Stability and Performance Analysis of Centralized and Distributed Multi-rate Control Architectures for Multi-user Haptic Interaction

Abstract

This paper is concerned with multi-user haptic simulation environments in which users can interact across an Ethernet-based Local Area Network (LAN) or a Metropolitan Area Network (MAN). Using network protocols such as the UDP and TCP/IP under normal network traffic conditions, the achievable real-time packet communication rate can be well below the 1 kHz update rate suggested in the literature for high fidelity haptic rendering. However by adopting a multi-rate control strategy, the local control loops can be executed at a much higher rate than that of the data packet transmission between the user workstations. Within such a framework, two control architectures, namely centralized and distributed are presented. Mathematical models of the controllers are developed and used in a comparative analysis of their stability and performance. The results of such analysis demonstrate that the distributed control architecture has greater stability margins and outperforms the centralized controller. It is also shown that the limited network packet transmission rate can degrade the haptic fidelity by introducing a viscous damping into the perceived impedance of the virtual object. Using the proposed models, this damping value is calculated and compensated by active control. Experiments conducted with a dual-user/dual-finger haptic platform confirm the analytical results.

KEY WORDS—Haptics and Haptic Interfaces, Haptic Interface Control, Multi-user Haptics, Cooperative Haptics, Collaborative Haptics, Collaborative Virtual Environments, Multi-rate Control Systems

1. Introduction

Haptic interaction in shared virtual environments (VEs) is an emerging area of research with promising applications (Waters and Barrus, 1997). These include but are not limited to the training of surgical tasks, haptic-enabled rehabilitation, teaching writing skills to children, sports training, as well as gaming and entertainment. For instance, a cooperative haptic simulator can permit an experienced surgeon to virtually guide a student trainee through various surgical procedures by providing corrective force feedback in real time. In such a scenario, the trainer and trainee may use identical haptic interfaces to collaboratively operate on a common virtual patient. In gaming applications, new series in group games can be developed by adding force feedback capability to multi-player games. The experimental study by Basdogan et al. (2000) has shown that the addition of haptic communication to visual feedback would significantly improve the sense of togetherness and task performance in shared virtual environments.

In its simplest form, cooperative haptics can involve multiple users with their haptic interfaces directly connected to a single host computer running the simulation. This streamlines the control design but is only feasible for a very limited...
number of users at the same workstation. Arguably, the more promising applications of shared virtual environments are in network-based haptics where users can interact across communication links such as Ethernet-based LANs, MANs or more broadly Wide Area Networks (WANs). Such configurations remove physical barriers and permit users to interact over long distances. The problem, however, is that network-based data communication is generally nondeterministic and suffers from delay, jitter, packet loss, and limited packet transmission rate. These can all adversely affect the performance and stability of cooperative haptics and pose a serious control design challenge.

Due to the above-mentioned limitations and in particular relatively long nondeterministic delays, achieving a robustly stable high-performance haptic interaction over a public communication medium such as the Internet is immensely difficult with the current technology. Instead, the scope of the present study is limited to cooperative haptic simulation over LANs and high-speed MANs. In this context, as justified by the network experiments the results of which are presented below, the network delay is assumed to be jitter-less and in the order of one (1) to three(3) transmission sample times, and the packet loss is considered negligible. The network packet transmission rate, however, remains fairly limited.

The above assumptions on the delay and packet loss, while restrictive, are valid in many important applications. LAN- and MAN-based networks can cover a single building, several buildings across a university campus, and even geographical areas as large as cities. They provide small and large companies, universities, hospitals, and other organizations with a fast and reliable means of communication. The target applications of this work are primarily in cooperative haptic-enabled training. For instance, in hospitals and medical schools connected via LANs or MANs, students and expert surgeons can haptically interact without all being present at the same location.

In Table 1, the results of a network communication experiment between two workstations running the VxWorks real-time operating system with the UDP protocol over a mixed 100Mbps/1Gbps LAN are presented. Although some of these numbers would change depending on the network traffic condition, they are representative of the typical characteristic of a LAN and to some extent a fast MAN. The data suggest that for the applications considered in this paper, the data packet loss is negligible and the network communication delay between the user workstations is a fraction of the network sampling time. However, as can be seen in the table, network protocols such as the UDP/IP and TCP/IP have limited update rates for real-time transmission of data packets. This limitation must be distinguished from that of the network bandwidth as in haptic applications the size of data packets is often small. The packet transmission rate can affect haptic fidelity, particularly in rendering of rigid objects and contacts. In our experience, the update rate of 1kHz for real-time control of haptic interfaces suggested in the literature, is well above that achievable over LANs and MANs using standard UDP/IP and TCP/IP implementations with a typical traffic load.

There have been prior attempts at achieving high-rate data communication for real-time control over an Ethernet connection. For instance in (Traylor et al., 2005), a streamlined implementation of the UDP/IP has been reported for high-rate communication between a haptic device running on an embedded controller and a host workstation through a dedicated Ethernet link. However, it is unclear whether such an approach would be applicable to a general Ethernet-based LAN or MAN with several general-purpose workstations that host the haptic devices in a cooperative environment. Obviously, attaining a high-rate real-time communication over the Internet is even more demanding. The development of new protocols for reliable high-packet-rate communication over computer networks suitable for realtime control will likely eliminate existing limitations in future. Nevertheless, in the meantime, control algorithms for cooperative haptics must be devised for improved stability and performance within the confinement of the existing protocols.

In this paper, the effect of a limited packet transmission rate and a relatively small communication delay on the stability and performance of a centralized and a distributed haptic control architecture is investigated. In the centralized framework, the user workstations send the haptic device positions to a server computer which detects collisions, simulates the virtual environment dynamics, and generates force-feedback signals. The results are returned to the local workstations for display to the users. The second approach is based on a distributed control architecture in which each workstation executes a local copy of the VE and simultaneously communicates with other workstations to coordinate the VEs and users. The stability margins of the two multi-rate control architectures as well as their transparency are analyzed and compared. It is shown that the down-sampling and delay in the communication link cause a viscous friction effect that can be compensated by active control. An experimental platform for dextrous dual-user/dual-finger haptic interaction is developed and used to validate the analytical results.

