Improved Haptic Fidelity Via Reduced Sampling Period With an FPGA-Based Real-Time Hardware Platform

Marcia K. O'Malley
e-mail: omalleym@rice.edu

Kevin S. Sevcik

Department of Mechanical Engineering and
Materials Science,
Rice University,
Houston, TX 77005

Emilie Kopp
National Instruments,
11500 N Mopac Expwy,
Austin, TX 78759-3504
e-mail: emilie.kopp@ni.com

A haptic virtual environment is considered to be high-fidelity when the environment is perceived by the user to be realistic. For environments featuring rigid objects, perception of a high degree of realism often occurs when the free space of the simulated environment feels free and when surfaces intended to be rigid are perceived as such. Because virtual surfaces (often called virtual walls) are typically modeled as simple unilateral springs, the rigidity of the virtual surface depends on the stiffness of the spring model. For impedance-based haptic interfaces, the stiffness of the virtual surface is limited by the damping and friction inherent in the device, the sampling rate of the control loop, and the quantization of sensor data. If stiffnesses greater than the limit for a particular device are exceeded, the interaction between the human user and the virtual surface via the haptic device becomes nonpassive. We propose a computational platform that increases the sampling rate of the system, thereby increasing the maximum achievable virtual surface stiffness, and subsequently the fidelity of the rendered virtual surfaces. We describe the modification of a PHANToM Premium 1.0 commercial haptic interface to enable computation by a real-time operating system (RTOS) that utilizes a field programmable gate array (FPGA) for data acquisition between the haptic interface hardware and computer. Furthermore, we explore the performance of the FPGA serving as a standalone system for communication and computation. The RTOS system enables a sampling rate for the PHANToM that is 20 times greater than that achieved using the "out of the box" commercial hardware system, increasing the maximum achievable surface stiffness twofold. The FPGA platform enables sampling rates of up to 400 times greater, and stiffnesses over 6 times greater than those achieved with the commercial system. The proposed computational platforms will enable faster sampling rates for any haptic device, thereby improving the fidelity of virtual environments. [DOI: 10.1115/1.3072904]

1 Introduction

This paper proposes a modification of the PHANToM haptic interface's computational platform to achieve increased sampling rates and high-fidelity stiff virtual walls. Virtual walls in a haptic environment must feel rigid if the simulation is to be high-fidelity or closely matched to reality. Therefore, virtual surfaces, which comprise the simulated environment, are typically modeled as a simple unilateral spring with large stiffness, so as to appear rigid when contact between the user and the virtual wall occurs. Because haptic devices are sampled-data systems, the spring model is discretized when implemented in a haptic rendering algorithm. The discrete nature of the virtual wall acts to generate energy in the simulated system, failing to capture the inherently passive and dissipative nature of a physical spring of identical stiffness [1]. The generated energy, if not dissipated by some means, results in chatter between the stylus of the haptic device and the virtual surface, thereby degrading the realism and fidelity of the simulation.

In actuality, when modest stiffnesses are rendered, the energy generated by the virtual wall is sufficiently dissipated by intrinsic device friction and thus remains transparent to the human operator. Specifically, when a virtual wall model is implemented on a particular haptic interface, one can guarantee the system to be passive if and only if the virtual spring stiffness to be displayed does not exceed the minimum of two ratios, shown in Eq. (1), where b is the inherent device damping, T is the sampling period, f_c is the device's inherent Coulomb friction, and Δ is the sensor quantization [1,2]. The relationship among the device damping, sampling rate, and the maximum achievable stiffness was defined by Colgate et al. [1]. Later, Abbott and Okamura [2] refined the relationship to account for a haptic device's inherent friction and the quantization of the position sensor data.

$$K \leq \min\left(\frac{2b}{T}, \frac{2f_c}{\Delta}\right) \tag{1}$$

Kazerooni [3] initially recognized the existence of energy generation in haptic virtual environments due to the sampled-data implementation of virtual walls. Gillespie and Cutkosky [4] described in detail the introduction of energy into the virtual wall interaction due to the switching nature of the virtual wall model, and the possibility that collision with the virtual surface can occur between samples. The work of Abbot and Okamura provides evidence that the quantized position signal necessary for impedance-based rendering can contribute to the nonpassive nature of a virtual wall [2]. Diolaiti et al. [5] also identified the contribution of position signal quantization, but note the role of the discretization of the position signal, in addition to quantization, in the generation of energy during virtual wall interactions.

Clearly, passive interactions between the user holding the haptic device stylus and virtual walls are desirable for the purpose of guaranteeing user safety and achieving high simulation fidelity. The objective of ensuring passivity in haptic interactions has been

Contributed by the Engineering Simulation and Visualization Committee for publication in the Journal of Computing and Information Science in Engineering. Manuscript received November 6, 2007; final manuscript received December 19, 2008; published online February 9, 2009. Guest Editors: J. Oliver, M. O'Malley, and K. Kesavadas.

approached through a number of means, some of which address the relationship among stiffness, device damping and friction, sampling period, and sensor quantization as given by Eq. (1). For example, a conventional approach is to increase physical dissipation in the haptic device, thus enabling higher virtual wall stiffnesses to be rendered, as proposed by Miller et al. [6]. They defined a function to determine the amount of inherent damping a haptic device should posses to sufficiently dissipate unwanted energy generation. However, excessive damping can compromise overall transparency, making free space no longer feel "free." Colgate and Brown [7] suggested implementing negative virtual damping to achieve increased virtual wall stiffness. They admitted, however, that this solution relies on exact cancellation of device friction with negative virtual damping. This precision is difficult to achieve, in practice, and even then, the system is borderline passive.

