We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Analysis and Experimental Study of a 4-DOF Haptic Device

Ma and Payandeh
Simon Fraser University
Canada

1. Introduction

This chapter presents a new configuration of a haptic device based on a 4-DOF hybrid spherical geometry with a design of a distributed computational platform. The Spherical Parallel Ball Support (SPBS) type device, as it is referred to, consists of a particular design feature with the intersecting joint axes of both active and passive spherical joints. The orientation of the device is determined through the active spherical joint using a special class of spherical 3-DOF parallel geometry. In addition, the passive spherical joint (ball and socket configuration) is introduced in the design to increase the mechanical fidelity of the device. A new forward and inverse kinematics analysis are presented. The study is motivated by deriving a mathematical model and a closed-form solution of the kinematics of the proposed device configuration. A novel desktop computational platform is proposed and studied for creation of haptic feedback. The experimental studies are presented by specifying and demonstrating a virtual wall as a performance-benchmarking tool and aimed to verify a desirable update rate of 1 kHz or more for the haptic effect. A computational algorithm is devised by using a reconfigurable experimental setup that would achieve the desired haptic effect. This framework consists of the 4-DOF prototype, a host computer, and a data acquisition system which uses a microprocessor, a FPGA, integrated circuits and pulse-width-modulation. Such architecture can offer a novel distributed system for tele-operation over the Internet and haptic rendering of deformable objects. Through analysis of the experimental setup, key parameters have been identified for synthesis of a future tele-operation environment.

1.1 Objectives

The major objective of this research is to design, develop and experiment with a novel haptic hardware environment. The design of this architecture and its building blocks are aimed to be reconfigurable, programmable, portable and scalable such that it supports the integration towards a distributed system framwork of a surgical simulator. In this work, we first specified the haptic effect that we wanted to create as an objective. We then demonstrated a virtual wall as a performance-benchmarking set-up and devised a computational algorithm to verify the desirable update rate for the haptic effect using our custom-designed experimental setup and the force-feedback device (Ma & Payandeh, 2008).

The Spherical Parallel Ball Support type device is a prototype based on a 4-DOF hybrid spherical geometry. The orientation of the device is determined by the mobile platform on the active spherical joint using a special class of spherical 3-DOF parallel geometry. In parallel mechanism configuration, the moving end effector is connected to a fixed reference base via multiple kinematic chains. Any two chains thus form a closed kinematic chain which is different in topology in comparison with open loop mechanisms such as the serial robotic arm. Parallel robots (such as the Stewart platform and the Delta robot) usually have wider mechanical bandwidth than traditional articulated robots. This is due to the location of the actuators which can be mounted on the supporting base which as a result can reduce the floating mass of the mechanism. However, it is also known that computational complexities involved in obtaining various kinematic solutions such as forward kinematics can result in more than one unique solution. Our study is motivated by deriving a computational model which can result in a closed-form solution of the kinematics for our proposed haptic device configuration.

In addition, we designed and developed a modular and distributed scheme aiming at a parallelization of the main components of haptic interaction tasks (haptic rendering). We present the design and performance study of a data acquisition system (DAS) which is adapted into our framework. The DAS is reconfigurable and capable of controlling the SPBS haptic device at a fast update rate. The local interconnection framework consists of the host control computer, the custom-designed data acquisition system, and the haptic device. The UDP/IP and TCP/IP socket interface are used for communications between the DAS and a host computer in order to collect performance benchmarking results.

1.2 Contributions
The major contributions of the research are summarized below.
Analysis of a 4-DOF haptic device and derivation of a closed-form solution: a mathematical model and analysis of a new device geometry/configuration is presented. The forward and inverse kinematic solutions and static force mapping are derived.

A distributed computational system framework: a novel desktop computational platform for haptic control of the device is developed. Such architecture can offer a novel distributed system for tele-operation over the Internet and haptic rendering of deformable objects using medical imaging data. In addition, software application development is designed to target multiple operating systems support. This provides the flexibility of the targeted operating systems (Windows, Linux, Solaris, etc.) for running the virtual environment (GUI, graphics, and haptics rendering).

2. Analysis of the Haptic Mechanism

In this section, we present the development and experimental results of the Spherical Parallel Ball Support type mechanism (Li & Payandeh, 2002). The distinctive feature of SPBS is that it uses a 4-DOF hybrid spherical geometry (see Fig. 1). The objective is to take advantage of the spherical 3-DOF parallel geometry as the supporting platform. Hence, the orientation of a stylus is determined by pure rotation of the platform in its workspace while the translational motion of the haptic handle is supported by a prismatic joint. Unlike model

presented in (Gosselin & Hamel, 1994), in this design, the rotational axes of the three actuators are coplanar. The center of the sphere is located below the mobile platform where the haptic gripper handle is connected and also located at this center a passive ball/socket supporting joint. This joint is used for supporting both the resultant user interactive forces and also the static weight of the parallel spherical mechanism. The kinematic architecture and geometric parameters of SPBS are presented first. In order to understand the kinematics of the mechanism, a closed-form solution for the forward and inverse kinematics is developed.
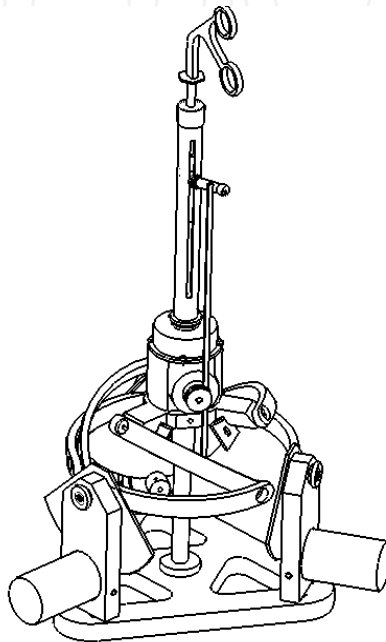
Fig. 1. Design model of the hybrid 4-DOF haptic mechanism.

2.1 Kinematic model

The architecture of SPBS consists of a particular design using an active/passive spherical joint and an active translational joint. The active spherical joint supports a moving platform connected to a fixed base via a spherical parallel mechanism configuration. There are three symmetrical branches which result in a total of nine revolute joints. Each branch has one active joint. Specifically, the mechanical structure of one of the three branches contains an actuator, an active cam, an active link and a passive link. The off-centre gripper handle is attached to the moving platform via a prismatic joint which constitutes to an additional translational degree of freedom. The rotational axes of all nine revolute joints intersect at a common point "O" known as the center of rotation of the mechanism (this point is also the center of the passive spherical joint in the form of a ball/socket configuration). For purposes of legibility only one of the three branches is shown in Fig. 2.