A large body of research in modelling of virtual environments and control of haptic interfaces has been dedicated to single-user applications. These efforts have already yielded

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave. Round-trip delay</td>
<td>2.4 ms</td>
</tr>
<tr>
<td>RMS Jitter</td>
<td>0.49 ms</td>
</tr>
<tr>
<td>Packet loss</td>
<td>2 packets out</td>
</tr>
<tr>
<td></td>
<td>of 2.7 million packets</td>
</tr>
<tr>
<td>Achievable packet rate</td>
<td>128 Hz</td>
</tr>
</tbody>
</table>
numerous haptic rendering techniques and stability results for such applications. Time domain passivity and virtual coupling have been used for stability analysis of haptic control systems (Colgate and Schenkel, 1997; Ryu et al., 2004; Han, 2003). Hayward et al. (2004) have a survey on design components, distinct properties and application of current haptic devices. In Mahvash and Hayward (2003), the passivity theorem is used to design an update strategy for calculating the interaction force between tool and deformable tissue to ensure stability and fidelity of the operation. In Constantinescu et al. (2005), using impulsive and penalty forces, a new approach has been developed for force rendering of rigid contacts to create infinite stiffness upon contact and limited stiffness during contact. Sirousspou et al. (2000) adopt a four-channel teleoperation framework with adaptive damping for stable high-performance haptic rendering.

Multi-rate haptic simulations have been considered in the literature before. In Cuvuoglu and Tendick (2000) and Wang et al. (2005), a local model bridges the discrepancy between high frequency haptic simulation rate and slowly run physical model to ensure a reliable haptic rendering. Also Barbagli et al. (2005) employ the concept of local models for multi-rate haptic rendering of complex deformable objects that can be touched by multiple users. In Cho et al. (2005), a multi-rate wave transformer is adopted to stabilize the zero-order-hold delay in a multi-rate control scheme for haptic simulation. Astley and Hayward (1998) propose a multi-rate scheme to simulate interaction with a deformable tissue using higher mesh update rates in the contact region compared with the remaining coarsely meshed region. d’Aulignac et al. (2000) demonstrate a multi-rate haptic simulation with a mass-spring model and a local model which updates itself based on the past history.

Shared virtual environments are often implemented over the Internet because of its existing infrastructure and global reach. Yoshikawa and Ueda (1996) propose a general structure for providing force-feedback over the Internet without explicitly addressing the network-induced stability and performance degradation. Buttolo et al. (1997) investigate three different implementations for shared haptic environments depending on how the virtual environment is manipulated by the users. In static applications, users can explore but not modify the shared environment. In collaborative haptics, although users can modify the environment by touching and moving objects, they cannot manipulate the same objects simultaneously. Finally, cooperative environments permit simultaneous handling of shared objects by two or more users.

The effect of communication time delay and jitter on the performance of shared haptic virtual environments has been experimentally studied by previous researchers (Hikichi et al., 2002; Wang et al., 2003). In Park and Kenyon (1999), it has been shown that delay jitter has a negative effect on cooperative virtual environments. In Ariouli et al. (2002) and Carignan and Olsson (2004), model-based controllers and wave variable-based techniques have been proposed for delay compensation in multi-user haptic rendering. Jeffay et al. (2001) developed an audio/video media adaptation for transferring data in distributed VEs and experimentally investigated the effect of data loss and delay jitter.

The main contributions of this work can be summarized as: (i) presenting a mathematical framework for the modelling and analysis of cooperative multi-rate haptic control systems for rigid object manipulation; (ii) comparative study of stability and performance of centralized and distributed architectures for cooperative haptics using the developed models; (iii) proposing a method for compensating the undesirable effect of network downsampling on cooperative haptic rendering.

The rest of this paper is organized as follows. The control architectures for cooperative haptics will be given in Section 2. The mathematical modelling of multi-rate cooperative haptic control will be discussed in Section 3. Results of stability analysis for the centralized and distributed control architectures will be presented in Section 4. These will be followed by a transparency analysis in Section 5. In Section 6, a platform for dextrous cooperative haptic simulation will be introduced and experimental results will be presented. The paper will be concluded in Section 7.

2. Control Architectures for Cooperative Haptics

In this paper two controller architectures, namely centralized and distributed are proposed for networked cooperative haptics. Arguments presented here are based on a dual-user configuration but they can be easily extended to multi-user configurations.

2.1. Centralized Cooperative Haptics

In a centralized control architecture, a server workstation that is host to the VE simulator collects and processes the information acquired by all user workstations and returns the calculated interaction forces along with objects and other users’ states for display to the users. Figure 1 illustrates a centralized architecture for a dual-user cooperative haptic environment. Due to the discrete nature of the simulation, the combined network and computation delay is assumed to be a multiple of packet communication period and is lumped in a single round-trip element represented by $z^{-1}$. In most LAN/MAN-based applications $n = 1–2$ samples as the actual network delay is small compared to the packet transmission time interval and the delay is mainly due to the commutation cycles. In simple applications, all elements of the VE simulation can run at the high haptic update rate while the network communication rate is limited. When upsampling, the slow-rate signal is interlaced with constant samples. As will be seen later in the paper, the maximum achievable stiffness in the centralized framework for the users across the network can be fairly restricted due to this limited network packet transmission rate.
has been no analytical work concerning the effect of network delay mitigation in cooperative haptics: e.g., see Shen et al. (2004). However, to the best knowledge of the authors, there is no potential benefit of using a distributed architecture for contact stiffness. Some researchers have previously alluded to the transmission rate and delay restricting the maximum achievable static rigid objects. In the distributed framework, local high-rate feedback loops allow for the rendering of rigid contacts whereas in the centralized framework, the low network packet transmission rate and delay restrict the maximum achievable contact stiffness. Some researchers have previously alluded to the potential benefit of using a distributed architecture for delay mitigation in cooperative haptics: e.g., see Shen et al. (2004). However, to the best knowledge of the authors, there has been no analytical work concerning the effect of network packet transmission rate, delay, and the control architecture on the stability and performance of LAN/MAN-based multi-rate cooperative haptic simulation.

