LOCAL INTERACTION MODELS FOR HAPTIC RENDERING OF RIGID ENVIRONMENTS

D. Constantinescu [*] **S. E. Salcudean** [*] **E. A. Croft** [**]

[*] *Department of Electrical and Computer Engineering*
[**] *Department of Mechanical Engineering*
University of British Columbia, Vancouver, Canada

Abstract: A technique is presented for providing users with realistic haptic feedback as they interact with virtual rigid objects. The technique imposes no restrictions on the algorithms used to simulate the virtual environment. The force control loop is decoupled from the simulation by employing a local model of the interaction updated periodically by the virtual environment. This local model includes a proxy of the virtual object manipulated by the user and a geometric description of the motion constraints imposed on the proxy by neighboring objects. Two schemes are employed to coordinate the proxy and the haptic device. A planar haptic interaction system, including a rigid mass that can be translated and rotated in the presence of constraints, is used to evaluate the performance of the proposed technique.

Keywords: haptic rendering, local dynamic model.

1. INTRODUCTION

Haptic devices are computer interfaces that allow a user to interact with virtual environments (VEs) through touch and kinesthesia. They can be used for physical skill training, for scientific data understanding, and for entertainment. Due to their significant potential benefits, haptic interfaces have been the focus of a large research effort over the last ten years. New control (Adams and Hannaford, 1999) and simulation algorithms (Zilles and Salisbury, 1994), (Ruspini and Khatib, 1999) have been proposed, and a number of high-end applications have been developed (Basdogan *et al.*, 2001).

Regardless of important advantages for prospective users of all types, consumer-grade haptics have remained an illusive goal. This is partly because no technique exists to allow interfacing of a device to an arbitrary VE while guaranteeing stability of interaction and providing realistic feedback. The difficulty in developing such a technique comes from the requirements imposed on the VE simulation by both the device controller and our sense of touch, that needs force "refresh" rates of the order of hundereds of Hz.

Depending on whether an impedance or an admittance control scheme is employed, force reflection is achieved by detecting the motion or the force applied by the user and controlling the force or the motion of the device, respectively, such that the impedance or admittance rendered to the hand matches that of the VE as closely as possible. Thus, transparent interaction is achieved through control, while stable interaction is achieved through both control and simulation. Since power is transferred from the VE to the user, passive simulation techniques must be used to avoid instability (Brown and Colgate, 1997). Therefore, the VE must be rendered using passive numerical methods. Furthermore, for a large number of applications, haptic feedback is only useful if the rendered impedance accurately represents a physical phenomenon. In these applications, the simulation must compute realistic interaction forces at frequencies of the order of hundreds of Hz ("haptic rates"). However, reported rendering frequencies of typical interactive physically-based simulations of rigid bodies are only of the order of tens of Hz (Vedula and Baraff, 1996) and can slow down further during complex interactions. In response to the need for interaction stability and the need for high haptic rate, researchers have proposed three approaches.

The first approach has been the development of simulation techniques for specific applications (Nahvi *et al.*, 1998), (Thompson II and Cohen, 1999). This approach may not be suitable for consumer grade haptics, where software development costs become prohibitive.

A second approach has been the improvement of the real time performance of the VE via new collision detection methods (Gregory *et al.*, 2000) that allow general purpose rigid body dynamic algorithms to be used for rendering complex VEs. Simulations developed using this technique achieve haptic rates only by using these collision detection algorithms in conjunction with non-passive numerical methods (penalty-based dynamic algorithms and fixed step forward integration routines) and thus stability of interaction is not guaranteed (Gregory *et al.*, 2000).

The third approach is the decoupling of the force control loop from the VE. This decoupling alleviates the haptic rate demand on the simulation; therefore, VEs of increased complexity, based on passive algorithms, can be rendered. However, realistic haptic feedback is applied on the hand only if the setpoints of the control loop over one simulation step are suitable approximations of the forces acting on the virtual tool (VT) manipulated by the user. Therefore, the decoupling of the control loop from the VE requires a local model (LM) of the interaction to be available in the control loop. In essence, the LM must be a reduced simulation that runs at the fixed haptic rate (Adachi *et al.*, 1995), (Berkelman, 1999).

