_h = (1 \sin(5\pi) \text{N } 0 \text{N } 0 \text{N/m})^T$, i.e., the peg is shaken horizontally by a sinusoidally varying force, as schematically represented in Figure 15. The experimental device trajectories are shown in Figure 16. They are obtained by interfacing the haptic device to the virtual environment through: (i) the intermediate representation ("IR"); (ii) the local model with active geometry (" LM_0 "); and (iii) the local model with ϵ -active geometry within 5mm of the virtual tool (" LM_5 "). These trajectories illustrate that, depending on the technique employed to haptically render the virtual hole, the device travels different distances along the x direction at different locations along the y direction in the virtual world.

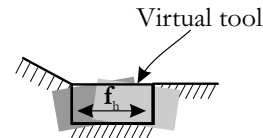
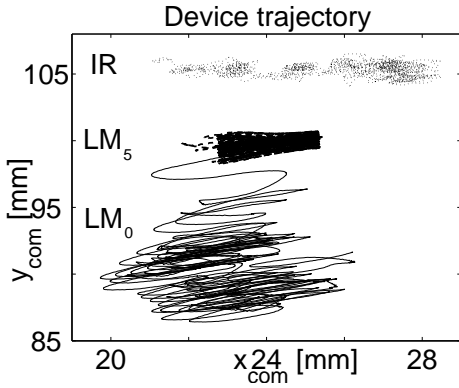
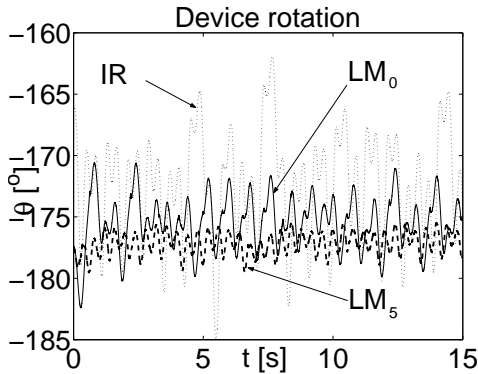


Fig. 15. Schematic of the experiment investigating the haptic rendering of tight virtual spaces. Initial virtual tool position is shown in black. Later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

Local geometry and local proxy deformation explain the differences in the three trajectories. Thus, the constraints (representing both the lateral and the bottom walls of the virtual hole) arrive late in the intermediate representation and in the local model with active geometry. The same constraints arrive before becoming active in the local model with ϵ -active geometry. Therefore, the intermediate representation and the local model with active constraints render only approximations of the hole geometry, while the local model with ϵ -active constraints renders the exact hole geometry



(a) The delayed updating of the active constraints causes larger horizontal device motion in the IR and in the LM₀ than in the LM₅. Bouncing due to discontinuous penetration in the IR and proxy deformation in the LM₀ change the vertical device position compared to the LM₅.



(b) The delayed updating of the active constraints causes larger device rotation in the IR and in the LM₀ than in the LM₅.

Fig. 16. Device trajectories obtained when the virtual tool is inserted in the tight-fitting bottom hole of the rigid enclosure in Figure 12 and pushed horizontally with a sinusoidally varying force (as shown in Figure 15). The intermediate representation (IR), the local model including active geometry (LM₀), and the local model including ϵ -active geometry within 5mm of the virtual tool (LM₅) interface the device to the simulation. The contact stiffness is $k_{lc} = 4000\text{N/m}$, which is the maximum value for stable interaction in the IR, LM₀, and LM₅. Note that the device is better constrained to the virtual hole in the LM₅ than in the IR and in the LM₀.

