A Multithreaded Implementation of Assist Functions to Control a Virtual Reality Model of a 6-DoF Robot Arm for Rehabilitation Applications

Eduardo J. Veras, Ramya Swaminathan, Rajiv Dubey
University of South Florida
4202 E. Fowler Ave
Tampa, FL 33620
(813) 974-2280
everasjo@eng.usf.edu, rswamina@eng.usf.edu, dubey@eng.usf.edu

ABSTRACT

This paper describes a multithreaded platform to implement a haptic interface to control a Virtual Reality (VR) model of a 6-DoF arm with applications towards rehabilitation robotics. The approach allows the defined threads to execute concurrently, resulting in increased utilization of the processor resources and higher instruction execution rates consistent with a realistic rendering of the touch sensation and smooth graphics transitions for the VR simulation. The haptic interface allows for physical rendering of the human–machine interactions through force feedback and velocity scaling enhancing the manipulation capabilities of persons with disabilities. Assist functions are modeled through the use of physical simulation of springs, spring with damping, and velocity scaling. The level of assistance is calculated by concurrent threads running the virtual model of the arm and the haptic loop at two different update rates.

Keywords

assist functions, haptics, multithreading, rehabilitation robotics, virtual reality.

1.INTRODUCTION

This paper describes a multithreaded PC-based implementation of a controller for a haptic interface to control a Virtual Reality (VR) model of a 6-DoF robot arm. In order to provide a realistic rendering of the feedback forces to constraint the user's motion to follow a prescribed trajectory, the update rate of the feedback signals from the haptic device must be at least 1000Hz. This rate will allow for the generation of rigid body sensations in the user's hands [1, 2]. In order to accomplish this, the following threads were defined:

1- The determination of the target position (in joint or Cartesian space),

2- The computation of the joint angles to reach the desired position,

3- A trajectory generation thread which computes position set point commands, and

4- A graphic thread which updates the display of the VR model of the Puma 560.

5- An assist function thread which updates at the same frequency of the haptic loop and it computes the level of assistance provided to guide the user along the linear trajectory.

Since there are multiple threads running at the same time, there is a chance of conflict when accessing shared memory or data structures. For example, if one thread is writing data to the memory and a second thread is reading from that memory. To avoid data corruption, we need to synchronize these threads execution in order to have exclusive access to shared resources. In our implementation, a scheduler class is used for synchronization. Three types of assist functions are provided to guide the user with motion impairment:

1. A simple spring-type force feedback where an attractive force is created when the user is close to the desired trajectory. The closest point between the tip of the haptic device and the tracking path is a predefined threshold value.
2. Velocity scaling where the velocity component in the direction of the desired trajectory is amplified; otherwise, it is attenuated.
3. Constant force assistance where the force projection in the direction of the desired path is scaled up; otherwise, it is scaled down.

Each type of assist function is computed in a separate thread and they are updated at the frequency of the haptic loop of 1.0 KHz. Experiments (1, 2, and 3) were conducted applying these assist functions, as it is discussed in detail in section 7.