The present research proposes such a LM. The LM is suitable for adding haptic feedback to the manipulation of virtual rigid objects that interact through contacts. In this LM, the interaction between the VT and the VE is approximated through the interaction between the VT and neighboring objects. A proxy of the VT together with constraints imposed on the VT motion by neighboring objects comprise the LM (see Figure 1). The quality of the approximation is maintained locally by updating the LM at each step of the simulation. Local geometric awareness, i.e., knowledge of the constraints, improves the realism of the haptic feedback in two ways: (i) by allowing much stiffer environments to be rendered because of the high control rate that can be achieved; and (ii), by enabling the user to feel physical phenomena that rely upon fast force transitions, e.g., collisions and stick-slip friction.

As depicted in Figure 1, the VE coordinates the proxy and the VT using proxy state information, while the controller coordinates the proxy and the device. When the LM is updated, undesirable discontinuities may arise in the forces computed locally. These can be avoided by the techniques in Section 3.

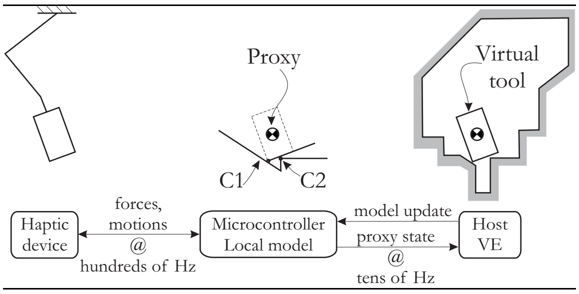

Fig. 1. Communication between the VE, the LM, and the haptic device.

The rest of the paper is organized as follows. Section 2 reviews related research that decouples the control loop from the VE. Section 3 details the LM proposed in this work for achieving such decoupling. Section 4 describes the system used to implement it. Results obtained using the LM in conjunction with two different control schemes for coordinating the proxy and the device are presented in Section 5. Conclusions and future work are discussed in Section 6.

2. PREVIOUS WORK

The decoupling of force computation from the VE has been proposed before in a haptic extension of the X-Windows graphical user interface (GUI) (Kelley and Salcudean, 1994). A haptic model of the GUI, running on a microcontroller, was used to compute forces at a fixed control rate of 1kHz. The model included constraints corresponding to window borders, pull-down menus, cursor forces towards icons, etc., and was updated asynchronously by the X-Windows host according to the status of the GUI. A LM was not needed due to the relative simplicity of the VE (few constraints, typically horizontal and vertical).

The earliest LM of force computation used in the control loop was proposed by Adachi *et al.* (1995) for point interaction within VEs. It consists of the position and outward normal of the active constraint. The geometric awareness shifts the communication delay from delay in force computation to delay in updating the geometry. Therefore, larger contact stiffnesses are achievable, but force discontinuities arise at model updates. These may destabilize the interaction. Vedula and Baraff (1996) improved the LM by using a kinematic description of the constraint that diminished force discontinuities due to contact with dynamic objects, while Mark *et al.* (1998) proposed a smoothing scheme based on linear interpolation between old and new constraints.

A LM for rigid body interaction in VEs was put forward by Berkelman (1999). The model includes all active constraints, enabling the user to feel crisp, stiff contacts. However, due to computed rotational contact stiffnesses larger than device capabilities, instability may arise during tightly constrained motions of the VT, such as during peg-in-hole insertion. Therefore, Berkelman (1999) used the local geometry only to constrain the translation of the device and he employed a virtual coupler to constrain the rotation.

3. APPROACH

A technique is proposed for adding haptic feedback to arbitrary VEs comprised of rigid bodies that interact with each other through contacts. The technique decouples the force computation from the simulation by providing the control loop with a LM of the interaction between the VT and the VE. The model is designed such that it preserves stability and achieves the required haptic rates.

The LM comprises a proxy of the VT and the constraints imposed on the VT motion by neighboring objects. The proxy has the same dynamics as the VT, but its geometry consists only of the VT features that are potentially colliding with the environment (see Figure 1). The interaction between the VT and the VE is approximated by forces applied on the proxy that have a normal component, modeling contact stiffness, and a tangential component, modeling dry friction. The normal component is computed by:

$$\boldsymbol{F}_n = (K_c s - B_c C_0) \cdot \boldsymbol{n} \qquad (1)$$

where $\boldsymbol{F}_n$ is the normal contact force, K_c and B_c are the contact stiffness and damping, s is the proxy separation from the constraint, $\boldsymbol{n}$ is the outward normal to the constraint, and C_0 is the contact velocity along the contact normal. The tangential component is computed using a simplified Coulomb model (Figure 2).