An alternative solution to ensure passive interactions between the user and virtual wall is to monitor the energy flow in the system and to adjust the virtual wall model within the simulation software. Adams and Hannaford [8] dissipated unwanted energy through the process of "virtual coupling," a scheme that modifies the simulated environment by means of a filtering mechanism to ensure that the passivity of the system is maintained. Hannaford and Ryu [9] took a similar approach of preserving system passivity, deploying an online passivity observer, which appropriately prompts a passivity controller to adjust the parameters of an adaptive and dissipative rendering element.

Instead of compensating for energy generation with physical or virtual dissipation, a more direct method of extending the passivity region of a virtual spring is to minimize the energy generation stemming from the disparity between the continuous spring model and its discrete sampled-data implementation. If we assume that we will not modify the haptic interface's mechanical hardware (since the mechanical design of most impedance-based haptic interfaces is optimized with respect to friction, damping, and sensor resolution), we leave the b, f_c, and Δ terms unmodified in Eq. (1). Still, we are left with the option of decreasing the sampling period T (an approach that is equivalent to increasing the system's sampling rate) in order to minimize the amount of energy generated by the sampled-data system and maximize the achievable virtual surface stiffness in our haptic virtual environment.

Several means for increasing sampling rates have been explored. One solution is to simplify the rendering algorithm to reduce its computational cost [10], although this almost always sacrifices virtual scene complexity. Mahvash and Hayward [11] combined this method with a consideration of passivity, thus achieving an increase in sampling rate for complex virtual environments.

Many haptic researchers have opted to streamline communication methods with a force-reflecting device by employing a RTOS, a type of event-driven OS that abides by a strict hierarchy of task execution [5,12,13]. This approach has not yet been explicitly employed to upgrade the performance of passive virtual walls by enabling faster loop rates, thereby decreasing the sampling period. Still, RTOS have benefitted sampled-data control systems in general. The primary advantage of a RTOS lies in its ability to complete tasks deterministically on a time deadline, reducing jitter commonly associated with a pre-emptive multitasking OS (PMOS). Furthermore, the lack of pre-emption and its associated overhead can allow a RTOS to achieve computational loop rates generally faster than those of a typical OS, such as is typically used with commercial haptic interfaces.

Additional gains in sampling rate can be realized through the use of a FPGA, an array of logic gates that can be configured at will to execute arbitrary digital operations. These operations can range from Boolean logic to digital counters and multiplication, all occurring as fast as the signal propagation permits [14]. The flexible nature of an FPGA allows for unrelated operations to occur completely in parallel; where a standard processor must perform multiple math operations sequentially, an FPGA can perform the operations simultaneously. In addition, the interface between an FPGA and a host operating system can be customized to the programmer's specifications to minimize the necessary communication. Speed and flexibility come at a price, however, as the number of available logic gates on a given FPGA is limited, imposing restrictions on the ultimate number of operations that can occur. Despites these limitations, FPGAs have been previously used as supplementary velocity estimators [15] and low power haptic controllers [16]. Vasudevan et al. [17] compared performance in terms of maximum achievable wall stiffness for a simple one degree-of-freedom custom haptic interface with a conventional haptic control loop and a haptic control loop that uses FPGAs. They achieved virtual wall stiffnesses four to five times greater than a conventional haptic controller with the use of FPGA. Galvan et al. [18] implemented an FPGA based controller for the NASA JPL force reflecting hand controller of comparable complexity to commercial haptic devices, indicating the FPGA's suitability for use in our proposed modification of the PHANToM haptic interface computational platform.

This paper examines the possible gains in sampling rate and virtual wall stiffness of an impedance-based haptic interface (PHANToM 1.0) paired first with a RTOS using FPGA for data acquisition, and then using an FPGA for all computation and communication. Sampling rates and maximum achievable virtual surface stiffnesses for both computational platforms are compared with the performance of the PHANToM haptic interface controlled via a standard PMOS similar to the commercial system's out of the box configuration.

2 Methods

We propose a method of increasing the fidelity of a PHANToM Premium 1.0 commercial haptic interface by controlling it via a FPGA both alone and with a RTOS control system. This section will present the haptic device, system architectures, and experiment design.

2.1 Haptic Interface System. We investigate the performance of the PHANToM 1.0 haptic interface, a common commercially available haptic device, which is characterized by the system parameters given in Table 1 [5]. These values, which correspond to the performance of the PHANToM 1.0 as it is commercially distributed, are used in Eq. (1) to determine the theoretical maximum stiffness for maintaining passivity. Based on the inherent damping and the sampling rate, the maximum achievable stiffness is calculated as 10 N/m. Based on the inherent Coulomb friction and encoder resolution of the PHANToM, the maximum stiffness to ensure passivity is 2612 N/m. This discrepancy implies that the passivity criterion for the PHANToM is dominated by sampling rate, and therefore the PHANToM is an excellent candidate for our study of the potential benefit of modified computational platforms.