throughout the interaction. Approximate (i.e., partial) geometry accounts for the larger rotation and horizontal motion of the device in the IR and the LM₀ trajectories in Figure 16. Partial geometry together with discontinuities in the intermediate representation and together with proxy deformation in the local model explain the variances along the y axis in the device trajectories. In the intermediate representation, late constraints result in force discontinuities which cause the device to bounce off the bottom wall⁵. In the local model with active geometry, proxy deformation maintains force continuity despite the late arrival of the active geometry. Therefore, the device settles on the local constraints. However, the local constraints incompletely represent the virtual hole, i.e., they exist in the local

model sporadically. The proxy deforms along the y axis and the device drifts along this direction. In the local model with ϵ -active geometry, the proxy is constrained by the exact virtual environment geometry and the motion of the device is only due to the limited stiffness of the virtual hole. The results in Figure 16 validate that users perceive tight virtual spaces more realistically via the local model with ϵ -active geometry than via the local model with active geometry or via the intermediate representation. Note that users drift from the virtual hole by several mm along each direction and by up to 20° about the vertical axis when incomplete hole geometry is rendered to them during the interaction.

C. The effect of the dynamic proxy on contact stiffness

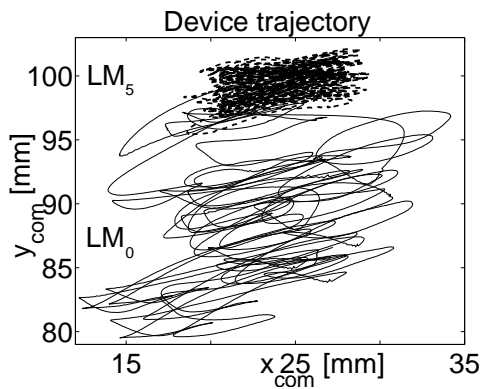
To demonstrate the effect of the dynamic proxy on the achievable contact stiffness, the peg-in-hole experiment (Figure 15) is repeated taking the stiffness and damping of the local contacts equal to $k_{lc} = 15000\text{N/m}$ and $b_{lc} = 100\text{N(m/s)}$, respectively. Moreover, the controlled wrench applied on the device is $\mathbf{F} = (3\sin(5\pi)\text{N } 0\text{N } 0\text{N/m})^T$. The contact stiffness employed in this experiment is almost four times larger than the contact stiffness used in the previous experiments. Since the stiffness value used in the previous experiments is the maximum value for which the interaction is stable in the intermediate representation, only the local model connects the device to the virtual world in this experiment. Hence, the local model provides the first approximation of rigid body interaction that stably constrains both the translation and the rotation of the haptic device during operation in tightly constrained virtual tool configurations⁶.

The experimental device trajectories obtained using the local model with active (“LM₀”) and with ϵ -active geometry (“LM₅”) are plotted in Figure 17. These trajectories demonstrate that much higher contact stiffness (four times higher for the testbed virtual environment shown in Figure 12) can stably be rendered to users via the proposed local model than via the intermediate representation. This is the result of the damping introduced via the position coordination channels of the haptic controller [26]. In turn, position coordination is enabled by the proxy dynamics.

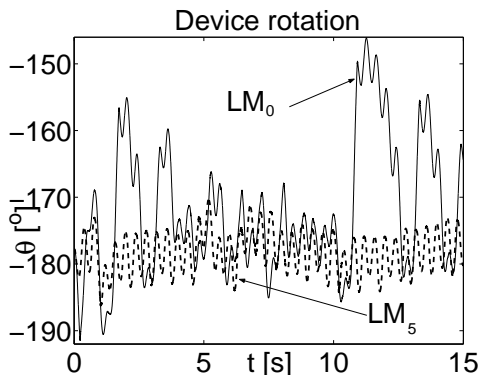
The experimental trajectories in Figure 17 also validate the positive effect of ϵ -active geometry on user’s perception of tight clearances. Note that the device travels approximately 2cm more along both the x and the y axes and rotates more around the horizontal orientation of the virtual peg when the local model incorporates active geometry than when it incorporates ϵ -active geometry within 5mm of the virtual tool. However, due to overshoot, the device penetrates into the constraints more than 5mm and the neighborhood selected for geometry prediction is not sufficient to eliminate the impact of the virtual environment delay on the interaction represented by the LM₅ trajectory. This impact is represented by the larger device motion along the x axis than permitted by the contact stiffness.

