

# New Control Architecture Based on PXI for a 3-Finger Haptic Device Applied to Virtual Manipulation

Ignacio Galiana, Jose Breñosa, Jorge Barrio, and Manuel Ferre

**Abstract.** To perform advanced manipulation of remote environments such as grasping, more than one finger is required implying higher requirements for the control architecture. This paper presents the design and control of a modular 3-finger haptic device that can be used to interact with virtual scenarios or to teleoperate dexterous remote hands. In a modular haptic device, each module allows the interaction with a scenario by using a single finger; hence, multi-finger interaction can be achieved by adding more modules. Control requirements for a multifinger haptic device are analyzed and new hardware/software architecture for these kinds of devices is proposed. The software architecture described in this paper is distributed and the different modules communicate to allow the remote manipulation. Moreover, an application in which this haptic device is used to interact with a virtual scenario is shown.

**Keywords:** Haptic devices, Remote manipulation, Teleoperation, Multifinger devices.

## 1 Introduction

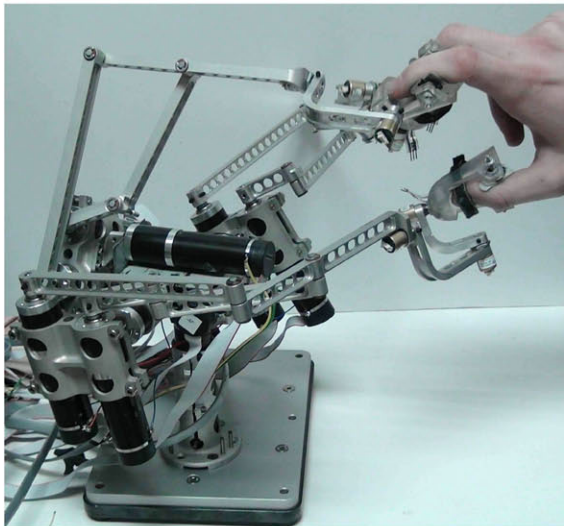
Haptic devices are mechatronic systems designed to exert forces to a human user that mimic the sensation of touching or manipulating real objects. Usually, kinesthetic haptic interfaces not only exert forces to the user, but also capture the position and movements of the hand or fingers. From another point of view, haptic devices can be understood as a bidirectional communication channel between a human and a machine, where the information involved in a manipulation task is exchanged as forces or attitudes. Not only they are able to supply big amounts of information to the end-user, but also this information is delivered in a very intuitive, natural and effective way: forces appeared in the virtual or remote environment are reflected as forces to the end-user. Thanks to this, haptic interfaces offer exceptional opportunities in the field of virtual reality. Immersion in virtual environments gets highly enhanced when taking advantage of the user's haptic perception.

In order to exploit these intrinsic features that haptic interfaces have, it is very important to allow the user to interact with the virtual environment by means of several fingers: It seems obvious that exploring or manipulating a scenario using one finger

results in a much poorer experience than doing it with the whole hand. But allowing multi-finger haptic interactions is not a simple task, so the vast majority of the commercially available devices are designed for one finger which can lead to interesting applications related to exploring the virtual or remote environment [1],[2],[3], but may not be enough for precise manipulation such as grasping.

An immediate solution could be the use of off-the-shelf haptic interfaces. However, it is possible to observe that nearly all of the commercially available devices were conceived for a single finger or contact point. It is a well known problem [4] that, when trying to put together two or more conventional haptic devices, many problems come up because of collisions between mechanical structures, greatly reducing the workspace and making it non-viable. This is a well known problem that led researchers to design specific multifinger devices i.e. [5],[6],[7],[8],[9],[10],[11].

In the following sections, new control architecture that satisfies all requirements for a 3-finger haptic device is presented and described. In section II, a quick introduction to the haptic interface will be done, and the basic requirements will be summarized, then, in section III the chosen electronic hardware for implementing the controllers will be described. An example of a 3-finger virtual manipulation is described in section IV and, finally conclusions are summarized in section V.



**Fig. 1.** Multifinger Haptic Interface. Force feedback is generated in all directions for three fingers.

## 2 System Requirements

In this section, the haptic device used will be briefly described, most relevant requirements for haptic devices in general and some specific requirements for this device will be listed.

## **2.1 3-Finger Haptic Device**

The haptic device that will be used for this study has been designed as a modular system in which each finger is one module. Each module has 6 Degrees of Freedom (DoF) as shown in Fig.1. 3 of them are actuated to exert forces in any direction and the last 3 DoFs, corresponding to the gimbal structure, are just measured and allow the user to orientate the finger in any direction.