Table 1 Estimated PHANToM parameters [5]

Symbol	Value	Units
b	0.005	N s/m
T	0.001	S
f_c	0.038	N
Δ	29.1	μm

The commercial PHANToM 1.0A haptic interface is packaged so that it can be used in a plug and play fashion, with the user connecting the hardware to a standard personal computer via a custom interface card and installing the general haptics open software toolkit (GHOST) software to operate the haptic device. The C++ object-oriented development toolkit is intended to aid designers in modeling haptic environments through a hierarchal col-

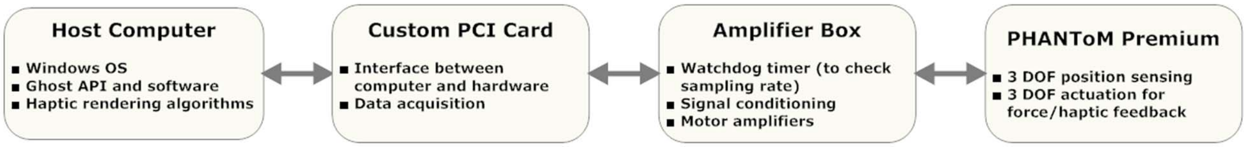


Fig. 1 PHANToM commercial package components

lection of geometric and spatial effects. As illustrated in Fig. 1, the haptic device itself is connected to an amplifier box, which contains pulse width modulation motor drive components and necessary signal conditioning electronics. The conditioned signals coming from the amplifier box are passed through a custom PCI card that interfaces the PHANToM with a standard personal computer.

In order to operate the PHANToM haptic interface on a custom computational hardware platform, the cable connecting the amplifier to the computer was intersected and the signals were observed with an oscilloscope while the PHANToM ran in its default configuration. In this way, the location, range, and purpose of various digital and analog signals involved in the robot's operation were determined. Although newer versions of the PHANToM premium devices (1.0, 1.5, and 3.0) are distributed with a parallel port interface, which would require additional information about data exchange between device and computer in order to implement a customized computational system; the findings presented in this paper are applicable to any haptic device's computational platform.

The FPGA chip to be interfaced with the PHANToM is housed within National Instrument's R Series intelligent DAQ device (PXI 7831R). The channels on this target are configured to perform the low-level read/write control of the haptic communication loop, processing the digital signals from differential encoders to acquire the joint vector angles and commanding analog signals to the PHANToM's motors. In order to define the behavior of the FPGA logic gates for custom I/O low-level operations, one must use hardware description language (HDL), specifically its most common form: very high speed integrated circuit (VHSIC) HDL (VHDL). This software development language typically has a steep learning curve and can therefore pose difficulties for those unfamiliar with its format. However, National Instruments has developed an FPGA software module that allows a user to develop and compile custom FPGA logic to the chip using LABVIEW graphical programming software; any prior knowledge of VHDL is not necessary.

The signals that are acquired and commanded by the FPGA device are targeted by a haptic rendering algorithm executed in real time on a separate dedicated processor located on National Instrument's PXI 8176 Embedded Controller. The time-critical software application is programmed in a similar fashion, using the LABVIEW real-time module to develop the haptic rendering process in a graphical programming environment on a Windows PC. The LABVIEW code is then deployed to the real-time target through a network connection. The haptic simulation loop executes with ultimate priority on the PXI 8176, while a front panel on the Windows host computer maintains lateral communication (separate timing), providing any visual interfacing to the operator. It is at the Windows host machine that the user designates virtual wall properties such as wall location and spring stiffness, commands that are updated to the real-time loop at the beginning of the next cycle. A schematic of this setup is shown in Fig. 2.

2.2 Computational System Architectures. Three possible control schemes are examined in this work, though all rely on an FPGA acting as a custom device I/O solution as illustrated in Fig. 2. The most complex solution requires additionally that all control calculations occur onboard the FPGA, with only control parameters and data passed to the host machine, thereby offering the greatest potential gains in sampling rates. Moving the calculations off the FPGA to a host machine running an RTOS still offers moderate sampling rate gains while greatly decreasing complexity and harnessing the floating point arithmetic capabilities of modern processors. Similarly, running a PMOS instead of an RTOS, while further reducing maximum possible sampling rates, offers simplified debugging and programming changes.

2.2.1 Multitasking Operating System. A control system based on a PMOS offers a reasonable approximation of the control scheme adopted by many commercial haptic interfaces, including the PHANToM haptic interface. These commercial devices rely on custom device I/O hardware connected to standard PC communication buses or ports. Device drivers and simulations run in a standard interruptible PMOS. To emulate the performance of the commercial out of the box PHANToM system, while still enabling objective comparison with the proposed computational platforms, we programmed an FPGA to mimic the function of custom device I/O hardware interfacing with the PMOS.