2.RELATED WORK

As far as rehabilitation is concerned, robotics has still a long way to go. Stroke rehabilitation is one of the main areas where robot assisted devices are extensively used. They help in the study of functional adaptation after a stroke. The greatest impacts of the application of robotics in rehabilitation are not just the devices themselves, but also the infrastructure supporting rehabilitation. The research helps to provide more precise, objective, and detailed data on what actually happens during the recovery. This in turn, would provide better understanding of the key biomechanical and neurological factors required for successful rehabilitation. The main advantage of robot assisted therapy is that, they allow semi-autonomous practice of therapeutic tasks. Now, when we apply haptic technology to robotics, we provide another dimension to teleoperation assisted rehabilitation. The term *haptic* is derived from the Greek word *haptesthai,* meaning “to touch”. The haptic sensory system employs both cutaneous and kinesthetic receptors when engaged in an active procedure. Basically, any kind of touch becomes "active", when the sensory inputs are combined with controlled body motion. Haptic rendering is defined as the process of computing and generating forces in response to user interactions with virtual objects. They offer important applicability in engineering and medical training tasks." The past research [3] in haptic interface implemented several forms of assistance functions designed to augment human performance. The test bed used for these tasks consisted of a six-degree-of-freedom force reflecting haptic interface device called the Phantom, with the GHOST SDK software produced by Sensable Tech. More recently, this company developed the Omni Phantom which is a more affordable haptic interface and it uses the OpenHaptics toolbox for programming. Pernalete [3] demonstrated that for a set of chosen tasks, the assistance functions significantly reduced execution times and enhanced the individual's performance. Arsenault et al [4] implemented a haptic device interface to test eye-hand coordination during the manipulation of any 3D object in the virtual world. They proved that haptic rendering of virtual objects improved the eye-hand coordination for user interactions. In Teleoperation applications, a user is able to perform complex tasks in a remote environment. For example, removal of bombs, mines and inspection of underwater structures require the intervention of a remote operator. The visual feedback plays an important role for these task executions and [5] proved that, the use of a haptic Interface with force feedback assistance increases the user's perception of a workspace of a mobile robot. However, the Teleoperation implementations are generally used by skilled operators and they don't include features to accommodate persons with disabilities in a straight forward manner. On these lines, we interfaced the Omni Phantom to the virtual reality model of the Puma 560 manipulator. The robot arm is driven by the haptic interface. The implementation of the assist force functions help the user to feel the virtual environment and to assist in the manipulation of virtual objects. The user would be guided by the force feedback function when moving on a linear or curvilinear trajectory. Primarily, this would help people with hand disabilities to control their hand oscillations when moving the stylus on a trajectory path and enable them to reach their target easily.

A key problem as well as possible enhancement to robotic frameworks for rehabilitation applications is the integration of human-machine interactions (HMI) through haptic response or sense of touch when it is embedded in the control software. Turro et al [6] described a system for haptically augmented teleoperation using a master/slave scheme. QNX RTOS was used to implement the slave controller and a multi-processor Linux PC (4 CPU's) to control the master device (five CPU's total). Charles et al [7] developed the Robot-Assisted Microsurgery (RAMS) telerobotic workstation in collaboration with JPL/NASA to augment micro-surgical dexterity. The system includes a 6-DoF robotic manipulator (slave) that holds surgical instruments. Motions of the instruments are commanded by moving the handle on a master device in the desired trajectories. The system was designed to assist skilled and able-bodied surgeons and is not suitable to assist people with disabilities to execute activities of daily living, ADL's. A bilateral teleoperation approach was implemented by Everett et al [8], where a slave manipulator (a RRC robot manipulator) is controlled by tracking the motion of a master manipulator (Phantom device). When the master touches an object, the slave reflects the forces back to the master device held by the operator. It was developed using a SGI workstation and the VxWorks OS. In contrast to these systems, the design described in this paper allowed us to create a simplified PC-based framework which can be implemented widely to benefit the community, particularly, users with motor impairment in a rehabilitation setting. The system incorporates the advantages of concurrency and multithreading programming in a PC-based framework providing the benefits of a research laboratory setup to the user's desktop.

3.HAPTIC RESPONSE FILTERING AND SCALING

Figure 1 shows a block diagram of the haptic controller. The control strategy is a form of generalized bilateral control, which maps positions and velocity components between the haptic workspace and the VR Puma 560 workspace. The rate of change, i.e., the velocity components are computed at a constant interval or sampling time. This scheme provides a suitable data flow for velocity mapping and scaling. The current system implements robotic technology and three different type of assist function to help people with disabilities to perform activities of daily living, ADL's [9, 10]. The regularly required tasks for the haptic interface are: following a prescribed path, force reflection, impedance simulation, and obstacle avoidance. A force computational model can also be implemented using a mass-spring governing equations [9, 10]. In order to solve these problems while providing assistance to the user's motion, we need to be able to develop haptic controlled interfaces to drive multiple degrees of freedom manipulators [12, 13, 14]. The nature of this type of application demands for a real-time response in order to be effectively usable for enhancing the manipulation capabilities of persons with disabilities. This means that it is not acceptable to have delays in the haptic response. For example, it is not acceptable that the haptic device tip penetrates the rigid body rendered in the graphical scene during a haptic cycle. A human-machine cooperative control of this nature, in which the human is always in control and the haptic scales the input to guide the user, deterministic timing is fundamental. Even though Windows is not a real-time operating system, we were able to achieve good performance to solve the problem of multiple threads running simultaneously at different frequencies for the haptic and the virtual reality simulation.