⁵Note that the bounces off the vertical walls counteract each other.

⁶Such configurations have been reported to be unstable in the prior intermediate representation of rigid body interaction [15].



(a) Despite the large contact stiffness, the translation of the device is stable regardless of whether active or ϵ -active geometry is used in the local model. Moreover, ϵ -active geometry tighter constrains the position of the device to the position of the virtual tool.



(b) Despite the large contact stiffness, the rotation of the device is stable regardless of whether active or ϵ -active geometry is used in the local model. Moreover, ϵ -active geometry tighter constrains the rotation of the device to the rotation of the virtual tool.

Fig. 17. Device trajectories obtained when the virtual tool is inserted in the tight-fitting bottom hole of the rigid enclosure in Figure 12 and pushed horizontally with a sinusoidally varying force (as depicted in Figure 15). The contact stiffness $k_{lc} = 15000\text{N/m}$ is four times larger than in previous experiments and the interaction is unstable in the intermediate representation. Only the local model with active geometry (LM_0) and with ϵ -active geometry within 5mm from the virtual tool (LM_5) interface the device to the simulation.

VI. CONCLUSIONS

The local model of rigid body interaction proposed in this work comprises partial virtual environment geometry and an approximate representation of the virtual tool dynamics called the dynamic proxy. The model selects the local geometry in a topology-independent manner. Therefore, it allows a haptic device to be coupled to any simulation with interactive performance, regardless of the algorithms and of the data structures used to generate the virtual world. At simulation updates, the local model alleviates perceptual artifacts and avoids instability due to model discontinuities via proxy deformation and via ϵ -active geometry (i.e., virtual environment geometry within ϵ distance from the virtual tool). Between simulation updates, the dynamic proxy allows physically-based virtual tool interactions to be computed locally regard-

less of whether users manipulate a virtual object or a virtual linkage. A second advantage of the dynamic proxy is that it enables the haptic controller to coordinate both wrenches and body positions between the haptic device and the proxy. Hence, it facilitates the use of a controller that enhances the realism of the interaction compared both to the virtual coupler [1], [2] and to the direct coupling approach [15]. Compared to the virtual coupler, this controller enables users to feel physical phenomena represented through fast force transitions, such as collisions and friction. Compared to direct coupling, the controller introduces additional damping via the position coordination channels and, thus, allows stiffer contacts to be rendered to users. Experiments performed using a planar virtual world interfaced to the haptic device through the proposed local model of interaction illustrate that users more realistically perceive tight virtual spaces and that they can manipulate stiffer objects. These experiments also show that, unlike the intermediate representation [15], the local model stably constrains both the translation and the rotation of the device. They report the first local approximation of rigid body interaction that can fully interface a haptic device to a multibody simulation with interactive performance.

The proposed local model does not guarantee the stability of the interactions for arbitrary virtual environment geometries. Moreover, its performance degrades during fast user motions. Future work will focus on computing physically-motivated forces that represent guaranteed limited stiffness and on enhancing the local geometry based on predicting user motion.