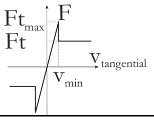

Fig. 2. Friction in the local model.

The proxy approach can improve rendering fidelity through the use of simultaneous coordination of motions and forces (Sirouspour *et al.*, 2000).

The LM is updated asynchronously, whenever the VE has advanced one time step. Simultaneously, the VE receives the state of the proxy and constrains the VT to the proxy through generalized penalty forces applied at the center of mass (COM) of the VT:

$$\boldsymbol{f}_t = \boldsymbol{K}_{ve}(\boldsymbol{x}_p - \boldsymbol{x}_t) + \boldsymbol{B}_{ve}(\boldsymbol{v}_p - \boldsymbol{v}_t), \qquad (2)$$

where $\boldsymbol{f}_t$ is the force at the COM of the VT, $\boldsymbol{K}_{ve}$ and $\boldsymbol{B}_{ve}$ are penalty coefficients, $\boldsymbol{x}_p$ and $\boldsymbol{v}_p$ are the proxy position and velocity as sent by the local model, and $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ are the position and velocity of the VT.

Two techniques are used in the LM to avoid undesirable force discontinuities at the updates: (i) the proxy is deformed locally if required; and (ii), the local geometric awareness is increased to some neighborhood of the VT. These extensions are detailed in Sections 3.2 and 3.3. In the following Section 3.1, the local geometry is briefly introduced.

3.1 Local Geometry

Increased complexity of local collision detection can improve the stability and the realism of interaction through resolving all contact transitions of the VT locally. However, this requires more geometric information, more communication bandwidth, and local processing power. Furthermore, it destroys the flexibility in coupling the device to a VE by imposing constraints on the data structures and the algorithms used by the simulation. For example, external and pseudo-internal Voronoi regions (Gregory *et al.*, 2000) must be available in the VE if adjacency information is needed locally. To allow the coupling of a device to an arbitrary VE, the local geometry is restricted to information available in any physically-based VE. Constraints are embodied in contact pairs. A contact pair has geometric and dynamic characteristics. Figure 3 shows the geometry of a contact pair. It consists of the constraint position ${}^w\boldsymbol{P}$ and its outward normal ${}^w\boldsymbol{n}$, both in world coordinates, and the contact position ${}^p\boldsymbol{r}$ in proxy coordinates. Each constraint is considered to extend infinitely. Stiffness, damping, and friction characterize the dynamic behavior of the contact.

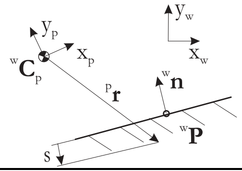

Fig. 3. Local geometry of a contact pair.

The only contact transitions that the LM is aware of are the making and braking of the contacts sent by the VE. Local collision detection iterates through all contact pairs and computes separations according to:

$$s = {}^w\boldsymbol{n}^T \cdot \left({}^w\boldsymbol{C}_p + {}^w\boldsymbol{R}_p \cdot {}^p\boldsymbol{r}\right), \qquad (3)$$

where s is the separation between the proxy and the constraint, ${}^w\boldsymbol{C}_p$ is the position of the proxy's center of mass (COM) and ${}^w\boldsymbol{R}_p$ is the rotation from the proxy frame to the world frame. An active contact has negative separation.

3.2 Local Proxy Deformation

An update of the LM may result in active constraints with significant penetration. This may be caused by: (i) the proxy not being collocated with the VT; and (ii), the user moving the VT very quickly. Discontinuities in the proxy separation from the constraints lead to discontinuities in the setpoint of the control loop and may destabilize the interaction or produce unacceptable force spikes. To deal with this problem, we propose to deform the proxy such that it only touches the new constraints. Thus, when a new contact is sent to the LM, its position in proxy coordinates is computed by:

$$^{p}\boldsymbol{r} = {}^{p}\boldsymbol{R}_w \cdot ({}^{w}\boldsymbol{P} - {}^{w}\boldsymbol{C}_p)\,, \qquad (4)$$

where $^{p}\boldsymbol{R}_w$ is the rotation from world to proxy coordinates.

The proxy is expanded back towards its actual geometry whenever a local contact is lost. This approach is independent of the local stiffness of the contact.

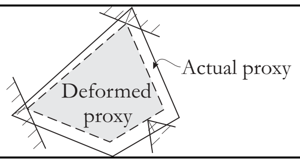

Fig. 4. Local deformation of the proxy due to violation of new constraints.