Although realistic haptic simulations often involve multiple degrees of freedom, motion in these axes can be reasonably decoupled, at least in the case of interaction with rigid single-body objects and single-point contacts. Therefore to simplify the modelling and analysis, a single-axis dual-user simulation with linear elements is considered here. The analysis could become involved if coupling among axes, e.g. due to multi-point contact between object and environment, or multi-body objects are considered. However, the results of experiments in Section 7 demonstrate that a single-axis analysis can accurately predict system behavior in multi-axis scenarios.

Mass-spring-damper models of the centralized and decentralized configurations are displayed in Figures 3 and 4, respectively. In these figures, \( m_0 \) and \( m_1 \) are the combined masses of the users and haptic devices, \( m_o = m_o1 = m_o2 \) is the mass of virtual object, \( k \) and \( b \) are the stiffness and damping of corresponding virtual couplers, \( x \) and \( \tilde{x} \) are local and network transmitted positions, and \( f^h_b \) and \( f^d_b \) are users’ exogenous force inputs. The additional virtual couplers between the virtual objects in the distributed controller, represented by \( k_o1 \) and \( k_o2 \) in Figure 4 are intended to prevent position drift between the two copies of the shared object. Note that these systems involve multi-rate discrete-time and continuous-time states due to the discrete nature of the controller, presence of the network element and zero-order-hold (ZOH) circuits, and the continuous-time dynamics of the haptic devices. However, the modelling, analysis and synthesis will be conducted in discrete time. The following two multi-input/multi-output (MIMO) representations of the system dynamics in the discrete-time domain are proposed in this paper.

### 3.1. Subsystem Re-sampling

In this approach, first the continuous-time dynamics of the haptic devices and the virtual object are discretized using a ZOH continuous-to-discrete transformation at their corresponding sample rates (Astrom and Wittenmark, 1997). The system dynamics including those of the controllers are then rearranged into two subsystems operating at sampling rates \( T_i \) and \( T_c \) corresponding to data transmission and control computation rates, respectively (see Figure 5). The difference equations governing the evolution of the states are given by

\[
\begin{align*}
x_i[k_i + 1] &= A_i x_i[k_i] + B_i u_i[k_i] \\
y_i[k_i] &= C_i x_i[k_i] + D_i u_i[k_i], \quad i = r, c
\end{align*}
\]

where the state vectors \( x_i \) contain positions and velocities of the haptic interfaces and the virtual object where applicable. The derivation of the state transition matrices in (1) for the simplified models in Figures 3 and 4 are straightforward and
will not be presented here. The computation/communication delays can simply be incorporated into the state-space models by augmenting the state vectors with the delayed signals. Assuming that the samplers are synchronized and $T_i = NT_c$, the fast discrete system can be resampled at the slower rate $T_t$. This is possible since $u_c[k_c]$ in Figure 5 is constant for $N$ samples between the sampling instants of $y_t[k_t]$. It is straightforward to show that the state transition matrices after resampling are given by

$$\begin{align*}
\tilde{A}_c &= A_c^N, \\
\tilde{B}_c &= A_c^{N-1}B_c + A_c^{N-2}B_c + \cdots + A_cB_c + B_c \\
\tilde{C}_c &= C_c \\
\tilde{D}_c &= D_c.
\end{align*}$$

\(2\)
At this stage, all difference equations describing the evolution of system states have the same sample rate and therefore, the feedback interconnection between the two subsystems can be closed. Despite its simplicity, this method normally is inapplicable to systems with more than two sample rates and/or when one rate is not a multiple of the other. Another drawback of this technique is that the design parameters, e.g. the virtual coupler parameters, cannot be separated from the rest of the system dynamics. Therefore, such a modelling approach has limited utility for control synthesis purposes.

### 3.2. Direct State-Space Representation

To overcome the shortcomings of the previous method, the state-space modelling approach for multi-rate sampled systems given in Araki and Yamamoto (1986) will be utilized here. In the rest of this section, it is demonstrated that how this technique can be employed for deriving the discrete-time model of the multi-rate cooperative haptic control system. The control system diagram is shown in Figure 6. The open-loop continuous-time model of the system, including the dynamics of the users, haptic interfaces and the virtual object, is given by

$$
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t)
\end{align*}
$$

where $x(t) \in \mathbb{R}^n$ is the vector of system states, i.e. positions and velocities, $u(t) = (u_t(t), u_c(t))^T$ is the vector of inputs and $y(t) = (y_t(t), y_c(t))^T$ is the vector of measurements. As before, subscripts $t$ and $c$ refer to the network and control sampling rates. It is assumed that all sample times are synchronized and are in the form of $T_i = T_{icm}/N_i$ where $N_i$’s are positive integers whose greatest common measure is one. As such, the sample times can be written as multiples of a base sample time $t_{icm}$, i.e. $T_i = l_i t_{icm}$, with $t_{icm} = T_{icm}/N_{icm}, l_i = N_{icm}/N_i$, and $N_{icm}$ is the least common multiple of $N_i$.