REFERENCES

- [1] J. Colgate, M. Stanley, and J. Brown, "Issues in the Haptic Display of Tool Use," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, 1995, pp. 140–145.
- [2] R. Adams and B. Hannaford, "Stable Haptic Interaction with Virtual Environments," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 465–474, June 1999.
- [3] C. Zilles and J. Salisbury, "A Constraint-based God Object Method for Haptic Display," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, Chicago, IL, 1995, pp. 146–151.
- [4] J. Brown and J. Colgate, "Passive Implementation of Multibody Simulations for Haptic Display," in *Proc. ASME International Mechanical Engineering Congress and Exhibition*, Dallas, TX, 1997, pp. 85–92.
- [5] B. Chang and J. Colgate, "Real-Time Impulse-Based Simulation of Rigid Body Systems for Haptic Display," in *Proc. ASME International Mechanical Engineering Congress and Exhibition*, Dallas, Texas, 1997, pp. 1–8.
- [6] D. Ruspini, K. Koralov, and O. Khatib, "The Haptic Display of Complex Graphical Environments," in *Proc. SIGGRAPH*, Los Angeles, CA, 1997, pp. 345–352.
- [7] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha, "Six Degree-of-Freedom Haptic Display of Polygonal Models," in *Proc. IEEE Visualization*, 2000, pp. 139–146.
- [8] Y. Kim, M. Otaduy, M. Lin, and D. Manocha, "Six-Degree-of Freedom Haptic Display Using Incremental and Localized Contact Computations," *Presence: Teleoperators and Virtual Environments*, vol. 12, no. 3, pp. 277–295, June 2003.
- [9] D. Johnson and P. Willemsen, "Six Degree-of-Freedom Haptic Rendering of Complex Polygonal Models," in *Proc. 11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Los Angeles, Ca, 2003, pp. 229–235.
- [10] S. Vedula and D. Baraff, "Force feedback in interactive dynamic simulation," in *Proc. First PHANTOM Users Group Workshop*, 1996, pp. 54–57.
- [11] Y. Adachi, T. Kumano, and K. Ogino, "Intermediate Representation for Stiff Virtual Objects," in *Proc. IEEE Virtual Reality Annual International Symposium*, Research Triangle Park, NC, 1995, pp. 203–210.