Geometrically, the base and the moving platform can be thought of as two pyramidal entities having one vertex in common at the rotational center "O". The axes of the revolute joints of the base and of the mobile platform are located on the edges of the pyramids. For purposes of symmetry, the triangle at the base of each pyramid is an equilateral triangle.
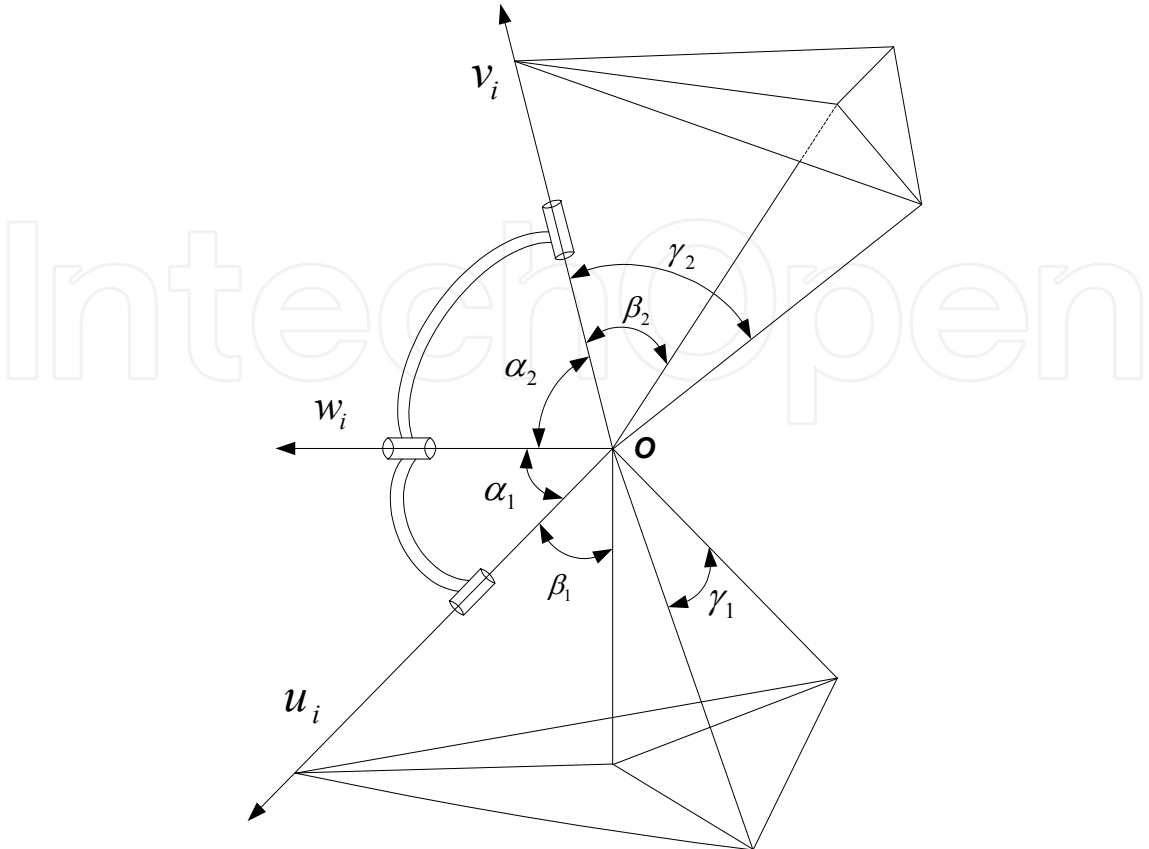
Fig. 2. Geometric parameters of a spherical 3-DOF parallel mechanism.

Let angle γ1 be the angle between two edges of the base pyramid, angle γ2 be the angle between two edges of the mobile platform pyramid, and angle βi i = 1, 2 be the angle between one edge and the vertical axis. The angles are related through the following equation (Craver, 1989):

$$\sin \beta_i = \frac{2\sqrt{3}}{3} \sin \frac{\gamma_i}{2}, \quad i = 1, 2 \tag{1}$$

In addition, angles α1 and α2 represent the radial length associated with the intermediate links. The designs presented in (Gosselin & Hamel, 1994) and (Birglen et al., 2002) use a special class of the geometry which lead to a simplification of the forward kinematics problem. The geometry of SPBS also takes into account some implicit design by explicitly defining coplanar active joints. This results in the following geometric parameters being used in the design of SPBS, namely, α1 = 90°, α2 = 90°, γ1 = 120°, and γ2 = 90°, respectively.

It has been shown that for the general case the forward kinematic problem can lead to a maximum of eight different solutions. One isotropic configuration has been studied in order to obtain an optimized solution of the kinematic problem. Other approaches have been considered in the past using numerical solutions such as artificial neural networks and polynomial learning networks (Boudreau et al., 1998) to solve the kinematic problem. In the

following, a new closed-form algebraic solution of the inverse and forward kinematics problem of the configuration used by SPBS is presented.

Let u_i $i = 1, 2, 3$ be a unit vector (see Fig. 2) defining the revolute axis of the ith actuator. Let η_i $i = 1, 2, 3$ be an angle measured from u_1 to u_1, u_2 and u_3, respectively. The schematic of SPBS and the reference coordinate frame are shown in Fig. 3. By symmetry, $\eta_1 = 0°$, $\eta_2 = 120°$, and $\eta_3 = 240°$, the following can be defined:

$$u_i = \begin{bmatrix} 0 & -\sin \eta_i & \cos \eta_i \end{bmatrix}^T \tag{2}$$
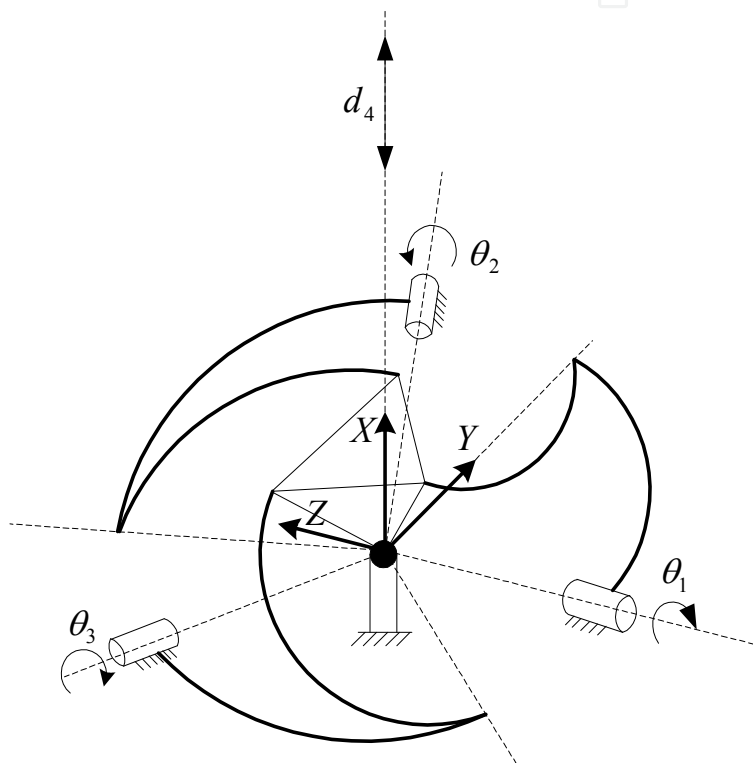
Fig. 3. Schematic of SPBS and the reference coordinate system

Let θ_i $i = 1, 2, 3$ be the rotation angle of the ith actuator. Then, vector w_i $i = 1, 2, 3$ can be defined as a unit vector associated with the revolute joint between the passive link and the active link. Using standard transformation matrices, we can obtain

$$w_i = \begin{bmatrix} -\sin \theta_i & \cos \eta_i \cos \theta_i & \sin \eta_i \cos \theta_i \end{bmatrix}^T \tag{3}$$

Similarly, vector v_i $i = 1, 2, 3$ can be defined as a unit vector along the axis of the ith revolute joint on the mobile platform. Since each of these axes make an angle $\gamma_2 = 90°$ with the others, an orthonormal coordinate frame can be attached to the mobile platform for describing its orientation relative to the reference coordinate frame.