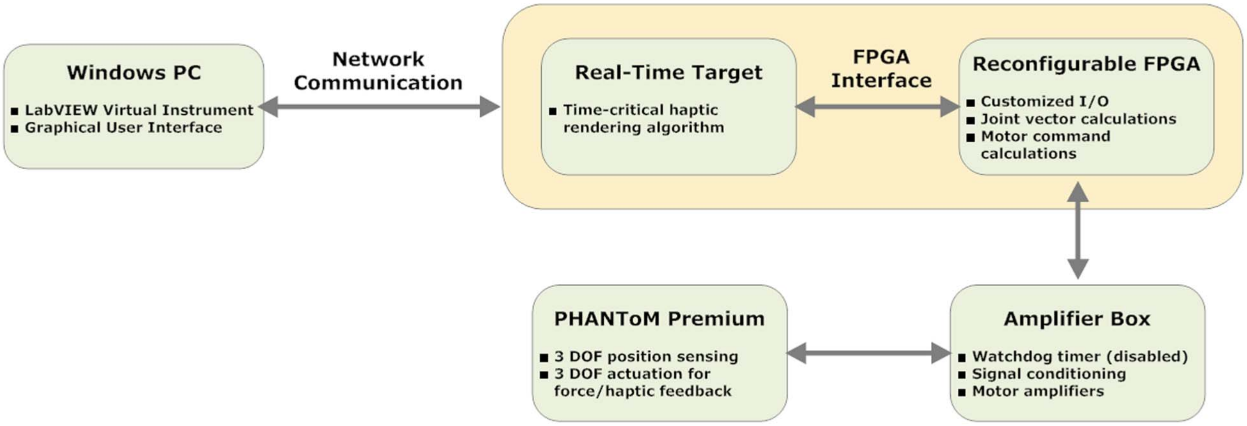

Windows PC
■ LabVIEW Virtual Instrument
■ Graphical User Interface
Network Communication
Real-Time Target
■ Time-critical haptic rendering algorithm
FPGA Interface
Reconfigurable FPGA
■ Customized I/O
■ Joint vector calculations
■ Motor command calculations
PHANToM Premium
■ 3 DOF position sensing
■ 3 DOF actuation for force/haptic feedback
Amplifier Box
■ Watchdog timer (disabled)
■ Signal conditioning
■ Motor amplifiers

Fig. 2 Custom hardware platform setup with real-time operating system and FPGA

Software running in a PMOS environment must contend with numerous hardware interrupts and the multitasking nature of the OS. Without strict control of other running software, a computationally intensive haptic application risks being starved of enough processor time to maintain a constant loop rate without missing updates. In addition, the application must deal with interrupts caused by other hardware and software on the host machine. These interrupts halt all other software while they are being handled. Since the source of the interrupts is beyond the control of the haptic application, interrupts inject an essentially unpredictable amount of delay into the operation of the haptic application. However, the application does benefit greatly from being executed on a modern processor with strong floating point capabilities. Floating point math allows arithmetic operations on numbers of greatly varying magnitude with little loss of precision. While more complicated and computationally expensive than integer math, the use of floating point operations ensures that an already quantized system suffers no further loss of information.

2.2.2 Real-Time Operating System. An RTOS is similar to a PMOS in that it is a multitasking environment and it must still interface with a haptic interface through custom I/O hardware and device drivers. The fundamental difference is that an RTOS is designed to make it possible for the program to meet specified timing deadlines, through the use of specialized scheduling software.

The fundamental difference between the RTOS and the PMOS is that the RTOS is capable of running deterministically, guaranteeing that properly developed software will run at a specified rate without missing any updates or suffering from interrupt-induced jitter. The fact that an RTOS typically runs only the desired application and the necessary support software helps to reduce the total load on the processor and increase possible loop rates.

2.2.3 Field Programmable Gate Array. A system based on the concept of performing all necessary calculation on an FPGA differs greatly in performance and implementation from the previous two control systems. The FPGA acts as custom I/O hardware interfacing with the haptic device as in the previous computational platform. In addition, all necessary calculations for a haptic rendering algorithm are also carried out on the FPGA. This structure eliminates the possibility of any other software applications delaying the calculations required to compute the desired haptic feedback. The parallel nature of the FPGA fundamentally changes the nature of the communication path.

With the FPGA calculating the appropriate feedback, a constant loop rate is guaranteed. The FPGA can, in fact, operate completely independent of input from the host machine, so long as parameter updates are not necessary. However, this does not come without a price. It is impractical to implement floating point arithmetic on an FPGA for all but the simplest haptic devices. In practice, all arithmetic on an FPGA must be carried out with fixed point numbers.

As all nontrivial haptic applications will require decimal numbers, a fixed point arithmetic algorithm must be developed in lieu of floating point arithmetic. This entails the use of a binary number that has a number of bits before and after the radix point. To convert a decimal number to a fixed point of this form one simply multiplies the original number by 2 raised to the power of the number of digits right of the radix. The answer is truncated and the remaining integer represents the original decimal number [19].

However, fixed point arithmetic on the FPGA does have drawbacks. While basic mathematical operations remain largely the same, calculating any transcendental function would prove too computationally costly to be practical. The simplest solution to this would be to use a precalculated lookup table to acquire any necessary values. Additionally, a fixed point number lacks the dynamic range of a floating point number. This implies both a fixed precision and fixed upper and lower values for the number, so great care must be taken to maximize the precision while ensuring that there will be no value overflows [19].