- [12] W. Mark, S. Randolph, M. Finch, J. Van Verth, and R. Taylor II, "Adding Force Feedback to Graphics Systems: Issues and Solutions," in *Haptic Virtual Reality for Blind Computer Users, Assistive Technologies*, 1998, pp. 92–99.
- [13] C. Mendoza and C. Laugier, "Realistic Haptic Rendering for Highly Deformable Virtual Objects," in *Proc IEEE Virtual Reality*, 2001, pp. 264–270.
- [14] J. Park and G. Niemeyer, "Haptic Rendering with Predictive Representation of Local Geometry," in *Proc. 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, IL, 2004, pp. 331–338.
- [15] P. Berkelman, R. Hollis, and D. Baraff, "Interaction with a Realtime Dynamic Environment Simulation using a Magnetic Levitation Haptic Interface Device," in *Proc. International Conference on Robotics and Automation*, Detroit, Michigan, 1999, pp. 3261–3266.
- [16] J. Basch, J. Erickson, L. Guibas, J. Herschberger, and L. Zhang, "Kinetic Collision Detection between Two Simple Polygons," in *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999, pp. 102–111.
- [17] J. Erickson, L. Guibas, J. Stolfi, and L. Zhang, "Separation Sensitive Collision Detection for Convex Objects," in *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999, pp. 327–336.
- [18] D. Kirkpatrick, J. Snoeyink, and B. Speckman, "Kinetic Collision Detection Simple Polygons," *International Journal of Computational Geometry and Applications*, vol. 12, pp. 3–27, 2002.
- [19] D. Corning and O. Stadt, "Velocity-Aligned Discrete Oriented Polytopes for Dynamic Collision Detection," University of California, Davis, CA, Tech. Rep. TCSE-2005-25, September 2004.
- [20] J. Canny, "Collision Detection for Moving Polyhedra," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 200–209, 1986.
- [21] S. Cameron, "Collision Detection by Four-Dimensional Intersection Testing," in *Proc. IEEE International Conference on Robotics and Automation*, 1990, pp. 291–302.
- [22] B. Herzen, A. Barr, and H. Zatz, "Geometric Collisions for Time Dependent Parametric Surfaces," *Computer Graphics*, vol. 24, pp. 39–48, 1990.
- [23] P. Hubbard, "Collision Detection for Interactive Graphics Applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 3, pp. 218–230, September 1995.
- [24] D. Redon, A. Kheddar, and S. Coquillart, "An Algebraic Solution to the Problem of Collision Detection for Rigid Polyhedral Objects," in *Proc. IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.
- [25] D. Ruspini, K. Koralov, and O. Khatib, "Haptic Interaction in Virtual Environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS'97*, Grenoble, France, 1997.
- [26] M. Sirospour, S. DiMaio, S. Salcudean, P. Abolmaesumi, and C. Jones, "Haptic Interface Control – Design Issues and Experiments with a Planar Device," in *Proc. IEEE International Conference on Robotics and Automation*, San Francisco, Ca, 2000, pp. 789–794.
- [27] J. Brown and J. Colgate, "Physics-based Approach to Haptic Display," in *Proc. Topical Workshop on Virtual Reality, International Symposium on Measurement and Control in Robotics*, vol. 1, Houston, TX, 1994, pp. 101–106.
- [28] J. Canny, *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1987.
- [29] D. Baraff, "Curved Surfaces and Coherence for Non-Penetrating Rigid Body Simulation," *ACM Computer Graphics*, vol. 24, no. 4, pp. 19–28, 1990.
- [30] —, "Rigid body simulation," in *An Introduction to Physically Based Modelling, Course Notes*. ACM SIGGRAPH, 1995, pp. 1–68.
- [31] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Fast Proximity Queries with Swept Sphere Volumes," University of North Carolina, Chapel Hill, NC, Tech. Rep. TR99-018, 1999.
- [32] J. Brown, "A Theoretical and Experimental Investigation into the Factors Affecting the Z-width of a Haptic Display," Master's thesis, Northwestern University, 1998.
- [33] D. Constantinescu, S. Salcudean, and E. Croft, "Haptic Rendering of Rigid Contacts using Impulsive and Penalty Forces," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 309–323, June 2005.
- [34] L. Rosenberg and B. Adelstein, "Perceptual Decomposition of Virtual Haptic Surfaces," in *Proc. IEEE Symposium on Research Frontiers in Virtual Reality*, San Jose, CA, 1993, pp. 46–53.
- [35] D. Lawrence, L. Pao, A. Dougherty, M. Salada, and Y. Pavlou, "Rate-Hardness: A New Performance Metric for Haptic Interfaces," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 4, pp. 357–371, 2000.
- [36] D. Ruspini and O. Khatib, "Dynamic Models for Haptic Rendering Systems," in *Advances in Robot Kinematics: ARK98*, Strobl/Salzburg, Austria, 1998, pp. 523–532.
- [37] —, "Haptic Display for Human Interaction with Virtual Dynamic Environments," *Journal of Robotic Systems*, vol. 18, no. 2, pp. 769–783, 2001.
- [38] A. Nahvi, J. Hollerbach, R. Freier, and D. Nelson, "Display of Friction in Virtual Environments based on Human Finger Pad Characteristics," in *Proc. ASME Dynamics Systems and Control Division*, vol. DSC-64, 1998, pp. 179–184.
- [39] V. Hayward and B. Armstrong, "A new computational model of friction applied to haptic rendering," in *Experimental Robotics VI. Lecture Notes in Control and Information Sciences*, P. Corke and J. Trevelyan, Eds. Springer: New-York, 2000, vol. LNCS 250, pp. 404–412.
- [40] O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework," *International Journal of Robotics Research*, vol. 13, no. 1, pp. 19–36, 1995.
- [41] J. Colgate, "Coupled Stability of Multiport Systems - Theory and Experiments," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 116, no. 3, pp. 419–428, 1994.
- [42] B. Hannaford and R. Anderson, "Experimental and Simulation Studies of Hard Contact Force Reflecting Teleoperation," in *Proc. IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May 1988, pp. 24–29.
- [43] D. Constantinescu, S. Salcudean, and E. Croft, "Impulsive Forces for Haptic Rendering of Rigid Contacts," in *Proc. 35th International Symposium on Robotics*, Paris, France, 2004, pp. 1–6.