In practice, there will be measurement errors between the desired position and orientation and the user's input due to the reduced physical capabilities of the disabled person interacting with the system. These error signals are used to compute force constraint's to guide the user towards the destination. For example, in the case of approaching the surface of a table, the contact force is computed as a function of the remaining distance to the surface. If this information is not available or is delayed by a changing system loads, the user might hit the target.

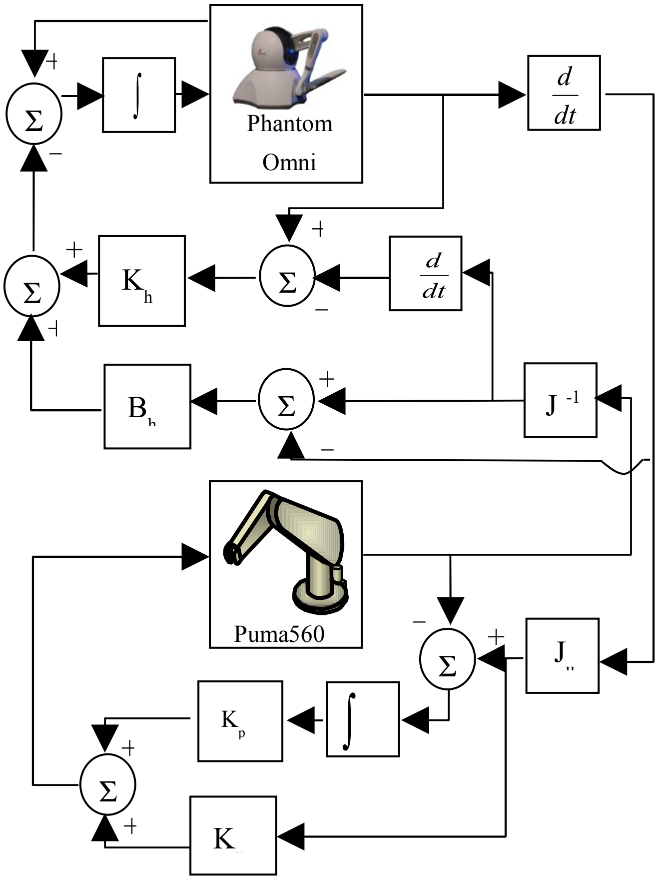

Figure 1. Haptic system block diagram

We are aiming at the development of a design architecture and controller design to meet the timing requirements of a real-time application while maintaining low cost and reduced complexity to the community of users with motion impairment. The PC-based approach allows for transferring research laboratory resources to the user's desktop, so the system can be accessed by wide community of users with constrained motion of their upper extremities. The virtual reality model of the robot arm can be driven using the haptic device while producing realistic touch sensation in the user's hand. Different computational models are used to derive the magnitude and direction of the assist functions in the form of a feedback force applied to the user's hand through the Phantom Omni device.

4.SOFTWARE IMPLEMENTATION

In a multithreaded implementation, the code where the shared resources are accessed is called a critical section or region [15]. The scheduler program implemented in this work allows multiple threads to access the CPU concurrently. As mentioned, the main threads implemented are: 1) the determination of the target position (in joint or Cartesian space), 2) computation of the joint angles to reach the desired position, 3) a trajectory generation thread which computes position set point commands, 4) graphic thread which updates the display of the VR model of the Puma 560, and 5) the assist function thread which updates at the same frequency of the haptic loop and it computes the level of assistance provided to guide the user along the linear trajectory. Since there are multiple threads running at the same time, there is a chance of conflict when accessing shared memory or data structures. To avoid data corruption, the scheduler class is used as the synchronization mechanism.