The configuration of these three modules to conform the three finger haptic device is a result of an ergonomic study that assures a comfortable workspace for the user minimizing the collisions between each module and guaranteeing a force of 3N in any direction in every point of the total workspace. More details about this study can be found at [11].

The resulting haptic device has a total of 18 DoF, which results in a very high number of signals and in demanding computational requirements to assure a realistic interaction with the virtual environment.

## **2.2 General Requirements for Haptic Devices**

Haptic devices allow users to interact with virtual scenarios and to perceive the changes produced on it by means of their different senses. For the system presented in this paper, users can receive multimodal information as a combination of images (visual), sounds (aural) and force feedback (haptic). In order to interact with the user in a realistic way, different considerations have to be taken into account for each sense or mode (visual, aural and haptic).

It is well known that 50Hz frequency is required to reach smooth transitions in graphics. So, visual information should be refreshed at least at 50Hz to get realistic monocular graphics, if stereoscopic vision is required, this frequency should be increased to at least 100Hz. Force feedback has the higher requirements for the control system not only to provide enough realism but also to assure stability, 1 KHz update rate is necessary to achieve the required haptic fidelity [12].

In order to satisfy these requirements, a real-time control architecture has to be used.

## **2.3 Specific Requirements**

The control design architecture described in next sections is designed for a modular 3-finger haptic device called Masterfinger-3. This device is a three-finger haptic device designed to manipulate objects in virtual environments or to control robotic multi-finger hands [11].

To better understand the chosen control architecture, in this section, all the signals that have to be processed by the controller will be described.

Each finger module consists of a five-bar linkage mechanism with three actuated degrees of freedom (DoF) that allow exerting forces in any direction within the workspace, and three non-actuated DoFs at the end-effector that allow a free orientation of

the finger tip with no torque transmission thanks to a gimble configuration [4], [11] shown in Fig.1.

In addition, magnetic encoders are placed in the gimbal to determine the fingers orientation and some force sensors are placed in the thimble to monitor the forces perceived by the user when interacting with the virtual environment [13][14].

To sum up, every module has the following sensors:

- 3 motors each with each its PWM signal, current measurement and encoder.
- 3 magnetic sensors at the gimbal.
- 4 force sensing resistors at the thimble.

Hence, the controller has to manage the following input and output signals to control the system:

### **PWM.**

To actuate the DC motors, a PWM is calculated using the duty cycle provided by the PowerPC and a 10-MHz clock signal, current measurement is also needed to close the current loop.

This PWM is sent to the motors through their power wires (positive and negative in each motor). For this purpose, two output power connections are needed.

### **Motor Encoder.**

The encoder provides a simple square signal that is further processed by the control system. There are three channels (A, B, I) and two power wires (Vcc, GND).

Channels A and B are phase shifted signals that are used to determine the direction of rotation. Channel I (Index) is used as reference point for precise determination of rotation angle.

The line driver produces complementary signals of each channel, so a total of eight wires are needed to manage encoders: two power outputs and six signal inputs.

### **Gimble encoders.**

The orientation of the fingertip is obtained by magnetic encoders located at the gimble's rotational axes.

The encoders used for this system are MA3-P12 from US Digital. The MA3 is a miniature rotary absolute magnetic shaft encoder that reports the shaft position over 360° with no stops or gaps.

MA3-P12 produces a 12-bit PWM output, so it adds two output power connections to the system requirements and one signal input for the PWM angular position.

### **Force Sensors (output).**

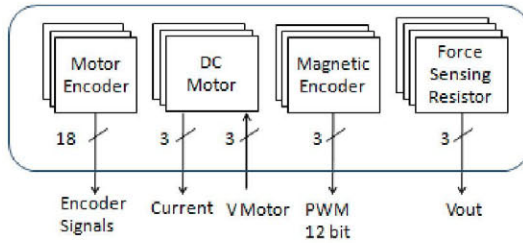
The contact sensors located inside the thimbles provide information about the force perceived by the user.

Four Tekscan's FlexiForce A201 are used in each thimble, one of them located under the finger, one at the end of the finger, and one on either side of the finger. The

FlexiForce sensing area is treated as a single contact point to estimate the normal and tangential forces perceived by the user [11].

This resistive sensor needs one voltage input and a voltage output for measuring the force; therefore a power output and one input signal are needed.