2.3 Experiment Design. In order to compare passivity of the PHANToM during haptic interactions, a virtual environment was implemented on PMOS/FPGA (mimicking the commercial haptic system), RTOS/FPGA, and FPGA platforms. The environment consisted of selectable pairs of opposing virtual walls in the x, y, and z planes, forming a cube-shaped room. The FPGA device was a National Instruments PXI-7831R intelligent DAQ device, housed inside a NI PXI-1083 chassis with a PXI-8186 embedded controller.

In the PMOS and RTOS platforms, the FPGA was configured to act simply as an encoder counter, watchdog generator, and D/A motor controller to interface with the PHANToM proprietary amplifier box. The virtual room was implemented on the NI embedded controller via NI's LABVIEW V8.2 RTOS or in NI LABVIEW, as appropriate. For the RTOS, the virtual room was controlled via a TCP/IP connection from a remote computer to the embedded controller, but all processing and data storage occurred on the embedded controller.

The FPGA platform was implemented using 32-bit integer math with a fixed point shift of 22 bits. Trigonometric functions were implemented using 32-bit lookup tables to directly derive trigonometric values from encoder counts. However, due to lookup memory constraints of the FPGA, the functional workspace of the FPGA platform was about 90% the size of the other platforms, with invalid values returned at the extreme edges of the workspace. Otherwise, all necessary position calculations, collision detection, along with force and torque calculations were handled on-board the FPGA with loop rates of up to 400 kHz, with the RTOS system acting merely as an interface and data acquisition system.

We examined the virtual wall responses of eight control platforms: a PMOS/FPGA platform at 1 kHz (equivalent to the commercial PHANToM system); an RTOS/FPGA platform at 1 kHz, 5 kHz, 10 kHz, and 20 kHz; and an FPGA platform at 20 kHz, 100 kHz, and 400 kHz. These will be referred to as PMOS-1, RTOS-1, RTOS-5, RTOS-10, RTOS-20, FPGA-20, FPGA-100, and FPGA-400.

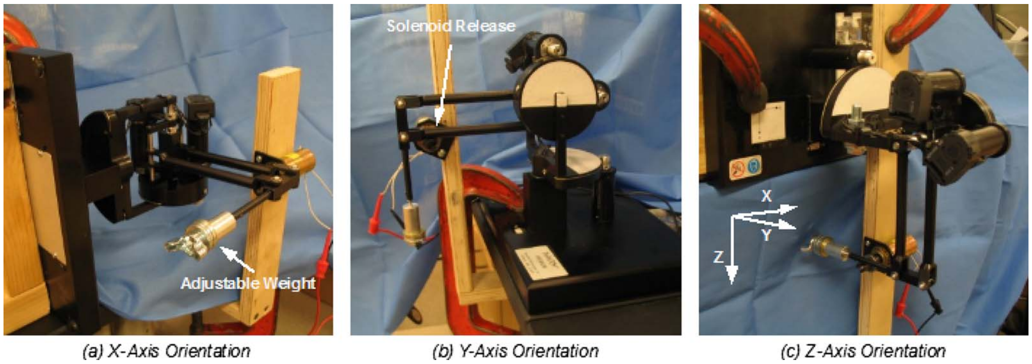
To test the passivity of the virtual wall, weighted drop tests were performed at varying stiffnesses and sample rates. During a trial, the PHANToM was oriented with gravity acting parallel to the axis under test. Figures 3(*a*)–3(*c*) present the PHANToM oriented for tests in the x, y, and z axes, respectively. An adjustable weight was positioned at the end of the PHANToM in place of a stylus to simulate human touch. This weight was adjusted for each orientation such that the end of the linkage exerted 0.15 N on a scale.

The virtual environment under test in each trial consisted of a pair of virtual walls oriented normal to gravity and the axis under test. These walls were situated 2 cm from the zero position of the PHANToM and the stiffnesses were varied from 5 N/m up to 40,000 N/m where possible. It should be clarified that these manipulations of virtual wall stiffness were strictly computational, and performance limitations of the hardware (such as the rigidity of the links of the manipulator) would limit the actual rendered stiffness at the stylus of the device.

The PHANToM itself was supported in the zero position by a digitally controlled solenoid release. At the start of the trial, the solenoid would release the PHANToM and the control platform would record the Cartesian coordinates of the weight at 1kHz for 10 s to provide ample data to examine the platform's response while resting on the wall. Three trials were run in each direction, at each stiffness, for each platform.

3 Experimental Results

Figure 4 presents the responses of two RTOS-20 trials in the y-direction with stiffnesses of 250 N/m and 5000 N/m. The re-



(a) X-Axis Orientation

(b) Y-Axis Orientation

(c) Z-Axis Orientation

Fig. 3 PHANToM experiment hardware orientations. A solenoid release was triggered to initiate the stylus dropping to the virtual surface. Orientations were varied to leverage gravitational forces, and adjustable weights ensured consistent applied forces on the virtual surfaces.

sponses have been cropped to 2 s for clarity, but it is easy to see the empirical difference between the passive response at 250 N/m and the nonpassive response at 5000 N/m.