We introduce the rotation matrix Q in order to describe the instantaneous orientation of the mobile platform with X-Y-Z fixed angles rotation. Hence, three successive rotations are defined by a rotation of angle ϕ_3 about the X-axis, a rotation of angle ϕ_2 about the Y-axis, and a rotation of angle ϕ_1 about the Z-axis (see Fig. 4). Let v_{o1} = X, v_{o2} = Y, and v_{o3} = Z, respectively. The orientation of the mobile platform can be expressed as

$$Q = \begin{bmatrix} c\phi_1 c\phi_2 & c\phi_1 s\phi_2 s\phi_3 - s\phi_1 c\phi_3 & c\phi_1 s\phi_2 c\phi_3 + s\phi_1 s\phi_3 \\ s\phi_1 c\phi_2 & s\phi_1 s\phi_2 s\phi_3 + c\phi_1 c\phi_3 & s\phi_1 s\phi_2 c\phi_3 - c\phi_1 s\phi_3 \\ -s\phi_2 & c\phi_2 s\phi_3 & c\phi_2 c\phi_3 \end{bmatrix} \quad (4)$$

$$v_i = Q v_{oi}, \quad i = 1, 2, 3 \quad (5)$$

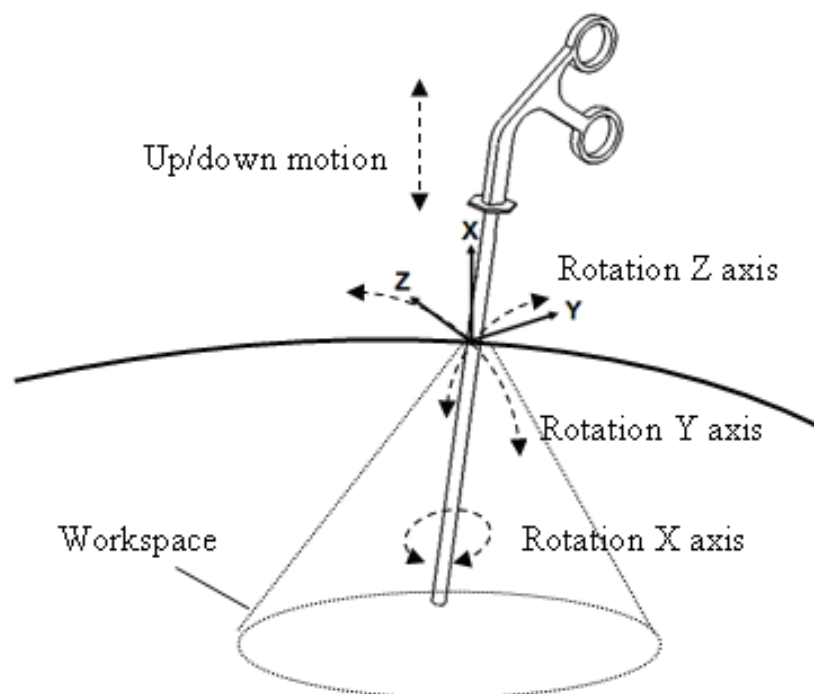where $c\phi_i$ and $s\phi_i$ stand for $\cos\phi_i$ and $\sin\phi_i$.

Fig. 4. X-Y-Z fixed angles rotation relative to the reference coordinate system.

2.1.1 Derivation of Inverse Kinematics

Suppose the vector components v_{ix}, v_{iy}, v_{iz} for i = 1, 2, 3 specify a known orientation of the mobile platform relative to the reference frame.

$$v_i = \begin{bmatrix} v_{ix} & v_{iy} & v_{iz} \end{bmatrix}^T \quad (6)$$

The inverse kinematic solution can be obtained through solution to the following equation

$$w_i \cdot v_i = \cos \alpha_2 \tag{7}$$

Since the vectors w_i and v_i, $i = 1, 2, 3$ are orthogonal when $\alpha_2 = 90°$, the substitution of (3) and (6) with the geometric parameters of SPBS then lead to simple equations in the sine and cosine of the actuated joint angles,

$$\tan \theta_1 = \left(\frac{v_{1y}}{v_{1x}} \right) \tag{8}$$

$$\tan \theta_2 = \left(\frac{1}{2} \frac{\sqrt{3}v_{2z} - v_{2y}}{v_{2x}} \right) \tag{9}$$

$$\tan \theta_3 = \left(\frac{1}{2} \frac{\sqrt{3}v_{3z} + v_{3y}}{v_{3x}} \right) \tag{10}$$

2.1.2 Derivation of Forward Kinematics

The solution of the forward kinematic problem for this configuration is discussed below. Using (4) and (5), expressions of vectors v_i $i = 1, 2, 3$ as functions of the angles ϕ_1, ϕ_2, and ϕ_3 are obtained. These expressions are then substituted into (7) together with (3). This leads to three equations with the three unknown (ϕ_1, ϕ_2, and ϕ_3) as follows,

$$-s\theta_1 c\phi_1 c\phi_2 + c\theta_1 s\phi_1 c\phi_2 = 0 \tag{11}$$

$$\begin{aligned}
&-s\theta_2 (c\phi_1 s\phi_2 s\phi_3 - s\phi_1 c\phi_3) \\
&-\frac{1}{2}c\theta_2 (s\phi_1 s\phi_2 s\phi_3 + c\phi_1 c\phi_3) \\
&+\frac{\sqrt{3}}{2}c\theta_2 c\phi_2 s\phi_3 = 0
\end{aligned} \tag{12}$$

$$\begin{aligned}
&-s\theta_3 (c\phi_1 s\phi_2 c\phi_3 - s\phi_1 s\phi_3) \\
&-\frac{1}{2}c\theta_3 (s\phi_1 s\phi_2 c\phi_3 - c\phi_1 s\phi_3) \\
&-\frac{\sqrt{3}}{2}c\theta_3 c\phi_2 c\phi_3 = 0
\end{aligned} \tag{13}$$