As shown in fig.2, a total of 27 signals for sensor information and 3 control signals are required to manage each finger. Three fingers compose this device, so a total of 81 signals have to be managed by the control system. The signals of the 3-finger haptic device will be processed in a single computer.



**Fig. 2.** Control Signals for one module of the 3-finger Haptic Device

As the system interacts with a virtual collaborative environment, a distributed architecture is required. In this architecture, each module is connected using an Ethernet connection with an IP/UDP protocol when in a local network or the IP/BTP [15] protocol when in a nonlocal network. This connection will satisfy the frequency required by MasterFinger3 and should also be easy to install, maintain and update.

The equipment used must be powerful enough to process all the required signals and to close the control loop with a frequency that guarantees a good performance of the system (1kHz) [12].

It has to be also considered that although electronics could be designed specifically for one module and managed with a centralized control, it is preferable to use a single computer where everything can be integrated more easily and establish connections among different degrees of freedom via software.

The following sections will address the hardware chosen for the control equipment and the software architectures designed for that purpose.

### 3 Electronic Hardware

Due to the high system requirements for Multifinger haptic applications and to the high number of signals (81) that have to be managed and controlled, selecting the correct hardware is essential to guarantee a high performance.

For this three fingered haptic device a PXI chassis system from National Instruments (Fig.3.) [16] was chosen. This system consists on three parts: A Real Time Controller, a FPGA and the power electronics.

Both the FPGA and the Real Time Controller can be programmed in LabView graphical development environment thus reducing the overall programming time for a non-expert in VHDL hardware definition language and C/C++ programming.

While the user is interacting with the virtual scenario by means of the described Multifinger interface, the Scenario Server calculates the resulting interaction force that has to be exerted to the user.

The interaction Force with ‘the virtual world’ is calculated by a laptop running Windows by using the PhysX module by NVIDIA which has the advantage that most of the physical calculus can be processed very fast by the GPU.

### 3.1 Real Time Controller

The PXI chassis incorporates a 1.9 GHz Dual-Core RealTime embedded controller NI PXIe-8102 running VxWorks.

This Interaction force calculated by the Scenario Server is sent to the Real-Time Controller by the UDP interface.

The controller is used to program the complex mathematic calculations (Jacobian, Kinematic Equations, Controller Equations, etc.) to transform the interaction force to a current setpoint in the actuators and to calculate the position (x,y,z) from the encoders position of the actuators.

The RT Controller communicates with the FPGA via a MXI high-speed cable connection to receive the encoders’ position and to send the current set point.

### 3.2 FPGA

The PXI Chassis communicates via a high-speed MXI connection with a FPGA Virtex-5 integrated in a compact module that allows connecting up to 14 modules with different functionalities directly to the FPGA (Acquisition of Analog and Digital Signals, power electronics, etc.) (Fig.4.).



Fig. 3. PXI System with RT Controller

The FPGA Virtex-5 is configured using LabView graphic programming language. For this application, the FPGA is used to:

1. Acquire and process Signals: encoders to calculate position and velocity of the device and current measurement.
2. Actuators' Current Loop: A PI controller is implemented to control the current of the actuators and generate a PWM in each actuator to exert to the user the required force.
3. Communication with the Real Time Controller: The encoder position of the actuators is sent to the RealTime Controller and the Current Command required to exert the interaction force to the user is received from the RT Controller.



**Fig. 4.** FPGA Virtex-5 with Chassis to connect different modules

### 3.3 Power Electronics

For each actuator a Full H-Bridge Brushed DC Servo Drive Module (Fig.5.) from NI directly connected to the FPGA was used.

Each of these modules provides a measurement of the current that circulates over the DC actuator, has a data acquisition for the Encoder and provides power to the DC actuators used.



**Fig. 5.** Full H-Bridge Brushed DC Servo Drive Module

## 4 Software Architecture

Software architecture design is an important decision that must be taken carefully when a new platform is developed. Decisions as the union between haptic and visual interfaces, or the software election or how the virtual scenarios are recreated, should be analyzed. Also, software architecture should be as independent as possible from the hardware architecture.

There are two important parts in the architecture proposed:

1. Low level software in charge of controlling the haptic device.
2. High level software, responsible for updating the scenario, calculating force feedback and processing visual and aural information.

The suggested solution shown in Fig.6., splits the software into different blocks that run in the different electronic devices as described in the previous section and communicate via an Ethernet connection.