To compare the passivity of the various systems at varying stiffnesses, the steady state response of the system between 7000 ms and 10,000 ms is analyzed. The root mean square (rms) of the data around the minimum steady state value is recorded as a rough measure of the excess energy in the system. Passive response was characterized by an rms less than 0.003 cm, or the resolution of the encoders, observed for three trials. A summary of the findings is presented in Table 2. In the table, squares with "x" symbols represent configurations that resulted in desirable passive behavior when the stylus interacted with the virtual environment. White squares indicate highly unstable behavior for these configurations, so rms values were not recorded and trials were aborted to protect the hardware. All other configurations (shaded squares) resulted in nonpassive virtual wall interactions.

RMS values in the FPGA and RTOS platforms followed different trends as virtual wall stiffnesses were increased, as displayed in Fig. 5. The RTOS and FPGA platforms both initially transitioned to nonpassive behavior gradually, but the FPGA platform did not increase as rapidly in nonpassive behavior. In addition, the RTOS-20 and FPGA-20 values track closely, as do the FPGA-100 and FPGA-400 values.

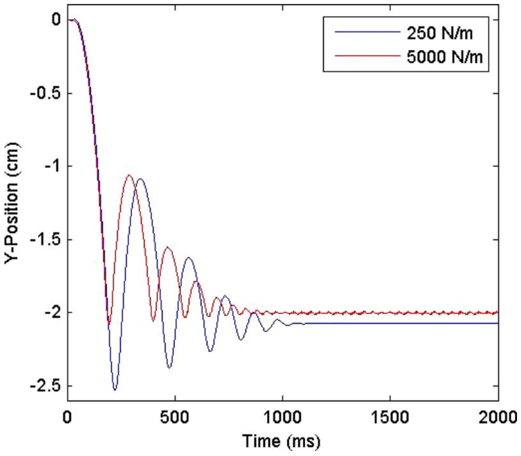


Fig. 4 RTOS-20 passive versus nonpassive behavior comparison. The virtual wall is located at −2 cm in the *y*-direction. The lower-stiffness virtual wall (500 N/m) allows deeper penetration into the virtual surface, and at steady state, the interaction is passive. The higher-stiffness virtual wall (2500 N/m) exhibits nonpassive behavior at steady state (beyond approximately 1000 ms).

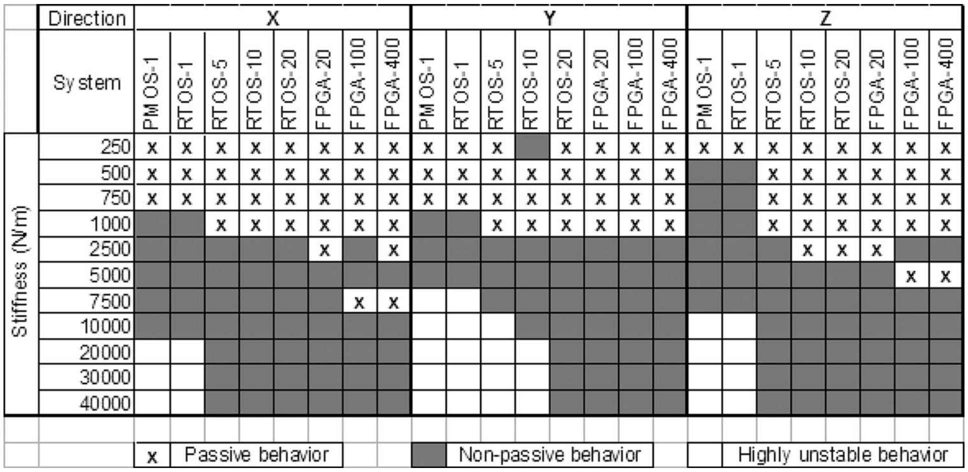
4 Discussion

The experimental results show that increasing sampling rates through the use of an RTOS/FPGA platform can greatly increase the range of achievable stiffnesses of an impedance-based haptic device while maintaining passivity. The maximum achievable stiffness can clearly be seen to increase as the sampling rate increases, a correlation that demonstrates the benefit to be gained by employing RTOS platforms for haptic control, even with more modest increases in sampling rate. As shown in Table 2, when comparing the PMOS and RTOS systems at the same update rate (1 kHz), there are no changes in the achievable stiffness for passive interactions. However, by increasing to 5 kHz update rate on the RTOS, the achievable stiffness is at least two times greater than the PMOS. There are also notable improvements in stiffness achievable when using an all-FPGA computational platform rather than the RTOS with FPGA only for data acquisition for the *x* and *z* axes of the device, since much higher loop rates can be achieved when all computation occurs on the FPGA. As noted in Table 2, for these axes, the all-FPGA configuration enables maximum virtual surface stiffnesses more than five times greater than those realized with the PMOS and RTOS architectures. These gains in stiffness are comparable to those achieved by Vasudevan et al. [17] for a one degree-of-freedom custom haptic device and FPGA-based haptic controller. The FPGA-100 and FPGA-400 platforms exhibited very similar performance at high stiffnesses, suggesting that increasing the sampling rate on the RTOS or FPGA platforms beyond 100 kHz has little effect on the passivity of the system. One possible explanation is that the higher loop rates have exceeded the bandwidth of the device actuators or the PHANToM amplifier, and thus the motors and/or amplifiers are now the limiting factor. Another possibility is that the encoder resolution is now what limits the achievable stiffness in the PHANToM. The differences in performance between the RTOS system and FPGA system depend on the axis of the device being tested, since each axis exhibits different dynamic behavior due to the mechanical design, inertia, and transmissions.