3.3 Local Prediction of Constraints

While proxy deformation ensures stable interaction through maintaining separation continuous, it allows constraints to drift relative to the VT. This drift is noticeable during fast motions in locally cluttered environments. For example, if the user moves the VT quickly in the VE shown in Figure 5, significant local proxy deformation may occur in order to maintain separation continuous. Therefore, the free space perceived haptically may be much larger that the one perceived visually. To diminish drift, the LM is augmented by including constraints within some neighborhood of the VT. The relevant neighborhood is defined by enlarging the VT by a distance ϵ along the normal to each face, as shown in Figure 5. By varying ϵ, constraints to be sent to the LM are selected from a small neighborhood of the VT if the tool motion is tightly constrained and they are selected from a larger neighborhood if the VT motion is unconstrained locally.

This technique lessens the drift by adding prediction capabilities to the proposed model: constraints are sent to the LM before contacts become active. Moreover, the approach is simple to implement and general enough to be applicable to any VE. However, if ϵ is sufficiently large, extraneous contacts may be introduced (see Figure 5).

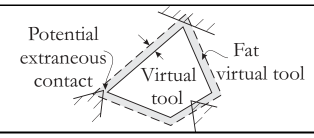

Fig. 5. Neighboring constraints included in the LM through a fat virtual tool.

The position of a predicted contact in local proxy coordinates is computed by:

$$^{p}\boldsymbol{r} = {}^{p}\boldsymbol{R}_w \cdot ({}^{w}\boldsymbol{P} + (s+\epsilon) \cdot {}^{w}\boldsymbol{n} - {}^{w}\boldsymbol{C}_p)\,, \qquad (5)$$

where s is the separation between the fat VT and the VE as reported by the simulation.

4. IMPLEMENTATION DETAILS

The inclusion of a proxy in the LM allows various coordination techniques to be used between the proxy and the device. Therefore, the realism of the haptic feedback can be improved not only through an improved LM of force computation, but also through improved device control. Two coordinating controllers are used in the present implementation: (i) an impedance controller; and (ii) a four-channel teleoperation controller optimized for transparency (Sirouspour *et al.*, 2000). The proxy is collocated with the device and the interaction forces computed locally serve as setpoints for the control loop when the impedance scheme is used, whereas the proxy state is updated locally and force and position information are coordinated between the proxy and the device when the teleoperation scheme is employed.

4.1 System Architecture

The LM described in Section 3 has been used to interface a planar twin pantograph haptic device (Sirouspour *et al.*, 2000) to a VE using a two processor hardware architecture. The VE is generated on a 450MHz Pentium III personal computer running Windows 2000^{TM}, while the local model and the device control are computed on the haptic server, a 450MHz Pentium III personal computer running VxWorks^{TM}. Communication between the two computers is performed via a local area network and the haptic device is interfaced to the personal computer via a multifunction I/O adapter for joint angle sensing and actuation.

A UDP socket is used for inter-process communication, with the LM acting as the server and the VE being the client. The LM polls the socket for new data during each control step. When new data is available, the LM updates its state and acknowledges the receipt of the packet by sending back the generalized position and velocity of the proxy. The VE sends packets asynchronously, each time it has completed a simulation step. The haptic rate is 500Hz, while the VE runs at about

40-60Hz. The VE is generated using Vortex^{TM}, a physics engine developed by CMLabs Simulations Inc. (www.criticalmasslabs.com). Compatibility between the planar device and the three dimensional VE is maintained by projecting all contact information in the plane of interaction.

4.2 Implementation of the local model of interaction

For stable interaction and elimination of perceptual artifacts, it is important that time coherence of geometry is maintained locally, i.e., new contacts are distinguished from existing contacts at updates. Time coherence is needed for minimizing constraint drift through maintaining continuity of separation and for implementing effects such as force shading (Morgenbesser and Srinivasan, 1996) and friction (Karnopp, 1985).

However, the LM uses only limited contact information. Therefore, it maintains time coherence by using spatial coherence. It assumes that the VT does not move far during one simulation step and thus updated contacts have position and outward normal close to their previous values. This assumption breaks down if the user moves the VT quickly.

Communication bandwidth limits the number of contacts that can be updated in one step. Therefore, it is possible that not all contacts reported by the VE can be sent to the LM. Time coherence is maintained by the VE by sorting the contacts in a list and by sending them to the LM in decreasing order of priority. Contacts that existed at the last simulation step are listed before new contacts and new contacts with smaller separation are listed before new contacts with larger separation.