The software solutions proposed for low level blocks have to be developed ad hoc, as they are very dependent on the device's hardware and it was a development by the UPM. Meanwhile, a wide range of commercially available software products can be used for high-level blocks, as they are performed in an abstract level.

In the next paragraphs, explanations on which, why and how this blocks are implemented are shown.

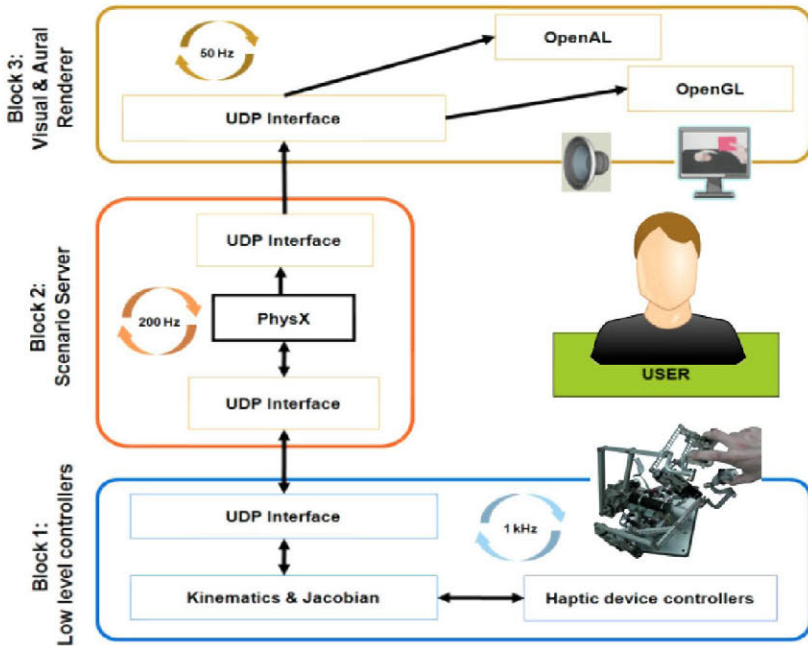


Fig. 6. Software architecture for controlling a 3-finger haptic device

### Block 1: Low level controllers.

This block is in charge of processing the measurements from the sensors, calculating the attitudes of the end-effector of each independent module, and controlling the motor currents and torques, and communicating via Ethernet with the immediately upper block.



In order to avoid unwished vibrations in the thimbles, this software must work at 1kHz [12].

Furthermore, this block synchronizes all the other processes and performs kinematic and jacobian calculations. Once the attitudes of the different thimbles are calculated, they are sent to the upper block, the physics engine.

In return, the forces to be reflected are received from the physics engine and the torques to be produced in each motor are calculated by means of the Jacobian calculations.

Delays in this block can lead to inconsistent situations where real fingers are in very different positions from the virtual ones. This can make the haptic interface become unstable, reflect wrong forces, vibrations, etc. Hence, a hard real time system is required here. For this purpose, National Instruments' hardware runs VxWorks.

The low level controller runs in the PXI platform that, as described in section 3 consists of a Real-time controller to communicate with the other modules via Ethernet and a FPGA to control the system and acquire the different signals.

The Low-level controller has been programmed using LabView graphical development environment (LabView RealTime module for the RT Controller and LabView FPGA module for the Virtex-5 FPGA).

## **Block 2: Scenario server.**

In this module, very heavy calculations are processed. Here, the virtual scenario is created and updated with the data received from the lower block.

Collision detection, interaction calculations and body dynamics are a few of the tasks assigned to this block. It also has to communicate the scenario status to the graphics interface and the interaction forces to be exerted to the virtual fingers to block 1.

The physics engine is in charge of: identifying collisions, estimating external forces resulting from the collisions (including frictional forces) and to produce a correct response to the residual forces [17].

Due to the high requirements of the physics engine, implementing ad hoc solutions for this block is very time-consuming and requires big development efforts. To tackle this problem, the adopted solution was to study the software kits commercially available and try to take advantage of their features. In literature, haptic interfaces using commercial physics engines are not very common. This happens mainly because the existing physical engines are often focused on a small set of physical laws and they are usually implemented aiming very accurate but computationally demanding behaviors that cannot be executed in real time.

Some others physics engines can be found working in medical haptic simulators [18]. Some other engines provide a framework where new physical laws can be implemented and included by using software containers.