The solution of these three equations for angles ϕ_1, ϕ_2, and ϕ_3 give the solution of the forward kinematic problem. For the special geometry of our proposed haptic design, a simpler expression for the forward kinematic problem can be obtained. In fact, because of the

definition of our fixed reference frame chosen here and the choice of the fixed angles rotation sequence, eq. (11) can be solved for:

$$\phi_1 = \theta_1 \quad or \quad \phi_1 = \theta_1 + \pi \tag{14}$$

Once ϕ_1 is determined, equations (12) and (13) can be rewritten as follows

$$A_1 c\phi_3 + B_1 s\phi_3 = 0 \tag{15}$$

$$A_2 c\phi_3 + B_2 s\phi_3 = 0 \tag{16}$$

where

$$A_1 = s\theta_2 s\theta_1 - \frac{1}{2} c\theta_2 c\theta_1 \tag{17}$$

$$B_1 = -s\theta_2 c\theta_1 s\phi_2 - \frac{1}{2} c\theta_2 s\theta_1 s\phi_2 + \frac{\sqrt{3}}{2} c\theta_2 c\phi_2 \tag{18}$$

$$A_2 = -s\theta_3 c\theta_1 s\phi_2 - \frac{1}{2} c\theta_3 s\theta_1 s\phi_2 + \frac{\sqrt{3}}{2} c\theta_3 c\phi_2 \tag{19}$$

$$B_2 = -s\theta_3 s\theta_1 + \frac{1}{2} c\theta_3 c\theta_1 \tag{20}$$

Since $\cos(\phi_3)$ and $\sin(\phi_3)$ cannot go to zero simultaneously, equations (15) and (16) lead to

$$A_1 B_2 - A_2 B_1 = 0 \tag{21}$$

Substituting (17) to (20) into (21) and rearranging, one can obtain:

$$C_1 (c\phi_2)^2 + C_2 (s\phi_2)(c\phi_2) + C_3 = 0 \tag{22}$$

where

$$C_1 = \frac{1}{2} c\theta_2 c\theta_1 s\theta_3 s\theta_1 + s\theta_2 s\theta_3 c\theta_1^2$$
$$+ c\theta_2 c\theta_3 - \frac{1}{4} c\theta_2 c\theta_3 c\theta_1^2 \tag{23}$$
$$+ \frac{1}{2} s\theta_2 s\theta_1 c\theta_3 c\theta_1$$

$$C_2 = \frac{\sqrt{3}}{2} s\theta_3 c\theta_1 c\theta_2 - \frac{\sqrt{3}}{2} c\theta_3 s\theta_2 c\theta_1 \tag{24}$$

$$C_3 = -s\theta_2 s\theta_3 - \frac{1}{4}c\theta_2 c\theta_3 \qquad (25)$$

Since (23) to (25) are only expressed in terms of the actuated joint angles θ1, θ2, and θ3, these coefficients in (22) can be calculated instantaneously. Four solutions can be obtained algebraically for ϕ_2 from (22). Using the two sets of solutions obtained in (14), a total of eight solutions can be obtained for ϕ_2.

Once angle ϕ_2 is determined, either (12) or (13) can be rearranged to compute ϕ_3 as follows

$$\phi_3 = \tan^{-1}(D_1 / D_2) \qquad (26)$$

where

$$D_1 = s\theta_1 s\theta_2 - \frac{1}{2}c\theta_1 c\theta_2 \qquad (27)$$

$$D_2 = c\theta_1 s\theta_2 s\phi_2 + \frac{1}{2}s\theta_1 c\theta_2 s\phi_2 - \frac{\sqrt{3}}{2}c\theta_2 c\phi_2 \qquad (28)$$

For the sets of solutions of $\phi1$, $\phi2$, and $\phi3$ that can be obtained, the computation of the rotation matrix Q will result in a maximum of eight different X-Y-Z fixed angles rotation matrices with respect to the reference frame. One of these solutions represents the orientation of the mobile platform corresponding to the input actuated joint angles. By taking into account the physical workspace of SPBS given its joint limits, the corresponding orientation can be selected among the solution set with a sufficient conditional check of $v_{1x} > 0$, $v_{2x} > 0$, $v_{3x} > 0$, and $v_{1z} < 0$.

2.2 Jacobians

In robotics, the Jacobian matrix of a manipulator, denoted as J, is generally defined as the matrix representing the transformation between the joint rates and the Cartesian velocities. For the case of a closed-loop manipulator the notion of this mapping for the direct and inverse kinematic problems are interchanged (Angeles & Gosselin, 1990). The Jacobian matrix is defined as:

$$J\omega = \dot{\theta} \qquad (29)$$

where ω is the angular velocity of the platform, $\dot{\theta}$ is the actuated joint velocity vector.

An alternative form of (29) with the matrices A and B is rewritten in variant form as,

$$A\omega = B\dot{\theta} \qquad (30)$$

where

$$A = \begin{bmatrix} (w_1 \times v_1)^T \\ (w_2 \times v_2)^T \\ (w_3 \times v_3)^T \end{bmatrix} \qquad (31)$$

$$B = \begin{bmatrix} (u_1 \times w_1 \bullet v_1) & 0 & 0 \\ 0 & (u_2 \times w_2 \bullet v_2) & 0 \\ 0 & 0 & (u_3 \times w_3 \bullet v_3) \end{bmatrix} \qquad (32)$$

Equations (30) to (32) are derived using the general case of the spherical 3-DOF parallel geometry (Angeles & Gosselin, 1990). Similarly, the equations are applicable to the geometry of the SPBS device. Equation (30) shows that the angular velocity of the end-effector can be obtained as an expression of the joint velocities. For haptic rendering purposes, the time derivatives of the rotation angles are used, expressing the angular velocity vector, ω, as

$$\omega = R \dot{\phi} \qquad (33)$$

where ϕ is the vector of the Z-Y-X Euler angles, ϕ_1, ϕ_2, and ϕ_3. The matrix R is derived by using the definition of the angular velocity tensor (a skew-symmetric matrix) and taking partial derivatives of the orthonormal matrix in (4). One can obtain,

$$R = \begin{bmatrix} 0 & -\sin\phi_1 & \cos\phi_1\cos\phi_2 \\ 0 & \cos\phi_1 & \sin\phi_1\cos\phi_2 \\ 1 & 0 & -\sin\phi_2 \end{bmatrix} \qquad (34)$$

Combining equations (29) to (33),

$$\dot{\theta} = JR\dot{\phi} = (B^{-1}A)R\dot{\phi} \qquad (35)$$

Equation (35) gives a practical relationship relating the velocities of the active joint rates as a function of an angle set velocity vector.

In addition, we would like to obtain a relationship between the input actuator torques and the output torques exerted on the end-effector about the origin O. In particular, we have the relationship,

$$\text{power} = \text{torque} \times \text{angular speed} \qquad (36)$$

where power is in watts, torque is in Nm, and angular speed is in radians per second.

Let w be the torque vector exerted by the end-effector and τ be the active joint torque vector. By using a static equilibrium model and the concept of virtual power, we equate the input and the output virtual powers and obtain the relationship in (39).

$$\mathrm{w}^T \omega = \tau^T \dot{\theta} \qquad (37)$$

$$\mathrm{w}^T \omega = \tau^T J \omega \qquad (38)$$

$$\tau = J^{-T} \mathrm{w} \qquad (39)$$

Equation (39) provides a mapping of the desired output torque vector in Cartesian space to the joint torque vector. As given by equations (30), (31) and (32), an algebraic solution can be used to compute the matrices J, A, and B, respectively. Note that the vectors u_i, w_i and v_i for i = 1, 2, 3, are all known at any instant during the device simulation. The vectors u_i correspond to the reference configuration, whereas the vectors w_i and v_i have been derived in symbolic forms in the previous inverse and forward kinematics sections. The results from this section form a set of basis equations that can be experimented with the prototype.