For the discrete-time realization of the system, an augmented state vector is defined as

$$
\begin{pmatrix}
x((k-1)T_{icm} + t_{icm}) \\
\vdots \\
x((k-1)T_{icm} + (N_{icm} - 1)t_{icm}) \\
x(kT_{icm})
\end{pmatrix}
$$

The augmented output vector $y_D$ is

$$
y_D[k] = \begin{pmatrix} y_{D_t}[k] \\ y_{D_c}[k] \end{pmatrix}
$$

where

$$
y_{D_i}[k] = \begin{pmatrix} y_t(kT_{icm}) \\ y_t(kT_{icm} + T_i) \\ \vdots \\ y_t(kT_{icm} + (N_i - 1)T_i) \end{pmatrix}
$$
The augmented input vector \( u_D \) can be defined similarly. Using the above definitions, it can be shown that (Araki and Yamamoto, 1986)

\[
\begin{align*}
\dot{x}_D[k+1] &= A_D x_D[k] + B_D u_D[k] \\
y_D[k] &= C_D \left[ U_1 x_D[k+1] + U_2 x_D[k] \right]
\end{align*}
\] (7)

The expressions for \( A_D, B_D, \) and \( C_D \) are given in Appendix A and \( U_1, U_2 \) are block diagonal matrices as follows:

\[
\begin{align*}
U_1 &= \text{diag}(I_{n_0}, I_{n_0}, \ldots, I_{n_0}, 0) \\
U_2 &= \text{diag}(0, 0, \ldots, 0, I_{n_0})
\end{align*}
\] (8)

and \( n_0 \) is the dimension of the original state vector \( x(t) \) in (3).

By replacing \( x_D[k+1] \) in second line of (8) from the first line of (7), one may obtain a standard state-space representation as follows:

\[
\begin{align*}
\dot{x}_D[k+1] &= A_D x_D[k] + B_D u_D[k] \\
y_D[k] &= \hat{C}_D x_D[k] + \hat{D}_D u_D[k]
\end{align*}
\] (9)

where \( \hat{C}_D = C_D U_1 A_D + C_D U_2 \) and \( \hat{D}_D = C_D U_1 B_D \).

The delay elements associated with computation and data transmission in Figure 2 can be incorporated into the discrete-time model by augmenting the state vector with the delayed input signals. The reader is referred to the Appendix B for the derivation of the new state transition matrices, \( \hat{A}, \hat{B}, \hat{C}, \) and \( \hat{D} \), as well as the new augmented input \( \tilde{u}_D \).

Once the open-loop discrete-time difference equations are obtained, the closed-loop dynamics can be formed using the feedback law

\[
\tilde{u}_D = F_D * y_D
\] (10)

where \( F_D \), given in Araki and Yamamoto (1986), is the feedback gain matrix whose elements are constant and consist of the stiffness and damping parameters of all virtual couplers present in the system. Finally, the closed-loop space transition matrix \( A_D' \) can be computed as

\[
A_D' = \hat{A}_D + \hat{B}_D F_D (I - \hat{D}_D F_D)^{-1} \hat{C}_D.
\] (11)

The closed-loop system is stable if and only if all eigenvalues of this matrix lie inside the unit circle.

4. Stability Analysis

In this section, the stability margins of the centralized and distributed control architectures are compared with respect to changes in the stiffness parameters of the virtual couplers. The discrete-time models of multi-rate cooperative haptic controllers have been developed using the techniques introduced in Section 4. The values of constant parameters in all scenarios are \( m_1^h = m_2^h = 0.1 \) kg, \( m_o = m_{o1} = m_{o2} = 0.4 \) kg, and \( b=10 \) N.s/m for all couplers in both configurations, \( T_1 = 1/128 \) s and \( T_c = 1/1024 \) s. The combined communication/computation delays of \( n = 0, 1, 2, 3 \) network sample times have been considered in the analysis. It should be noted that given the results of Table 1, under normal LAN/MAN conditions the combined delay value is expected to be in the range of \( n = 1–2 \) network sample times.

4.1. Centralized Controller

Figure 7 illustrates the stability region of the centralized architecture, when all system parameter except \( k_1 \) and \( k_2 \) in Figure 3 are fixed. The marginal values of the parameters determine the maximum stiffness that can be presented to each user. As seen in Figure 7 for \( n = 0 \), the limited network packet rate has contributed to a significant reduction in the margin of stability with respect to \( k_2 \), the coupling stiffness for the user across the network. As expected the marginal stiffness value for the user across the network further decreases as the \( n \) increases whereas the local user is unaffected by the communication delay.

4.2. Distributed Controller

The stability analysis was carried out for the distributed control architecture as well. In Figure 8, the stable region with
Fig. 7. The region of stability for the centralized control architecture for different communication channel sample delays.

Fig. 8. The region of stability for the distributed control architecture when \( k_{11}/4 k_{22}/4 \) 1000 N/m are fixed for different communication channel sample delays. Respect to the parameters \( k_{11} = k_{22} \) and \( k_{12} = k_{21} \) in Figure 4 is plotted while \( k_{01} = k_{02} = \) 1000 N/m are constant. Note that the stability region is noticeably enlarged when compared to that of the centralized architecture in Figure 7. To study the effect of the coordinating virtual couplers between the objects, \( k_{12} = k_{21} = \) 300 N/m were fixed and \( k_{11} = k_{22} \) and \( k_{01} = k_{02} \) were varied with the results given in Figure 9. Alternatively, Figure 10 demonstrates the stable region for the case in which \( k_{11} = k_{22} = \) 2000 N/m are constant while \( k_{01} = k_{02} \) and \( k_{12} = k_{21} \) are changed. It should be noted that the communication delay reduces the marginal values of \( k_{0} \) and \( k_{12} \) as expected.

Fig. 9. The region of stability for the distributed control architecture when \( k_{12} = k_{21} = 300 \) N/m are fixed for different communication channel sample delays.

Fig. 10. The region of stability for the distributed control architecture when \( k_{11} = k_{22} = 2000 \) N/m are fixed for different communication channel sample delays.
Fig. 11. Model of haptic interaction with $k_w = 30,000$ N/m and $b_w = 20$ N·s/m in hard contact and $b_w = k_w = 0$ in free motion: (a) local user in centralized controller; (b) remote user in centralized controller; (c) distributed controller; (d) the ideal system.