Force shading is obtained through contact normal interpolation:

$$\boldsymbol{n}_{current} = \boldsymbol{n}_{last} + \frac{i \cdot (\boldsymbol{n}_{new} - \boldsymbol{n}_{last})}{N_{avg}} \qquad (6)$$

where $\boldsymbol{n}_{current}$ is the contact normal at the current step, $\boldsymbol{n}_{last}$ is the normal before the last update, $\boldsymbol{n}_{new}$ is the updated normal, N_{avg} is the average number of steps between updates, and i is the number of steps since the last update. Normal interpolation in conjunction with time coherence enables the system to handle arbitrarily shaped constraints.

5. EXPERIMENTAL RESULTS

The VE used for evaluating the proposed LM of interaction is shown in Figure 6. It consists of an enclosure of rigid walls, the VT (light color) together with its neighborhood of interest (dark color), and another virtual object (dark color).

The stable contact stiffness and damping achieved under impedance and under four-channel teleoperation control are given in Table 1 together with the values of the penalty coefficients. The penalty coefficients are limited by the stability of the VE, because the VE is simulated using a penalty technique. The attainable contact impedance is independent of the VE. As seen from Table 1, much stiffer contacts can be implemented if the four-channel controller is employed than if an impedance controller is used.

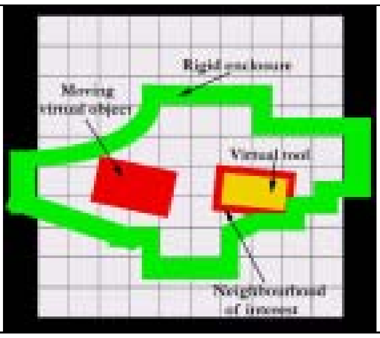

Fig. 6. Testbed VE.

Table 1. Values of the contact impedance in the LM and of the penalty coefficients in the VE for stable interaction.

Local Model	Contact impedance	
Impedance controller	$K_c = 2 \cdot 10^3$N/m	$B_{ct} = 10$Ns/m
Four-channel Teleoperation	$K_c = 10^4$N/m	$B_c = 20$Ns/m
VE	Position	Rotation
Penalty coefficients	$K_{ve} = 10^5$N/m $B_{ve} = 10^4$Ns/m	$K_{ve} = 10^2$Nm/rad $B_{ve} = 10$Nms/rad

Figures 7 and 8 depict experimental results obtained during a peg-in-hole insertion task under four-channel teleoperation control. Similar results were obtained with the impedance controller, but contacts were perceived by the user to be less stiff. The user inserts the VT in the bottom hole of the enclosure shown in Figure 6, tries to rotate it in both directions, and then quickly shakes the peg along the horizontal direction.

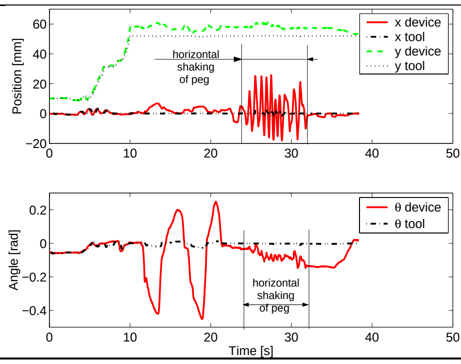

Fig. 7. Position data from a peg-in-hole task (LM with active constraints).

Figure 7 shows that the user perceives significant drift while shaking the peg horizontally when only active constraints are sent to the LM. The VE sends the bottom and the right walls when the peg is pushed to the right, and the bottom and

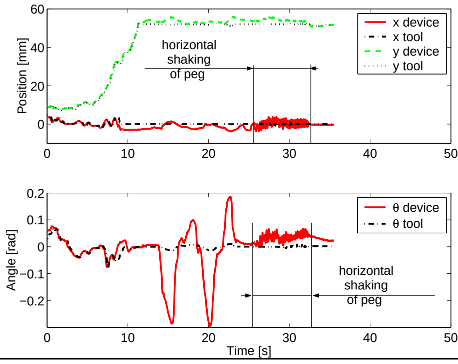

Fig. 8. Position data from a peg-in-hole task (local constraints within 5mm of the VT).

the left walls when the peg is pushed to the left. The user changes the direction of motion quickly, hence proxy penetration into the new constraints is large and the drift is noticeable. The drift is eliminated by sending constraints within a 5mm neighborhood of the VT (Figure 8). Angular errors are due to the device controller and constraint prediction decreases them little.