5.HARDWARE IMPLEMENTATION

The Phantom Omni device and the VR model of the Puma 560 manipulator are run on a Pentium computer, with 1GHz single processing unit. The Phantom Omni software uses the OpenHaptics development kit running under Windows XP OS [16]. The complete system was developed under MS C++ 6.0 IDE.

6.EXPERIMENTAL SETUP

The experimental setup consists of a display window depicting a linear trajectory and the end-effector represented by the sphere, as shown in Figure 2.

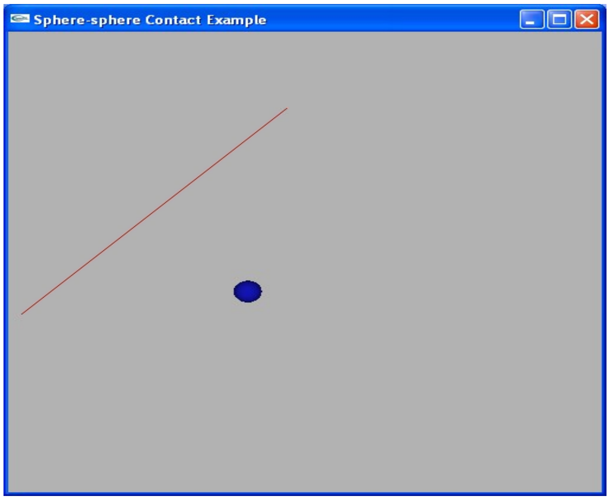

Figure 2. UI with the desired trajectory and end-effector

This sphere follows the motion of the end-effector of the Phantom Omni device. The user controls the motion of the sphere with the haptic stylus. The goal of the user is to move the sphere towards the trajectory and move it along the linear path. The user traverses the trajectory path as shown in Figure 3.

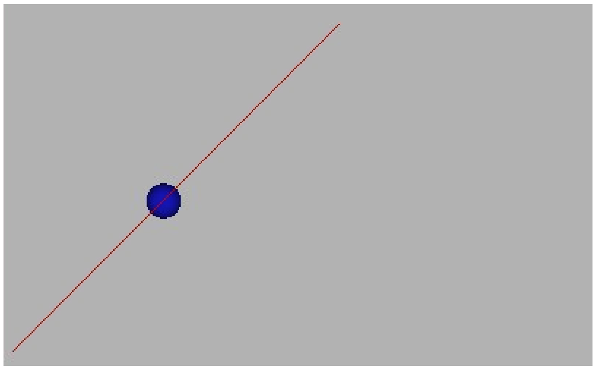

Figure 3. UI showing the end-effector on the trajectory path

As soon as the end-effector falls within a predefined threshold value, three types of assist functions are implemented, as discussed in the introduction.

7.RESULTS

Results of the three experiments performed are shown next. Figure 4 corresponds to the tracking points of Puma 560 and the Omni Phantom. As explained before, experiment 1 corresponds to a spring-type force computed using the Hooke's law.

$$\vec{F} = K\left(\vec{S}_{haptic} - \vec{S}_{trajectory}\right) \qquad (1)$$

where

$\vec{F}$ defines a force vector applied to the end-effector ,

$\vec{S}_{haptic}$ is the end -effector coordinates and

$\vec{S}_{trajectory}$ is a vector along the trajectory path.

Eq. (1) is used for guidance of the user's motion, but it is not amplified or attenuated. Figure 4 shows the results of this experiment.

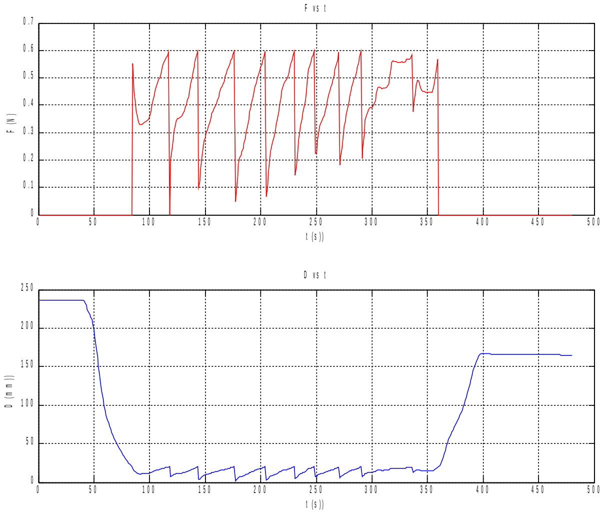

Figure 4. Force based on Hooke's Law

For the second experiment, the velocity components in the direction of the user's motion are amplified in order to provide assistance. The feedback force is computed as follows:

$$\vec{V}_{projected} = \left(\frac{\vec{V} \bullet \vec{T}}{\left\|\vec{T}\right\|^{2}}\right)\vec{V} \qquad (2)$$

where

$\vec{V}$ is the velocity vector of the end effector and,

$\vec{T}$ is a vector in the direction of the desired path.

The results using Eq. (2) are shown in Figure 5.

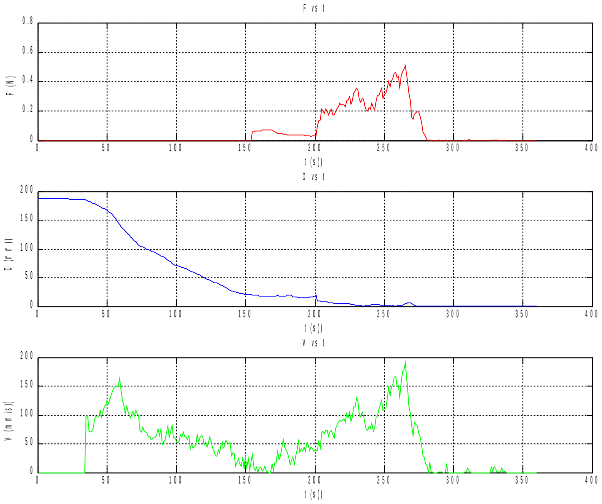

Figure 5. Force based on Velocity Scaling

Finally, in this experiment, a force with constant magnitude is projected in the direction of user's motion and is scaled to provide assistance. This force computation is as follows:

The results obtained using this approach is shown in Figure 6.

$$\vec{F}_{projected} = \left(\frac{\vec{F} \bullet \vec{T}}{\left\|\vec{T}\right\|^{2}}\right) * \vec{F} \qquad (3)$$

where

$\vec{F}$ defines a force vector applied to the end-effector and,

$\vec{T}$ is a vector pointing in the direction of the desired path.

The results using Eq. (6) are shown in Figure 6.

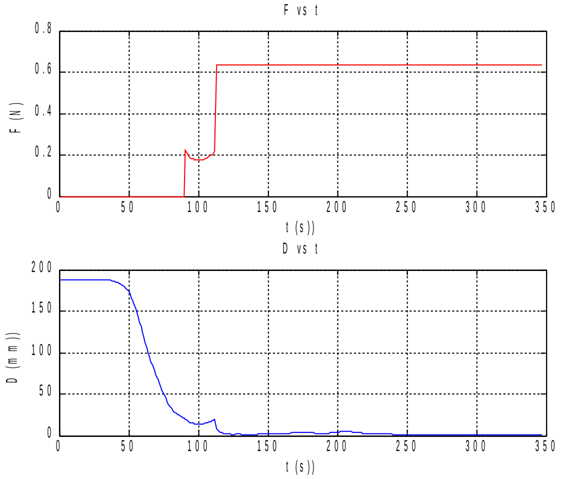

Figure 6. Constant force assistance.

8.CONCLUSIONS

This paper presents a multithreading programming interface to implement different types of assist functions to guide the user's motion during the manipulation of virtual objects. The haptic control of a virtual reality model of a 6-DoF robot arm enables the use of assistive technology devices without incurring in high costs and increased complexity associated with haptic interfaces for rehabilitation applications. Assist functions and a haptic interface integration can be achieved to assist people with disabilities by amplifying or attenuating human input in real time. The experimental results show that a better performance and stability of the response was obtained when a constant force projected in the direction of the desired trajectory is implemented. The reason for this is that oscillations introduced by shaking of the user's hands are eliminated from the force feedback computation unlike the first two experiments.