Table 2. The parameters used for performance analysis of the controllers in free motion.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>1000 N/m</td>
</tr>
<tr>
<td>$k_{o2}$</td>
<td>$k_{o1} = 300$ N/m</td>
</tr>
<tr>
<td>$b_1$</td>
<td>15 N·s/m</td>
</tr>
<tr>
<td>$b_{o2}$</td>
<td>$b_{o1} = 10$ N·s/m</td>
</tr>
<tr>
<td>$k_{11}$</td>
<td>$k_{22} = 3000$ N/m</td>
</tr>
<tr>
<td>$b_{22}$</td>
<td>$b_{11} = 20$ N·s/m</td>
</tr>
<tr>
<td>$T_c$</td>
<td>$1/1024$ s</td>
</tr>
<tr>
<td>$m_1$</td>
<td>$m_2 = 0.4$ kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>$m_1 = m_2 = 0.1$ kg</td>
</tr>
</tbody>
</table>

The results of the previous analysis demonstrate that the distributed cooperative haptic architecture is capable of rendering rigid contacts under typical delays and network packet rates expected in a LAN or MAN whereas the centralized controller can easily become unstable under such circumstances, unless the coupling stiffness of the network user is considerably reduced.

5. Performance Analysis

In this section the performance of the centralized and distributed architectures are studied. To this end, it is assumed that the users manipulate a virtual rigid object in free motion or in contact with a rigid wall. In Figure 11, the mass-spring-damper models of the system under different scenarios are depicted. In hard contact, the value of $k_w$ is relatively large whereas in free motion $k_w$ and $b_w$ are set to zero. In the succeeding analysis for the centralized controller, the users on the server and remote workstations are denoted as local and remote user, respectively. The decentralized controller is symmetric with respect to the two users and therefore, only one user is considered in its analysis. In an ideal case, the user should feel that he/she is interacting with a pure mass. To compare the controllers, the perceived admittances by the users in each architecture are compared with that of a mass in Figure 11(d) for a network delay of $\tau = 1$ in Figures 1 and 2.

5.1. Free Motion

In this case it is assumed that one user is manipulating the virtual object in free motion while the second user input force is set to zero. The perceived admittance of the object is defined as the ratio of the output velocity $v^i$ to the input force $f^i$ in Figure 11. Table 2 contains the parameters used in this case where the subscribe $o$ represent the moving object virtual coupler.
Fig. 12. Object perceived admittance $V_h/j\omega$ in free motion.

The frequency responses are shown in Figure 12 from which it is clear that the local user in the centralized architecture observes the closest admittance to that of the virtual mass, compared with the remote centralized and the distributed users. This should not be surprising since the network element is absent from the local user control loop and hence a high-rate feedback loop can be implemented. It should be noted, however, that even in this case at higher frequencies the effect of spring-damper coupler becomes dominant and the response deviates from that of a mass as seen in Figure 12(b). The network low packet rate and delay cause the largest deviation from the ideal response at medium to high frequencies in the perceived admittance of the remote user in the centralized controller.

At low frequencies, a dominant viscous behavior is observed in the responses of the distributed controller and the remote user in the centralized controller. The amount of this damping can be analytically calculated as a function of system parameters using the discrete-time multi-rate modelling techniques introduced in Section 3 and the mass-spring-damper models in Figure 11. This is achieved by finding the limit of the corresponding transfer functions as $z$ approaches one. However, a drawback of this approach is that for the rotational degrees of motion, the resulting damping values depend on hand-object contact point which is unknown and can change. To overcome this problem, a simplified model is used in which the hand-object virtual couplers in Figure 11 are removed and the hand forces are sampled with control and network rates and are directly applied to the masses. Using this approach for the distributed controller with a control to network sampling ratio of $N = 2$, i.e. $T_c = 2T_n$, and a delay-free communication channel, e.g. $n = 0$, one may obtain

$$b_{dist} = 2m_o k_o \left( 4\frac{m_o}{T_c} - T_c k_o - 2b_o \right)^{-1}. \quad (12)$$

The damping expressions obtained with the original models are rather long and are given in Appendix C. The analytical expressions of the delay in the case of $N > 2$ or when the network delay $n$ is nonzero are complex and will not be given for brevity. In Table 3, the damping values obtained from the two approaches using the parameters in Table 2 are compared. Although the simplified method somewhat underestimates the damping, it can still provide a reasonable estimate for the rotational axes of the motion. Obviously, for the linear axes of the motion one can employ the original values.

The viscous behavior at low frequencies can be attributed to the network zero-order-hold effect. To explain this, the case of the remote user in the centralized architecture moving a virtual mass at a constant velocity is considered. In the absence of sampling and using a continuous-time controller, at steady state, the virtual object would track the haptic device
with zero error at a constant velocity. Accordingly, the coupler force settles to zero and the user would not feel any resistance force, as predicted by the physics of the problem. However, in practice, the haptic device positions are sampled at the network rate, and are transmitted to the simulation workstation. The simulation rate is usually much higher than the network transmission rate and, hence, the low-rate haptic position acts as a step-wise position reference command for the virtual object with a proportional-derivative controller. This causes a transient response between the network sample times. In the steady state, the virtual object position always lags behind the haptic device position at the network sampling times by $\tau_d$, yielding a non-zero position tracking error. The value of $\tau_d$ depends on factors such as the virtual object dynamics, the network sampling rate, and the control sampling rate. Consequently, the coupling force reflected to the user calculated at the network sample times can be approximated by $f_2 \approx k_x (x^{lib}_2 (kT_d) - x^{lib}_1 (kT_d - \tau_d)) \approx ak_x x^{lib}_1 (kT_d)$ causing a viscous damping effect. A similar argument can be made about the distributed architecture.

To compensate the undesirable effect of the viscous-type friction on user’s perception of the object, it is proposed that a negative damping be added to the object dynamics. The damping computations can be performed off-line for every movable single-body object in the virtual environment using the modelling techniques presented in this paper. If the simplified model is used, the underestimated damping value would only depend on the virtual object parameters and control coupling gains which are precisely known. In the calculations based on the original model, the haptic device mass is also needed. This can be estimated with good accuracy in most cases. Since the communication/computation delay in the intended applications is usually around $n = 1–2$ network sample times, using $n = 0$ in the calculations would yield an underestimated damping value. Therefore, it is always possible to avoid a potential instability as a result of a net negative damping at the expense of an imperfect compensation. In addition, the haptic device has some damping which has been ignored in the analysis, hence the system stability is maintained in practice.