2.3 Static Relationship

We want to be able to compute the force exerted on the hand of the user holding the tool. A simple point/line model is used for the visualization of the physical tool (handle) of the mechanism. For example, as shown in Fig. 5, the vector r represents the position vector of the handle location at which the user would hold the tip of the tool. The triangular (yellow) surface represent the contacting surface and the force vector F represents the reaction contacting force generated by the computational model.
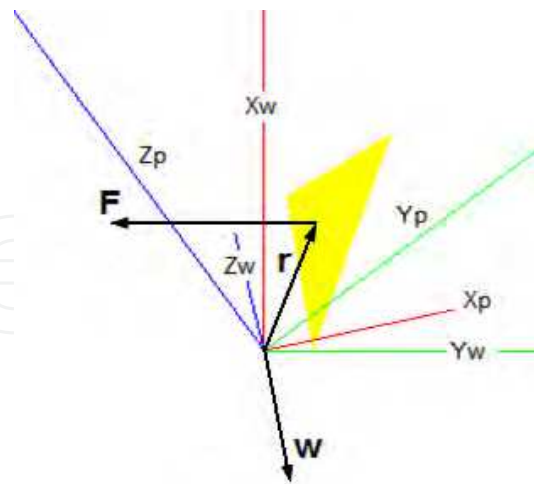

Fig. 5. Representation of contact force and moment vectors.

Fig. 6. Haptic device at equilibrium configuration shown in Figure 5.

Fig. 5 and Fig. 6 show an example orientation such that the physical tool is leaning against a virtual wall. The triangle in Fig. 5 illustrates a virtual plane (wall) defined by three arbitrary points in space. The contact point is where the position vector r intersects the plane. The vector F represents the normal force at the contact point having a direction vector perpendicular to the virtual plane. Therefore, by knowing this force vector with respect to the world coordinate frame, the moment of this force vector can be computed by using the vector cross product between r and F. The moment vector w consists of the x-y-z components which are the moments about each principal axis. This moment vector is the desired output torque vector in Cartesian space. Therefore, we can use it to resolve for the joint torque vector by equation (39). The linear force along the handle of the device is independent of the rest of the degrees of freedom and hence is solved separately. Table 1 shows a summary of the computed values of the key parameters used in this equilibrium example.

Parameter Names	Values
Joint angles (degrees)	$\theta 1 = 40.39°$, $\theta 2 = 8.42°$, and $\theta 3 = 9.43°$
Position vector (metres)	r = [0.0748, 0.0250, -0.00797]
Force vector (N)	F = [0, -813.981, 0]
Desired output torque (Nm)	w = [-6.49, 0, -60.88]
Joint torque (Nm)	τ = [- 33.636, 1.813, 44.873]
Jacobian	$J = \begin{bmatrix} 1.208 & 1.028 & 1.000 \\ 2.152 & -1.025 & -0.224 \\ 0.673 & 0.810 & -0.597 \end{bmatrix}$

Table 1. Numerical values for the example case study.

3. Computational Hardware Design

Haptic displays aim to provide operators with a sense of touch, rendering contact forces as if interactions occurring with real objects. Haptic displays are generally used in conjunction with visual displays, where objects are simulated in a virtual world. The applications of both graphical and haptic displays in virtual reality provide the user with the illusion of touching objects and a heightened sense of presence in the virtual world. The computed interactive force between the user representation and a virtual object is rendered to the user via the haptic device. Due to the elaborate sensory perception of the human hand, which registers even very small oscillations, many research studies have shown that update rates of 1 kHz or above are desired for haptic rendering of any physical rigid contact effects. Therefore, many efforts have been devoted to develop various hardware models, control schemes, and haptic rendering controllers in the past years. Our motivation is to develop a hardware setup which supports the real-time force feedback control of the proposed haptic device.

3.1 Conceptual architecture of the distributed system

Figure 7 shows a conceptual architecture of a distributed system framework. The experimental setup presented in this section includes the 4-DOF SPBS haptic device, a data acquisition system (DAS) and a personal computer (PC). Major processing components are experimented on selected hardware contributing to an integrated VR simulation. The host PC is responsible for haptic rendering and providing a Graphical User Interface. The data acquisition system that consists of a microprocessor, a FPGA, integrated circuits, pulse-width-modulation (PWM) is described in the following.
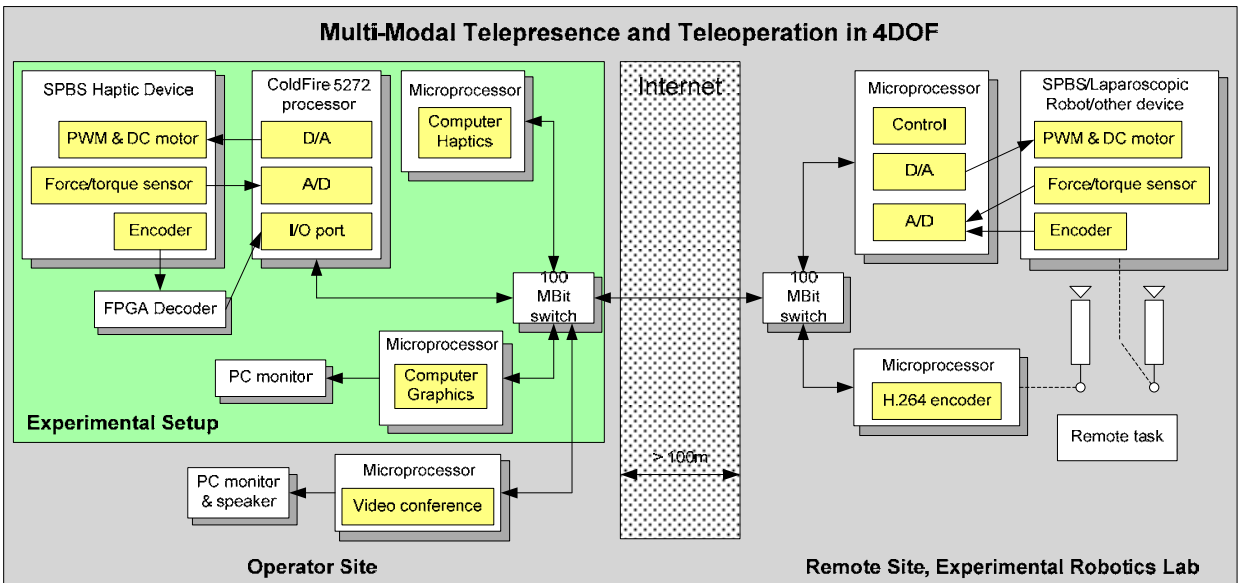