The observed maximum virtual environment stiffness on the PMOS-1 system, which mimics the commercially distributed PHANToM architecture, is significantly higher (750–1000 N/m) than the theoretical maximum stiffness of 10 N/m, which is limited by the inherent device damping and sampling period. This disconnect is consistent with findings of Dioliati et al. that showed nonlinear effects (e.g., Coulomb friction) of the haptic interface enable many common devices, including the PHANToM, to operate stably, yet in violation of the passivity criteria [5]. As a result, experimental findings often differ from the theoretically computed maximums.

The possible benefits of FPGA interfaces to the haptic community are great in spite of the challenge of their application. This

Table 2 Summary of passive behavior across hardware systems. The results for each axis (*x*, *y*, and *z* in Cartesian space) of the PHANToM are presented for the eight computational platforms. The PMOS-1 system is equivalent to the out of the box PHANToM performance. RTOS systems utilize a real-time operating system with FPGA for data acquisition. FPGA systems carry out all computation and communication on the FPGA. The number following the system abbreviation corresponds to the sampling rate of the system (in kilohertz). Passive behavior at the greatest possible virtual wall stiffness is desirable. The modified systems (RTOS and FPGA) enable increased sampling rates and subsequently increased virtual wall stiffnesses while maintaining passive interactions. Highly unstable trials were aborted to protect the hardware.



	Direction	X								Y								Z							
	System	PMOS-1	RTOS-1	RTOS-5	RTOS-10	RTOS-20	FPGA-20	FPGA-100	FPGA-400	PMOS-1	RTOS-1	RTOS-5	RTOS-10	RTOS-20	FPGA-20	FPGA-100	FPGA-400	PMOS-1	RTOS-1	RTOS-5	RTOS-10	RTOS-20	FPGA-20	FPGA-100	FPGA-400
Stiffness (N/m)	250	x	x	x	x	x	x	x	x	x	x	x	▨	x	x	x	x	x	x	x	x	x	x	x	x
	500	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	▨	▨	x	x	x	x	x	x
	750	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	▨	▨	x	x	x	x	x	x
	1000	▨	▨	x	x	x	x	x	x	▨	▨	x	x	x	x	x	x	▨	▨	x	x	x	x	x	x
	2500	▨	▨	▨	▨	▨	x	▨	x	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	x	x	x	▨	▨
	5000	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	x	x
	7500	▨	▨	▨	▨	▨	▨	x	x	□	□	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
	10000	▨	▨	▨	▨	▨	▨	▨	▨	□	□	□	▨	▨	▨	▨	▨	□	□	▨	▨	▨	▨	▨	▨
	20000	□	□	▨	▨	▨	▨	▨	▨	□	□	□	□	▨	▨	▨	▨	□	□	▨	▨	▨	▨	▨	▨
	30000	□	□	▨	▨	▨	▨	▨	▨	□	□	□	□	▨	▨	▨	▨	□	□	▨	▨	▨	▨	▨	▨
	40000	□	□	▨	▨	▨	▨	▨	▨	□	□	□	□	▨	▨	▨	▨	□	□	▨	▨	▨	▨	▨	▨
		x	Passive behavior							▨	Non-passive behavior							□	Highly unstable behavior						

paper demonstrates that using FPGA even just for data acquisition between the haptic device and control computer running a real-time operating system provides measurable gains in the achievable virtual surface stiffness on a haptic virtual environment, which enables higher fidelity simulations, especially to convey rigid contact. The extended use of the FPGA in this work, demonstrating that a haptic environment can be completely simulated on a modest FPGA, opens many new doors. Such a system could be configured to act as an adaptable interface to many different haptic devices. If the FPGA is capable of handling all translation to and from Cartesian space, virtual environments and haptic enabled programs could be coded in pure Cartesian space with the FPGA acting as translator to whatever haptic interface is required, greatly simplifying the development of new programs. More advanced FPGA devices could be configured to deterministically run fast local models to control haptic interfaces, leveraging the greater storage and floating point abilities of a host computer to update those models as necessary, greatly improving the fidelity of the haptic environment. For applications that involve greater complexity in their graphical models and rendering algorithms, porting of some of the data acquisition and communication between the device and software to FPGA still enables the system to realize increased sampling rates, thereby improving fidelity.

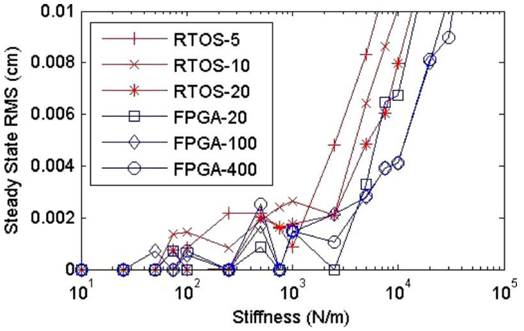


Fig. 5 *y*-axis rms values for RTOS and FPGA. Steady-state RMS values above 0.003 cm are considered as nonpassive interactions. FPGA systems are able to display higher stiffnesses before reaching the defined limit for nonpassive behavior than the RTOS systems. Increasing the sampling rate of the FPGA from 100 kHz to 400 kHz does not result in any further improvements in maximum stiffness.