Related to videogames, many physics engines have been developed recently. They are not optimized for haptics, but they usually have some very interesting features. It is very common to find fluids in videogames, complex rigid bodies and even soft bodies, which gives an idea about the potential of these tools applied in the field of haptic simulation. Some of these engines were tested: ODE, Bullet (which also have

the advantage of being open source), Havok and PhysX. Finally PhysX from Nvidia was used because it allocates most of the calculus in the GPU which leads to faster simulations.

### Block 3: Graphic and Aural renderer.

This layer's functionality consists on drawing the virtual objects and fingers and reproducing sounds. As it has been said before, it receives the data from the scenario server.

There are several possibilities regarding software for programming the visual interface, for this system two graphic API's were taken into account: OpenGL and DirectX. DirectX is an API developed by Microsoft and full features (as hardware acceleration) are only obtained when running on Windows based OS. However, OpenGL is an open source API that works under nearly any OS. Its features are enough to meet our demands and it also offers high compatibility, something which is very convenient for this project. For the same reasons, OpenAL was chosen for the aural interface.

## 5 Applications

Following the control structure described in the previous section, this 3-finger haptic device can be also used to manipulate objects in virtual environments or for teleoperating dexterous robotic hands,

In both situations the device control architecture is the same, by just changing the blocks 2 and 3 of the software architecture for a telemanipulator with a 3-finger robotic gripper and their corresponding controllers.

The haptic mounted and connected to a virtual environment allows to control three contact points, interacting with different objects and feeling the shape and reflected forces.

The following figure shows the manipulation of objects with three fingers in the developed virtual environment.

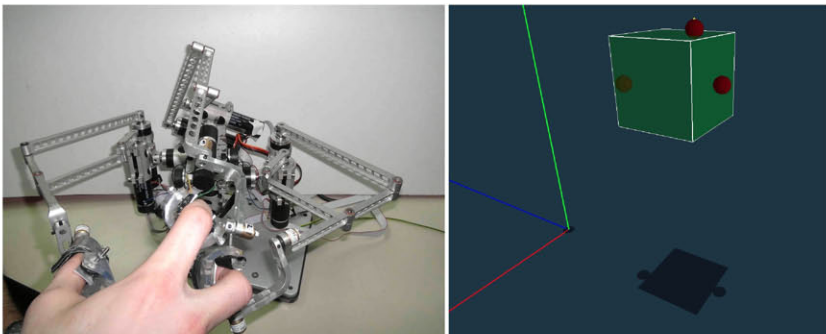
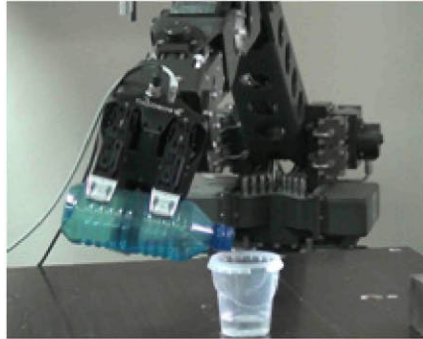


Fig. 7. 3-Finger haptic device to perform advanced virtual manipulation

Future application will focus on using the described device to control a 3-finger robotic hand connected to an industrial manipulator in order to perform precise manipulation tasks; an example of this is shown in Fig.8



**Fig. 8.** 3-Finger hand Robotiq connected to a Kraft Grips manipulator. Future application will focus on using the developed device to control this system.

## 6 Conclusion

Controlling Multifinger haptic devices adds very high requirements to the system in terms of control frequency, number of signals that are sent over the network and high computation that has to be done in Real Time to assure the required high fidelity of the haptic interaction.

To control this kind of devices, modular software architecture is proposed in this article that divides the problem in different systems: i) the low level controller for each module and the redundant axis considering them as a complete haptic system and ii) the virtual environment supported by the scenario server for calculating the physics interaction and the Visual and Aural rendering module.

The Hardware architecture has also different modules and consists of an FPGA for the low-level controller and a RT controller for managing all the complex calculations (jacobian, kinematics, etc.); this provides a compact and scalable solution for the required high computation capabilities assuring a correct frequency rate for the control loop (1 kHz).

A virtual scenario has been developed using the proposed architecture to manipulate a virtual box by using 3-fingers. Future works will focus on performing real tasks in remote environments by teleoperating dexterous robotic hands.

**Acknowledgments.** This work has been partially supported by Madrid Community in the framework of The IV PRICIT through the project TECHNOFUSION(P2009/ENE/1679), the TEMAR project under grant DPI2009-12283 from the Spanish Ministry of Science and Innovation (MICINN) and UPM under 'Formación de Personal Investigador'.