Fig. 7. System architecture of a conceptual microprocessor-based distributed system.
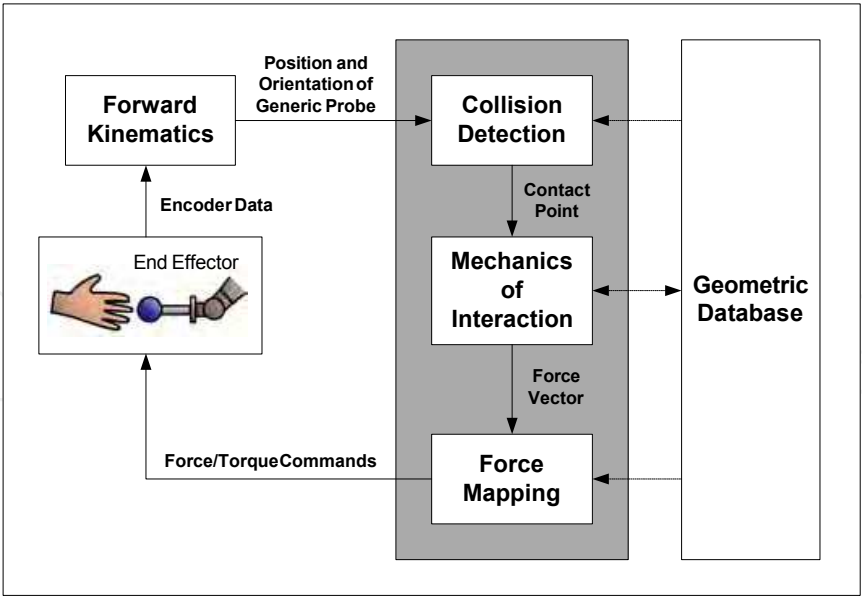
Fig. 8. Typical processes associated in haptic rendering with a force display.

In order to achieve the desired update rates and allow any future expansions, we intend to design and develop the system framework with concepts of concurrency, openness, scalability, and transparency. Potentially we can distribute computational loads, reconfigure, upgrade, and extend for different software and hardware components in order to achieve better system performance. Figure 8 illustrates the main components or processing tasks in a typical simulation hierarchy. The forward kinematics, inverse kinematics, and force mapping (Jacobian) equations were derived in the previous section. The above developments along with collision detection and simulation of materials, such as a virtual wall, are considered as computationally intensive components of any haptic interaction.

3.2 Hardware subsystems design
Our proposed local interconnection framework consists of three major components: the host control computer, our custom-designed data acquisition system (DAS) and the SPBS haptic device. The system block diagram of the experimental hardware setup is shown in Figure 9.
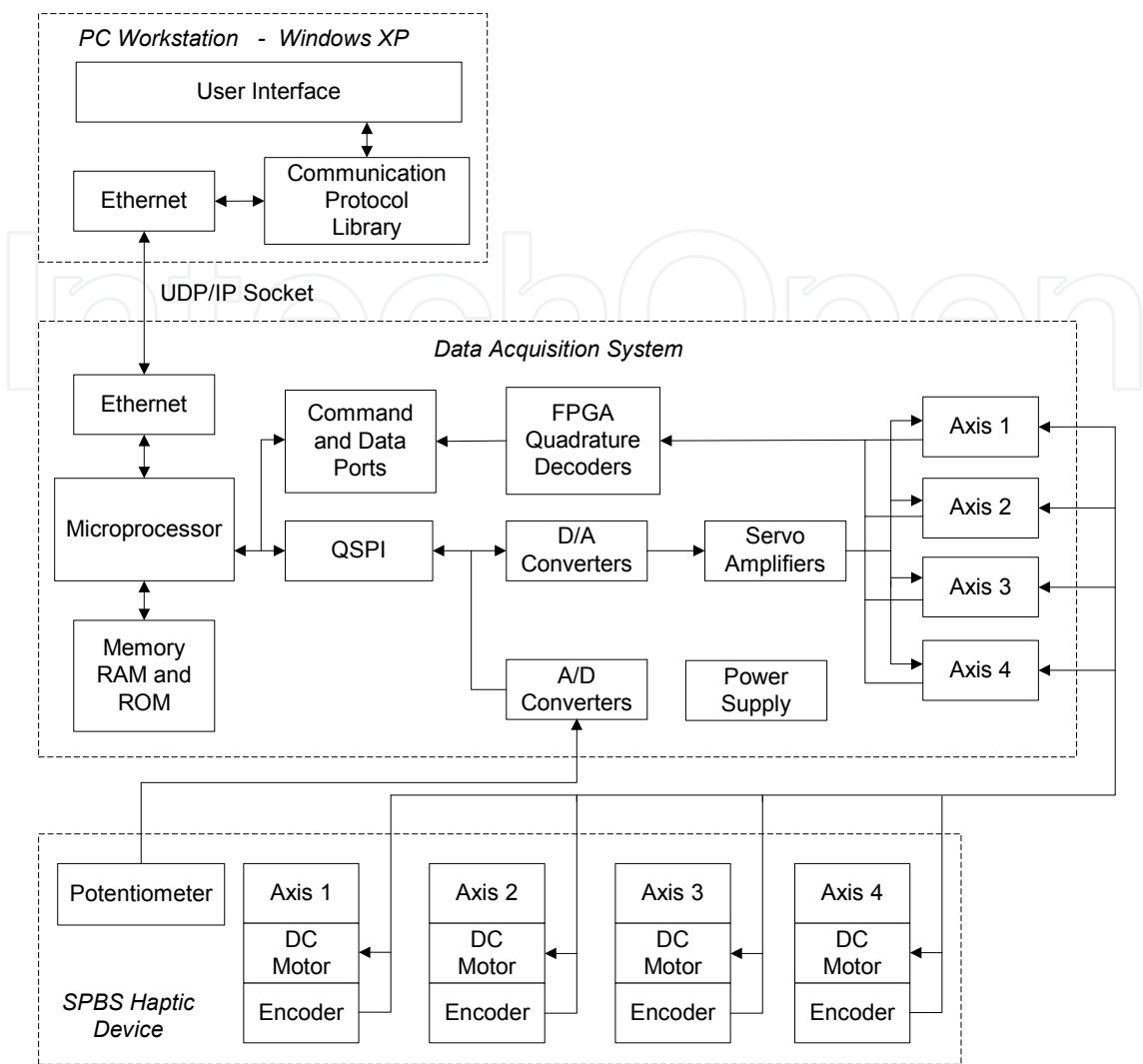
Fig. 9. System block diagram.

The host control computer is the driver for communicating with the DAS. The user perceives force feedback while manipulating the device. The four axes corresponding to the four DOFs are coupled with four DC brush motors. The user's movements with SPBS are acquired by DAS and sent to the host. The host PC provides visual displays and haptic rendering based on local models and simulation laws in the environment. The feedback values are sent back to DAS for real-time control of the motors.

A UDP/IP socket is selected as the primary data communication interface between the host PC and the DAS. The UDP/IP communication is connection-less and is known for having less overhead compared to the TCP/IP communication. The data acquisition system consists of a microprocessor, quadrature decoders, analog-to-digital converter (ADC), digital-to-analog converters (DAC), and servo amplifiers. The DACs provide the feedback voltages through the servo amplifiers to the four motors. We have used the NetBurner MOD5272 development board as the microprocessor unit for controlling data flow. The task of quadrature decoding is distributed and embedded on a Xilinx Spartan-IIE FPGA XC2S200E

evaluation kit. The quadrature decoders and ADC serve the purpose of data sampling. A Maxim MAX1203 IC is used for analog-to-digital conversion in this research. The ADC IC is capable of converting an analog voltage input ranged from 0V to 5V into a digital value with 12 bits of resolution. The maximum sampling frequency of the ADC is 2 MHz. By design, the ADC is used with a potentiometer for monitoring the opening and closing of the gripper. In this study, the ADC IC is also configured to collect experimental results (using current monitor output signals).