5 Conclusions

This paper investigates a method for improving haptic fidelity by increasing the achievable stiffness of virtual walls on a PHANToM 1.0 haptic interface via increased sampling rates on the computational control platform. We employed a RTOS for deterministic loop rate timing and a FPGA to reduce system overhead. First, a combined RTOS with FPGA platform was proposed to enable deterministic loop rate timing and fast communication, increasing the achievable sampling rate and subsequently the maximum virtual surface stiffness that could be rendered passively compared with a commercial haptic interface (PHANToM) running on a PMOS. Then, all computation and communication was moved to the FPGA to further increase sampling rates and achievable stiffnesses. The RTOS system passively displayed virtual walls twice as stiff, while the FPGA system passively displayed virtual walls six times as stiff as the commercially available system. Large virtual surface stiffnesses are desirable because they improve the fidelity and realism of the haptic display. The results validate the use of an RTOS and an FPGA as feasible methods of increasing the fidelity of haptic virtual environments via increased sampling rates. Though demonstrated on a single commercial haptic interface hardware platform, the improvements in fidelity gained through the use of such deterministic low-overhead computational components can be realized for any haptic interface hardware.

Acknowledgment

The authors acknowledge National Instruments for providing hardware for the experiments.

References

[1] Colgate, J., Grafing, P., Stanley, M., and Schenkel, G., 1993, "Implementation of Stiff Virtual Walls in Force-Reflecting Interfaces," *Proceedings of the IEEE Virtual Reality Symposium*, pp. 202–208.
[2] Abbott, J. J., and Okamura, A. M., 2005, "Effects of Position Quantization and

Sampling Rate on Virtual-Wall Passivity," IEEE Trans. Rob., **21**(5), pp. 952–964.

[3] Kazerooni, H., 1993, "Human Induced Instability in Haptic Interfaces," *Proceedings of the ASME Winter Annual Meeting*, Vol. 55, pp. 851–856.

[4] Gillespie, B., and Cutkosky, M., 1996, "Stable User-Specific Rendering of the Virtual Wall," *Proceedings of ASME IMECE*, Vol. 58, pp. 397–406.

[5] Diolaiti, N., Niemeyer, G., Barbagli, F., and Salisbury, J. K., Jr., 2006, "Stability of Haptic Rendering: Discretization, Quantization, Time Delay, and Coulomb Effects," IEEE Trans. Rob., **22**(2), pp. 256–268.

[6] Miller, B., Colgate, J., and Freeman, R., 2004, "On the Role of Dissipation in Haptic Systems," IEEE Trans. Rob. Autom., **20**(4), pp. 768–771.

[7] Colgate, J., and Brown, J., 1994, "Factors Affecting the Z-Width of a Haptic Display," *Proceedings of the IEEE International Conference on Robot and Automation*, pp. 3205–3210.

[8] Adams, R. J., and Hannaford, B., 1999, "Stable Haptic Interaction With Virtual Environments," IEEE Trans. Rob. Autom., **15**(3), pp. 465–474.

[9] Hannaford, B., and Ryu, J., 2001, "Time-Domain Passivity Control of Haptic Interfaces," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1863–1869.

[10] Hayward, V., and Astley, O., 1996, "Performance Measures for Haptic Interfaces," *Proceedings of the 1996 Robotics Research: Seventh International Symposium*, pp. 195–207.

[11] Mahvash, M., and Hayward, V., 2005, "High-Fidelity Passive Force Reflecting Virtual Environments," IEEE Trans. Rob., **21**(1), pp. 38–46.

[12] Mahvash, M., and Hayward, V., 2005, "Haptic Rendering of Cutting: A Fracture Mechanics Approach," http://www.haptics-e.org

[13] McKnight, S., Melder, N., Barrow, A. L., Harwin, W. S., and Wann, J. P., 2005, "Perceptual Cues for Orientation in a Two Finger Haptic Grasp Task," *Proceedings of the IEEE WHC'05*, pp. 549–550.

[14] National Instruments, 2006, "Introduction to LABVIEW FPGA," LABVIEW 8 FPGA Module Training, Retrieved from zone.ni.com/devzone/cda/tut/p/id/3555/

[15] Çavusoglu, M. C., Feygin, D., and Tendick, F., 2002, "A Critical Study of the Mechanical and Electrical Properties of the PHANToM Haptic Interface and Improvements for High Performance Control," Presence: Teleoperators and Virtual Envirnonments, **11**(6), pp. 555–568.

[16] Holbein, M., and Zelek, J. S., 2005, "A FPGA Haptics Controller," *Proceedings of the IEEE WHC'05*, pp. 588–589.

[17] Vasudevan, H., Srikanth, M. B., and Muniyandi, M., 2007, "Rendering Stiffer Walls: A Hybrid Haptic System Using Continuous and Discrete Time Feedback," Adv. Rob., **21**(11), pp. 1323–1338.

[18] Galvan, S., Botturi, D., and Fiorini, P., 2006, "FPGA-Based Controller for Haptic Devices," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 971–976.

[19] Kraeling, M. B., 1996, "Fixed-Point Math in Time-Critical C Applications," *Proceedings of WESCON/96*, pp. 587–593.