Figure 13 shows a significant improvement in the system response after introducing the compensator $b_o = -3$ N·s/m, obtained using $n = 1$, in the distributed architecture. The perceived admittance is now close to that of the virtual mass for a wide frequency range.

### 5.2. Rigid Contact

As shown in Section 4, the distributed architecture has significantly higher marginal values for the stiffness couplers compared to those of the centralized controller. In contacts with rigid environments, the stiffness seen by the user can be approximated by the virtual coupler stiffness.

![Bode Diagram](image)

**Fig. 14.** The perceived environment admittance $\frac{x^h}{F^h}(j\omega)$ when the box is in contact with a rigid wall.

Figure 14 displays the perceived admittance $\frac{x^h}{F^h}(j\omega)$ in the frequency domain when the user is pushing the object against a rigid wall. The controller parameters are the same as those in free motion and the wall stiffness and damping are $k_w = 30,000$ N/m, and $b_w = 20$ N·s/m, respectively. It turns out that the wall stiffness can be chosen substantially larger than the finger-box coupling stiffness. This can be explained by the fact that in the box-wall interaction, wall is a static object which a fixed position in both copies of the virtual environment. However, in finger-box interaction, the combined finger/haptic device mass introduces an extra dynamic mode in the system. As finger position is measured, transmitted over the network, and used in feedback control, this mode is subject...
to a low-rate network feedback loop which can limit the coupler stiffness. As expected, at low frequencies the magnitude of response is close to $\frac{1}{5}k_{11}$ and $\frac{1}{5}k_{11}$ for the remote, distributed and ideal system, respectively. This demonstrates a sharp contrast between the centralized and distributed controller in rendering rigid contacts where the achievable stiffness by the centralized architecture is quite limited.

In summary, it can be concluded that the distributed controller can achieve a superior performance over the centralized controller for manipulating objects in free motion and in rigid contact.

6. Experimental Results

Comparative experiments have been conducted with a dual-user/dual-finger haptic simulation platform. A description of the experimental setup and a discussion of the experimental results will be given in the following sections.

6.1. Experimental Setup for Two-finger Haptic Interaction in Co-operative Environments

The experimental setup shown in Figure 15 consists of two haptic interfaces with grasping capability, centralized and decentralized control architectures for VE simulation, as well as graphical displays. Using this system, two users can grasp and cooperatively manipulate a virtual box in a plane, i.e. by moving it in $x$ and $y$ directions and rotating it around the $z$ axis and making contact with four rigid walls. The elements of the cooperative haptic platform are briefly described in the rest of this section.

6.1.1. Haptic device

Two identical planar pantograph mechanisms obtained by modifying a Quanser planar haptic device constitute a two-finger haptic interface. These parallel manipulators are capable of measuring the endpoint position and producing force-feedback in the $x$–$y$ plane using rotary optical encoders and direct-drive DC motors. They are powered by two QPA linear current amplifiers from Quanser Inc. A total of four pantograph mechanisms enable dual-user/dual-finger cooperative manipulation of virtual objects.

6.1.2. Virtual environment simulator

The VE engine is responsible for simulating the dynamics of the virtual box and generating model-based reaction forces that will be displayed to the users through the graphic consoles and haptic interfaces. The VE engine consists of three subsystems described below:

- Collision detection: The collision detection (CD) routine is responsible for detecting any potential collision between the users’ fingers and the surface of the virtual box. The finger position is calculated using encoder readings and the pantograph kinematics and the object position is generated by the VE dynamics.

- Force calculation: A penalty-based approach (Constantinescu et al., 2005; Salcudean and Vlaar, 1997) is adopted to calculate the interaction force between the user finger and the virtual box in which spring-damper virtual couplers convert the penetration vector into a reaction force. In order to simulate frictional forces a similar approach to that of Melder and Harwin (2003) has been adopted. Melder and Harwin (2003) formulated the friction and surface forces using friction cone and god object concepts.

- Virtual environment dynamics: In general, the virtual world can consist of static and dynamic objects. The Euler integration routine with a fixed-step size of $1/1024$ s is invoked to simulate the dynamics of moving objects, the virtual box in this case, based on the calculated user/object interaction forces.

6.1.3. Graphic display

Matlab’s Virtual Reality toolbox has been employed for the graphics. The update rate is set to 32Hz which is sufficient for producing smooth motions on the displays.

Table 4. Control parameters in centralized architecture used in experiments.

<table>
<thead>
<tr>
<th>$k_N$</th>
<th>$b_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{N1} = k_{N2} = 800$ N/m</td>
<td>$b_{N1} = b_{N2} = 3$ N·s/m</td>
</tr>
<tr>
<td>$k_T$</td>
<td>$b_T$</td>
</tr>
<tr>
<td>$k_{T1} = k_{T2} = 1200$ N/m</td>
<td>$b_{T1} = b_{T2} = 10$ N·s/m</td>
</tr>
</tbody>
</table>
has to exert significant damping in order to barely grasp the object and move it in the plane. The instability in the centralized framework can be avoided by reducing the stiffness of the virtual couplers for the user across the network. However, this would degrade the haptic sensation, specially in rigid contact.

6.3. Performance

Experiments have been conducted to investigate the effect of low network transmission rate on the users’ perceived impedances. To apply consistent forces in different experiments, the user forces are emulated through the control signals. One of the pantograph mechanisms is moved along a sinusoidal path with an amplitude of 0.05 m and a frequency of 2 rad/s in the $y$ direction using a proportional-derivative controller, while a constant force was applied to the second pantograph along the same direction. With this arrangement, the box was grasped and moved along the $y$ direction by the two pantographs. Figure 17(a) shows a snapshot from graphical display of this experiment.

The control parameters are the same as those in the previous case. Ideally, the sum of the forces applied on the virtual object must be equal to the inertial force required for moving the box along the sinusoidal path.