A quadrature decoder module is implemented to keep track of the motor positions. We used four Maxon DC brush motors for the four axes. Each Maxon motor uses a HEDL 5540 quadrature encoder capable of generating 500 counts per turn. Each encoder line driver requires a line receiver. Hence, two commercial MC3486 ICs are selected for integration with the motors and encoders used in this study. The Spartan-IIE FPGA on the evaluation kit uses a 50 MHz oscillator to drive the system clock. Power for the board is provided by an external +5 VDC regulated supply. The evaluation board provides jumper-selectable reference, output and termination voltages on a bank of the FPGA to facilitate the evaluation of various I/O standards. The LVTTL I/O standard is selected for integration with the ICs and the NetBurner development board used in this research. In addition, the evaluation kit provides 87 general-purpose I/O via two 50-pin connectors. This is sufficient for the current requirements and future exploration. The channels A and B of each quadrature encoder are used. Therefore, four encoders result in eight input bits. A reset signal for the registers is implemented via a push button on the board for clearing all the motor counts. We use a 16-bit data bus and a 2-bit control bus as the interface between the FPGA and the microprocessor module.

The DAS uses the NetBurner development board based on Motorola ColdFire5272 microprocessor. The network capability (TCP/IP and UDP/IP) and general purpose I/O ports are particularly useful for system evaluation in this study. The 10/100 Ethernet provides a standard network interface for connection to the host computer or a hub. The data transmission between the Netburner board and the FPGA is achieved through a 2-bit command port and a 16-bit data port as stated. In addition, the Queued Serial Peripheral Interface (QSPI) hardware on board is able to achieve a data rate up to 33Mbps. The DAS uses the QSPI and communicates with the ADC and DAC ICs by enabling different chip select bits. The UART provides a RS-232 serial interface that can be used for monitoring debug output on the host PC for this study.

The DAC converts the digital data to an analog voltage that feeds into the servo-amplifier. The servo amplifiers are configured to use in current mode, hence desired torque output can be controlled via two reference voltages. The DAC IC used in this project is a Maxim MAX525, which has a serial interface for communicating with the NetBurner board. There are four DAC channels on one IC and each voltage output ranges from 0V to 5V with 12 bits of resolution. The servo-amplifier receives a differential analog input pair. The sign of the differential voltage determines the motor turning direction. The amplitude of the differential voltage determines the motor torque. In our application, we used DC motors with a stall torque rating of 872 mNm. In stall mode, the stall current (and stall torque) is proportional to the applied voltage. Applying twice the voltage results in twice the stall current because

when the motor is not rotating (stalled) the armature appears in the circuit as a resistor. Therefore, we can use the rated terminal resistance (0.334 ohms) and the torque constant (19.4 mNm/A) from the motor specification in order to estimate the conversion between our desired torque and the motor current and terminal voltage. For example, if the desired torque is 23.2 mNm, the motor should draw approximately 23.2 / 19.4 = 1.2 A from the power supply. The terminal voltage across the motor leads should be about 1.2 A * 0.334 ohms = 0.4V. In this study, we used a power supply with a 12A nominal output current. Servo amplifiers are the 25A8 models from Advanced Motion Controls. We configured the servo amplifiers to operate in current mode with maximum continuous current rating of 12.5A. Therefore, in our haptic rendering process, we consider a safe operating range for the motors by limiting the torque output to be below 100 mNm.

3.3 Communication Protocols

A simple data packet is used for performance benchmarking of the experimental setup. Figure 10 and Figure 11 show a sample data packet and a control data packet.
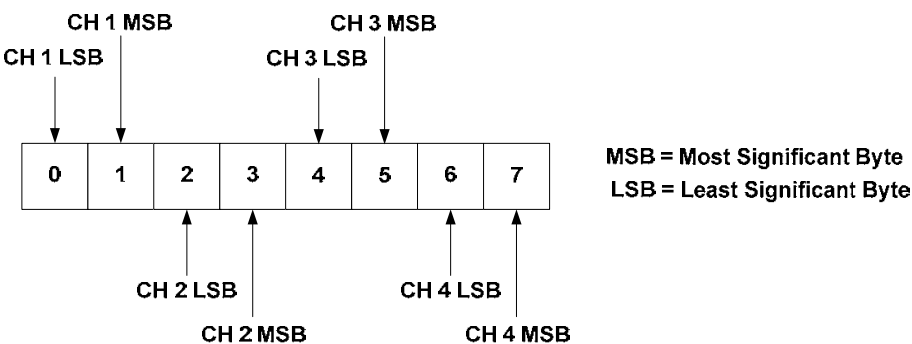
Fig. 10. Sample data packet (8 bytes).
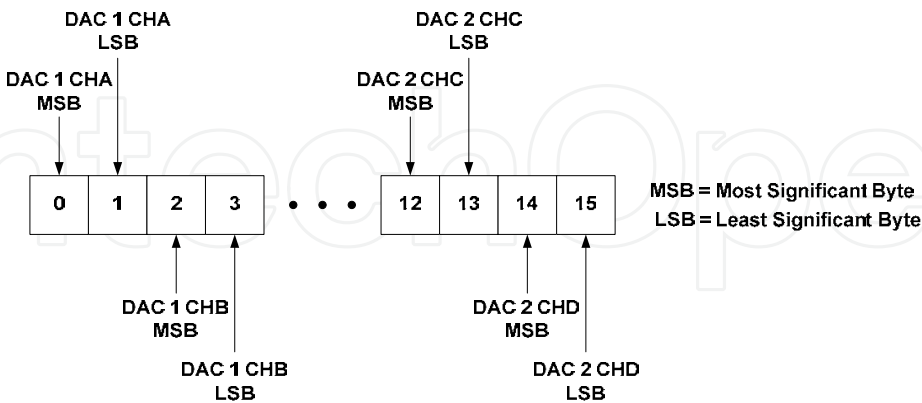
Fig. 11. Control data packet (16 bytes).