9.ACKNOWLEDGMENTS

Our thanks to the Rehabilitation Robotics Center at the University of South Florida for providing the resources to make this project possible.

10.REFERENCES

[1] J. K. Salisbury, and M. A. Srinivasan**,** "Phantom-based Haptic Interaction with Virtual Objects", Computer Graphics and Applications, IEEE Volume 17, Issue 5, Sept.-Oct. 1997, pp 6 – 10.

[2] N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. Salisbury, and C. Melchiorri,, "The Effect of Quantization and Coulomb Friction on the Stability of Haptic Rendering," in Proc. 1st World Haptics Conference, Pisa, Italy, Mar. 2005, pp. 237–246.

[3] N. Pernalete, W. Yu, R. Dubey, and W. Moreno, "Development of a Robotic Haptic Interface to Assist the Performance of Vocational Tasks by People with Disabilities", in Proc of 2002 IEEE International Conference on Robotics and Automation, Volume 2, 11-15 May, 2002, pp 1269-1274.

[4] R. Arsenault and C. Ware "Eye-Hand Co-ordination with Force Feedback", CHI Letters, Vol. 2, Issue 1, ACM Journal 2002, pp 408-414.

[5] K. Salisbury, F. Conti, and F. Barbagli, "Haptic Feedback: Introductory Concepts", IEEE Computer Graphics and Applications, Vol., Issue 2, March-April, 2004, pp24-32.

[6] N. Turro, O. Khatib, and E. Coste-Maniere, "Haptically Augmented Teleoperation", in Proc. IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21-26, 2001, pp 386-392.

[7] S. Charles, H. Das, T. Ohm, C. Boswell, G. Rodriguez, and R. Steele, "Dexterity-enhanced Telerobotic Microsurgery", Micro Dexterity Systems, Inc., Charles Retina Institute, Memphis, TN, and JPL/NASA, California Institute of Technology, Pasadena, CA, 1997.

[8] S. E. Everett and R. V. Dubey, "Sensor-Assisted Variable Trajectory Mapping for Telerobotic Task Execution", in Proc. of IEEE International Conference on Robotics and Automation, 1998.

[9] T. Massie and J. Salisbury, "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects," in Proc. ASME Winter Annual Meeting, Vol. 55-1, New Orleans, LA, 1994, pp. 295–300.

[10] N. Costescu, M. Loffler, E. Zergeroglu, D. Dawson, "QRobot – A Multitasking PC Based Robot Control System", Microcomputer Applications Journal Special Issue on Robotics, Vol. 18 No. 1, pages 13-22.

[11] N. Pernalete, "Development of an Intelligent Robotic Haptic Interface to Perform Vocational Tasks by Individuals with Disabilities", PhD Thesis Dissertation, University of South Florida, Tampa, FL, U.S.A. 2001.

[12] Yu, Wentao, "Intelligent Telerobotic Assistance for Enhancing Manipulation Capabilities of Persons with Disabilities", PhD Thesis Dissertation, University of South Florida, Tampa, Florida, U.S.A. 2004.

[13] P. Leskovsky, M. Harders, and G. Szekely, "Assessing the Fidelity of Haptically Rendered Deformable Objects", IEEE Haptics Symposium, Virginia, USA, March 25-26, 2006, pp 19-25.

[14] S. E. Salcudean, N. M. Wong, and R. L. Hollis, "Design and Control of a Force-Reflecting Teleoperation System with Magnetically Levitated Master and Wrist", IEEE Transactions on Robotics and Automation, Vol. 11, No. 6, December 1995, pp 844-858.

[15] R. Krten, 2001, "Getting Started with QNX Neutrino 2: A Guide for Realtime Programmers", Parse Software Devices, Ontario, Canada, ISBN 0-9682501-1-4.

[16] OpenHaptics Toolkit, Programmers Guide, Version 1.02, SensAble Technologies, Inc. Constitution Way, Woburn, MA 01801, USA, 2004.