In Figure 18, the local user force profile in the centralized architecture is compared with that of the user in the distributed architecture. When uncompensated, the user in the distributed system has to apply a noticeably larger force in order to generate the same motion. This is consistent with the analytical result that had predicted an extra viscous damping in the system response due to the network limited packet rate. As can be seen in Figure 18, a negative damping compensation can significantly improve the response. The values of the damping were 0.88 N·s/m for the linear axes of motion, and 0.025 N·s·m/rad for the rotational motion, all chosen based on the results of analysis. In the experiments, the users observed a noticeable improvement in the system response after the active damping compensation.

To compare the control architectures in rigid contact, the virtual box is pushed against a stiff wall with parameters $k_{NW} = 4000$ N/m, $k_{Tw} = 7000$ N/m, $b_{NW} = 30$ N·s/m, $b_{Tw} = 50$ N·s/m, using an emulated user force of $f_y = 1.5 \sin(3t)$ N, (see Figure 17(b)). To avoid instability, the remote user parameters are set to $k_N = 170$ N/m, $b_N = 3$ N·s/m.

### 6.1.5. Network communication

Due to its small overhead, the UDP protocol is used to communicate position data between the two user workstations that are located on a LAN with the characteristic given in Table 1. The data transmission rate is set to 128 Hz, a maximum reliable frequency that was attainable in the experiments under a normal network traffic.

6.2. Stability

The distributed and centralized architectures have been implemented on two machines running the VxWorks RTOS. The control parameters were set to values shown in Tables 4 and 5 in the case of centralized and distributed architectures, respectively. In these tables, $N$ and $T$ subscripts denote the normal and tangential directions along the contact between the finger and box, the indices $x, y$ and $R$ represent the $x, y$ and rotation coordinates of the box, and the subscript $o$ denotes the moving object virtual coupler. The sampling times were set to $T_i = 1/128$ s, $T_c = 1/1024$ s. Coulomb friction with coefficients of 0.05 and 0.15 in the linear and rotational degrees of freedom was added between the box and ground. The coefficient of Coulomb friction between the fingers and the box was set to 0.6.

The centralized controller with the given parameters is essentially unstable as predicted by the results of analysis in Figure 7. In the experiments, while the users could cooperatively grasp and move the virtual box under the distributed controller, such operation was almost impossible with the centralized controller due to its instability. This is evident from the force and position profiles in Figure 16 where the user has to exert significant damping in order to barely grasp the object and move it under the distributed system has to apply a noticeably larger force in order to generate the same motion. The values of the damping were 0.88 N·s/m for the linear axes of motion, and 0.025 N·s·m/rad for the rotational motion, all chosen based on the results of analysis. In the experiments, the users observed a noticeable improvement in the system response after the active damping compensation.

### Table 5. Control parameters in distributed architecture used in experiments.

| $k_{N11} = k_{N22} = k_{N12} = k_{N21}$ | $b_{N11} = b_{N22} = b_{N12} = b_{N21} = 3$ N·s/m |
| $k_{T11} = k_{T22} = k_{T12} = k_{T21} = 1200$ N/m | $b_{T11} = b_{T22} = b_{T12} = b_{T21} = 10$ N·s/m |
| $k_{o1x} = k_{o1y} = k_{o2x} = k_{o2y} = 400$ N/m | $b_{o1x} = b_{o1y} = b_{o2x} = b_{o2y} = 8$ N·s/m |
| $k_{o1R} = k_{o2R} = 10$ N·m/rad | $b_{o1R} = b_{o2R} = 0.1$ N·s/rad |
Fig. 16. Comparison of distributed and centralized architectures in experiment: (a) finger-box interaction force; (b) box’s path on the $x$–$y$ plane.

Fig. 17. The virtual environment in performance evaluation experiments: (a) free motion; (b) rigid contact; cylinders represent the user fingers.
$k_T = 270 \text{ N/m, } b_T = 5 \text{ N·s/m.}$ The resulting haptic device displacements are plotted in Figure 19. The user in the distributed controller and the local user in the centralized controller perceive a much stiffer contact, as is evident from their smaller penetration in the virtual wall.

7. Conclusions

Networked collaborative haptic environments present new challenges to the designers of haptic-enabled virtual reality systems. These are mainly due to nondeterministic network...
constraints such as limited packet transmission rate, latency, data loss and jitter. In this paper, the emphasis was on cooperative haptic simulation over LANs or MANs where the communication link can be characterized by its limited packet rate and a few \( n = 0 \rightarrow 2 \) combined network/computation sample delays, and where jitter and packet loss may be ignored.

Using mathematical descriptions for multi-rate \( MIMO \) control systems, the stability and performance characteristics of two control architectures, namely centralized and distributed were compared. Analytical results as well as experiments conducted with a dual-user/dual-finger haptic platform demonstrated that the distributed controller possesses noticeably larger stability margins. Also, it was shown that the network limited packet rate and delay can cause a viscous friction effect in the perceived impedance of the virtual object. This was eliminated by adding a negative damping to the virtual object. It was demonstrated that, when properly compensated, the distributed controller can provide higher haptic fidelity in free motion and in contact with rigid environments. Also, it was shown that the network delay can be characterized by its limited packet rate, latency, and where jitter and packet loss may be ignored.

In future, the impact of longer time delays, delay jitter and packet loss on cooperative haptic simulation over WANs will be investigated and methods for improving stability and performance under such conditions will be sought.

Acknowledgements

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Centres of Excellence (OCE) for supporting this research.