6. CONCLUSIONS

A technique for adding haptic feedback to the user interaction with a VE comprised of rigid bodies has been presented. It computes realistic interactions regardless of the collision detection or dynamics algorithms used by the VE. A dedicated haptics processor computes the forces reflected to the hand using a local model of the interaction enhanced with prediction capabilities. The LM improves the realism of interaction and maintains stability in the presence of variable delay. Performance degrades gracefully when the delay increases or the user moves quickly in the VE.

The main limitation of the method is the drift between the haptic device and the virtual tool during swift motions or large update delays. Future work will investigate techniques for mitigating this drift based on prediction of user motion, and extensions of the method to the haptic manipulation of articulated structures.

REFERENCES

Adachi, Y., T. Kumano and K. Ogino (1995). Intermediate Representation for Stiff Virtual Objects. In: *Proc IEEE Virt Real Annu Int Symp*. Research Triangle Park, NC. pp. 203–210.

Adams, R.J. and B. Hannaford (1999). Stable Haptic Interaction with Virtual Environments. *IEEE Trans Rob Autom* **15**(3), 465–474.

Basdogan, C., C.-H. Ho and M. A. Srinivasan (2001). Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration. *IEEE/ASME Trans Mechatronics* **6**(3), 269–285.

Berkelman, J.M (1999). Tool-Based Haptic Interaction with Dynamic Physical Simulations using Lorentz Magnetic Levitation. PhD thesis. Carnegie Mellon University.

Brown, J.M. and J.E. Colgate (1997). Passive Implementation of Multibody Simulations for Haptic Display. In: *Proc Int Mech Eng Congr Expo*. Dallasn, Texas. pp. 85–92.

Gregory, A., A. Mascarenhas, S. Ehmann, M. Lin and D. Manocha (2000). Six Degree-of-Freedom Haptic Display of Polygonal Models. In: *http://www.cs.unc.edu/ geom/6DHaptics*.

Karnopp, D. (1985). Computer Simulation of Slip-Stick Friction in Mechanical Dynamic Systems. *Trans ASME, J Dyn Syst Meas Control* **107**, 100–103.

Kelley, A.J. and S.E. Salcudean (1994). The Development of a Force Feedback Mouse and its Integration into a Graphical User Interface. In: *Proc 1994 ASME Int Mech Eng Congr Expo*. Vol. DSC-55-1. Chicago, IL. pp. 287–294.

Mark, W.R., S.C. Randolph, M. Finch, J.M. Van Verth and R.M. Taylor II (1998). Adding Force Feedback to Graphics Systems: Issues and Solutions. In: *Haptic Virt Real Blind Comput Users, Assist Technol*. pp. 92–99.

Morgenbesser, H.B. and M.A. Srinivasan (1996). Force Shading for Haptic Shape Perception. In: *Proc 1996 ASME Int Mech Eng Congr Expo*. Vol. DSC-58. Atlanta, GA. pp. 407–412.

Nahvi, A., D.D. Nelson, J.M. Hollerbach and D.E. Johnson (1998). Haptic Manipulation of Virtual Mechanisms from Mechanical CAD Designs. In: *Proc 1998 IEEE Int Conf Rob Autom*. Leueven, Belgium. pp. 375–380.

Ruspini, D and O. Khatib (1999). Collision/Contact Models for Dynamic Simulation and Haptic Interaction. In: *9th Int Symp Rob Res*. Snowbird, Utah. pp. 142–151.

Sirouspour, M.R., S.P. DiMaio, S.E. Salcudean, P. Abolmaesumi and C. Jones (2000). Haptic Interface Control – Design Issues and Experiments with a Planar Device. In: *Proc 2000 Int Conf Rob Autom*. San Francisco, Ca. pp. 789–794.

Thompson II, T. V. and E. Cohen (1999). Direct haptic rendering of complex trimmed nurbs models. In: *Proc ASME, Dyn Syst Control*. pp. 109–116.

Vedula, S. and D. Baraff (1996). Force feedback in interactive dynamic simulation. In: *Proc First PHANTOM Users Group Workshop*. pp. 54–57.

Zilles, C.B. and J.K. Salisbury (1994). A Constraint-based God Object Method for Haptic Display. In: *ASME Haptic Interfaces Virt Env Teleop Systs, Dyn Syst Control*. Chicago, IL. pp. 146–150.