## References

1. Haption, Inc., Haption-Products (April 2012), <http://www.haption.com/site/index.php/en/products-menu-en>
2. Sensable, Sensable-Products (April 2012), <http://www.sensable.com/products-haptic-devices.htm>
3. Massie, T.H., Salisbury, J.K.: The phantom interface: a device for probing virtual objects. In: Proc. ASME Winter Annual Meeting, Symposium on Haptic Interfaces for a Virtual Environment and Teleoperator Systems (1994)
4. Cerrada, P., Breñosa, J., Galiana, I., López, J., Ferre, M., Giménez, A., Aracil, R.: Optimal Mechanical Design of Modular Haptic Devices. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2011, Budapest, Hungary (2011)
5. Garcia-Robledo, P., Ortego, J., Barrio, J., Galiana, I., Ferre, M., Aracil, R.: Multifinger Haptic Interface for Bimanual Manipulation of Virtual Objects. In: IEEE International Workshop on Haptic Audio Visual Environments and Games, HAVE (2009)
6. García-Robledo, P., Ortego, J., Ferre, M., Barrio, J., Sánchez-Urán, M.A.: Segmentation of Bimanual Virtual Object Manipulation Tasks Using Multifinger Haptic Interfaces. IEEE Transactions on Instrumentation and Measurement 60(1), 69–80 (2011)
7. Bouzit, M., Popescu, G., Burdea, G., Boian, R.: The Rutgers Master II-ND Force Feedback Glove. In: Proc. IEEE Haptics Symposium, Orlando (March 2002)
8. Frisoli, A., Simoncini, F., Bergamasco, M.: Mechanical Design of a Haptic Interface for the Hand. In: Proc. ASME Design Engineering Technical Conf. and Computer and Information in Engineering Conf., Montreal, Canada, September 29–October 2 (2002)
9. Gosselin, F., Jouan, T., Brisset, J., Andriot, C.: Design of a wearable haptic interface for precise finger interactions in large virtual environments. In: Proc. Int. Conf. World Haptics 2005, Pisa, Italy, March 18–20, pp. 202–207 (2005)
10. Endo, T., Yoshikawa, T., Kawasaki, H.: Collision Avoidance Control for a Multi-fingered Bimanual Haptic Interface. In: Kappers, A.M.L., van Erp, J.B.F., Bergmann Tiest, W.M., van der Helm, F.C.T. (eds.) EuroHaptics 2010. LNCS, vol. 6192, pp. 251–256. Springer, Heidelberg (2010)
11. Breñosa, J., Cerrada, P., Ferre, M., Aracil, R.: Design of an Ergonomic Three-Finger Haptic Device for Advanced Robotic Hands Control. In: IEEE-World Haptics Conference (WHC), Istanbul, Turkey (2011)
12. Hannaford, B., Okamura, M.: Haptics. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, ch. 20, p. 720. Springer (2008) ISBN: 9789-3-540-23957-4
13. Galiana, I., Bielza, M., Ferre, M.: Estimation of Normal and Tangential Manipulation Forces by Using Contact Force Sensors. In: Kappers, A.M.L., van Erp, J.B.F., Bergmann Tiest, W.M., van der Helm, F.C.T. (eds.) EuroHaptics 2010, Part I. LNCS, vol. 6191, pp. 65–72. Springer, Heidelberg (2010)
14. Monroy, M., Ferre, M., Barrio, J., Eslava, V., Galiana, I.: Sensorized Thimble for Haptic Applications. In: IEEE International Conference on Mechatronics, Málaga, Spain (2009)
15. Wirz, R., Marin, R., Ferre, M., Barrio, J., Claver, J.M., Ortego, J.: Bidirectional Transport Protocol for Teleoperated Robots. IEEE Transactions on Industrial Electronics 56(9) (2009)
16. National Instruments (April 2012) (Online), <http://www.ni.com/pxi/>
17. Melder, N., Harwin, W., Sharkey, P.: Translation and rotation of multi-point contacted virtual objects. In: Proceedings of the World Haptics Conference, pp. 218–277 (2003)
18. Nourian, S., Shen, X., Georganas, N.D.: XPHEVE: An Extensible Physics Engine for Virtual Environments. In: Canadian Conference on Electrical and Computer Engineering, CCECE 2006, pp. 1546–1549 (May 2006)