Appendix A: State Transition Matrices for Discrete-time Multi-rate Dynamics with ZOH

Araki and Yamamoto (1986) contains a detailed derivation of the state transition matrices in (8). The final forms of the matrices are given here for the ease of reference. \( A_D \) is composed of \( N_{lcm} \times N_{lcm} \) matrix blocks of the form

\[
A_D = \begin{pmatrix}
0 & \cdots & 0 & A_{D1} \\
\vdots & & \vdots & \vdots \\
0 & \cdots & 0 & A_{DN_0}
\end{pmatrix}
\]

where \( A_{Dl} \) is given by

\[
A_{Dl} = e^{A_{lr}t_0}, \quad l = 1, \ldots, N_{lcm}.
\]

\( B_D \) is composed of \( 1 \times p \) matrix blocks of the form \( p \) is the number of sample times, i.e. \( p = 2 \) in the case of this paper

\[
B_D = \begin{pmatrix}
B_{D1} & B_{D2} & \cdots & B_{Dp}
\end{pmatrix}
\]

whose blocks are \( N_{lcm} \times N_{l} \) block matrices

\[
B_{Di} = [b_{Di,ju}], \quad l = 1, \ldots, N_{lcm};
\]

\[
\mu = 0, \ldots, N_{i} - 1.
\]

The block \( b_{Di,ju} \) of \( B_{Di} \) is an \( n_{a} \)-dimensional column vector

\[
b_{Di,ju} = \begin{cases}
0, & l \leq \mu_i \\
\int_{\mu_i}^{\mu_i+1} e^{A_{l}t_{jcm} - \tau} b_{di} d\tau, & \mu_i < l \leq (\mu_i + 1)\mu_i \\
\int_{\mu_i+1}^{\mu_i+2} e^{A_{l}t_{jcm} - \tau} b_{di} d\tau, & (\mu_i + 1)\mu_i < l.
\end{cases}
\]

\( C_D \) is a \( p \times 1 \) block matrix of the form

\[
\begin{pmatrix}
C_{D1} \\
\vdots \\
C_{Dp}
\end{pmatrix}
\]

whose blocks are \( N_{j} \times N_{lcm} \) block matrices

\[
C_{Dj} = [(c_{Dj,vl})^T], \quad v = 0, \ldots, N_{j} - 1;
\]

\[
l = 1, \ldots, N_{lcm}.
\]

The block \( (c_{Dj,vl})^T \) is an \( n_{a} \)-dimensional row vector given by

\[
(c_{Dj,vl})^T = \begin{cases}
c_j, & v = 0, l = N_{lcm} \text{ or } vl_l = l \\
0, & \text{otherwise}.
\end{cases}
\]

\( F_D \) is a \( p \times p \) block matrix whose \( (i - j) \)th block \( F_{Di,j} \) is the \( N_{j} \) by \( N_{j} \) matrix given by

\[
F_{Di,j} = (f_{Di,j,vl}), \quad \mu = 0, \ldots, N_{i} - 1;
\]

\[
v = 0, \ldots, N_{j} - 1.
\]

\[
f_{Di,j,vl} = \begin{cases}
f_{ij}, & vl_l \leq \mu_i < (v + 1)l_j \\
0, & \text{otherwise}.
\end{cases}
\]

Appendix B: Delay Augmentation

The network and computational delay in Figure 2 can be incorporated into the discrete-time dynamics of the systems. Assuming that the round-trip network delay is one sample, i.e. \( n = 1 \), the state vector is augmented with the delayed inputs and the transition matrices are modified as follows:
Appendix C: Viscous Damping Calculation using the Original Model

The damping values in the case of $N = 2$ and $n = 0$ for the single-axis model of Figure 11 have been derived using Matlab’s Symbolic toolbox. For the centralized controller:

$$b_{\text{cent}} = \frac{k_2}{2/T_c - b_2/m_o}$$  \hspace{1cm} (24)

and for the distributed controller:

$$b_{\text{dist}} = \frac{NUM}{DEN}$$  \hspace{1cm} (25)

where

$$NUM = T_c k_{11} (10m_1 k_2^2 b_{11} - 24T_c k_0^2 m_1 m_o)$$

$$- 40m_1 T_c m_o k_0^2 b_o - 24m_1 T_c m_o k_0^2 b_{12}$$

$$+ T_c^3 k_0^2 b_{11} - 24T_c b_{11} m_o k_0^2 - 4T_c^3 b_{11} m_o k_0^2$$

$$+ 10m_1 T_c^2 b_o k_0^2 b_{12} + 8m_1 T_c^2 b_o k_0^2$$

$$- 4T_c^3 b_{11} m_o k_o^2 b_{12} + 10T_c^2 b_{11} m_o k_o^2 b_{11}$$

$$+ 12m_1 T_c^2 b_o k_o^2 b_{11} + 8m_1 T_c^2 b_o k_o^2$$

$$+ 20T_c^3 b_{11} m_o k_o^2 b_{11} - 2T_c^2 k_0^2 m_o b_{12}$$

$$+ 2T_c^2 k_0^2 m_o b_{11} - 2T_c^2 k_0^2 m_o b_{12}$$

$$+ 4T_c k_0^2 m_o b_{12} + T_c^2 k_0^2 b_{11}$$

$$+ 8T_c^2 m_1 b_{11} k_o b_{12} - 40T_c m_o k_o b_{12}$$

$$- 4T_c^3 k_o b_o k_{12} + 8T_c^2 m_1 b_o k_o k_{12}$$

$$+ 12m_1 T_c^2 b_o k_o b_{12} + 12m_1 T_c^2 b_o k_o b_{12}$$

$$+ 20T_c^2 M_o k_o b_{12} + 12T_c^2 m_o k_{12} b_{11}$$

$$- 6T_c k_1 b_{12} b_{11} + 24T_c^2 m_o T_c b_{12}$$

$$+ 48m_1 m_o k_o^2 b_{12} - 24T_c^2 m_o k_o^2 b_{11}$$

$$- 24m_1 m_o T_c b_{11} k_{12}$$  \hspace{1cm} (26)

and

$$DEN = (T_c b_o + T_c b_{12} - 2m_o)(k_{31} k_{12} + k_o k_{12} + k_{11} k_o)$$

$$\times (-T_c^3 b_{11} k_o - 2T_c^2 b_{11} b_o + 4m_1 T_c b_o)$$

$$+ 4m_1 T_c b_{11} + 4m_o T_c b_{11} - 8m_1 m_o).$$  \hspace{1cm} (27)

References