3.4 Host Computer

Figure 12 illustrates the high-level design hierarchy of the development of a software application on the host computer. The software applications were targeted and tested on the Windows XP and the Linux operating systems. On Windows XP, Microsoft Visual C++ 6.0 was used as the compiler for building the software application. The WinSock API was used to support UDP/IP and TCP/IP communication with the DAS. On Linux, GNU g++ was used as the compiler for building the software application. The POSIX socket was used similarly on Linux in order to support the communication protocol. The GTK+ software library is used to provide the same look and feel of the GUI on Windows and on Linux.
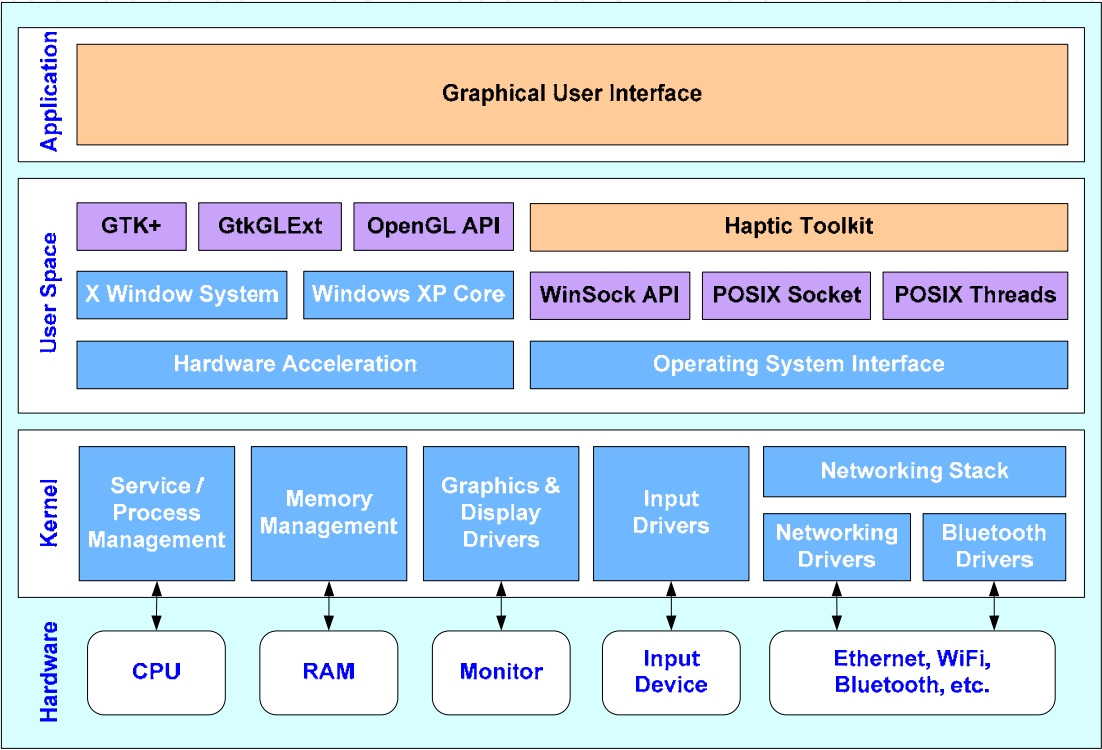
Fig. 12. High-level Design Hierarchy.

4. Experimental Results

Figure 13 shows the sequence diagram of the virtual environment (VE) and the allocation and distribution of the tasks.
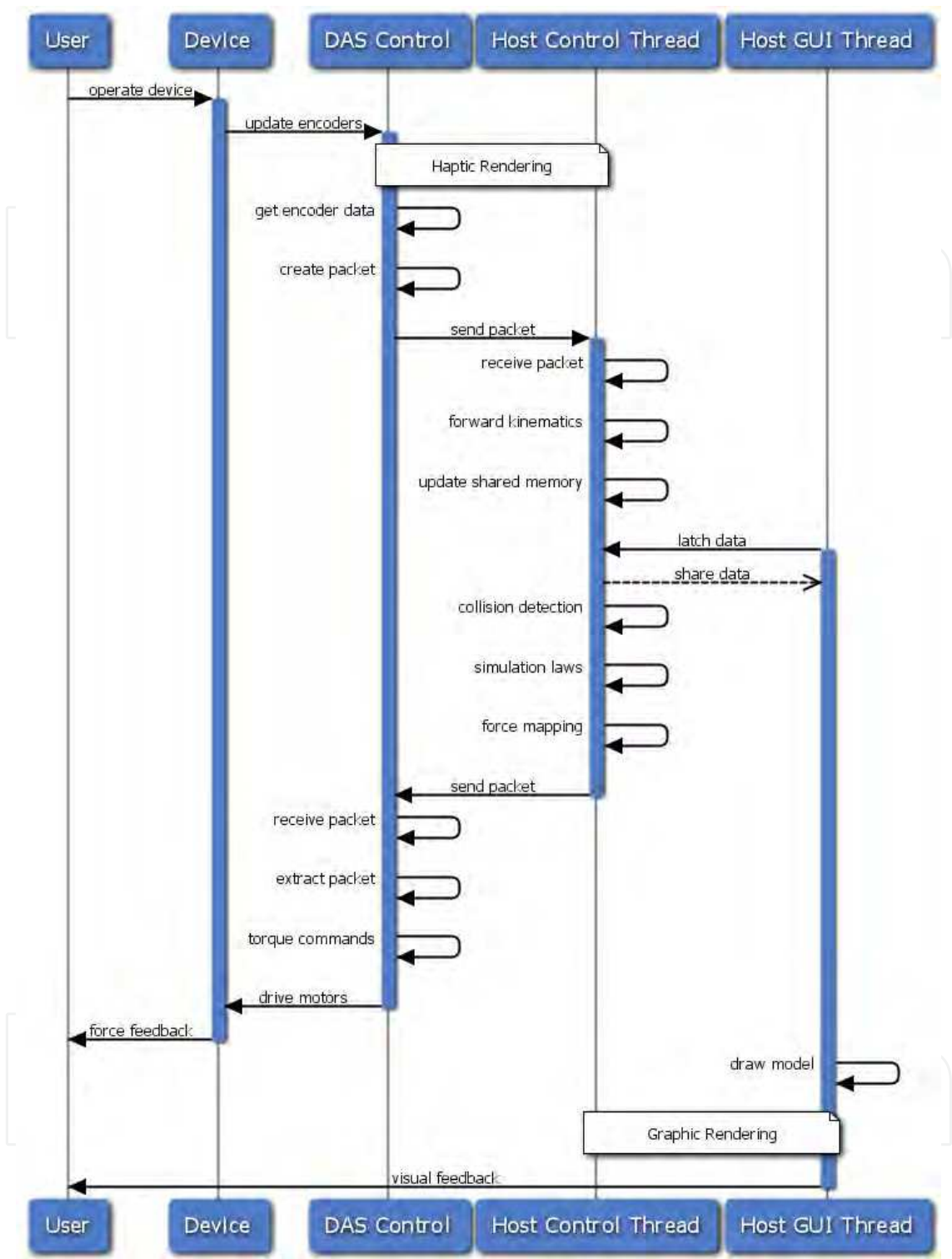
Fig. 13. Sequence diagram of the virtual environment.

As shown, the user and the device represent the human operator manipulating the physical SPBS device while observing the VE and GUI on the PC monitor. The DAS control task represents the control program executing on the NetBurner microprocessor. The multi-threaded software application executing on the host computer consists of two threads. The

GUI thread or task is responsible for graphics rendering such as the user interface, the graphical device model and virtual objects to the operator. The control thread on the host computer runs in the background and is responsible for network communication, kinematics, collision detection, mechanics-based simulation, and force mapping of the control process.

4.1 Performance Benchmarking

Haptic rendering is the process of computing and applying force feedback to the user in response to his/her interaction with the virtual environment. How haptic rendering is implemented should depend on the application requirements, since there is no unique or best solution. Two common approaches are impulsive haptic rendering and continuous haptic rendering (Buttolo & Hannaford, 1997). Impulsive haptic rendering models impulsive collisions such as kicking a ball and hammering a nail. Continuous haptic rendering models extended collisions such as pushing against a wall or lifting an object.

A common question is how fast should we sample, and how will delay affect performances. Previous studies suggest a threshold based on human perception, resulting in a requirement for force reflection bandwidth of at least 30-50 Hz for integrated graphics and impulsive forces [11]. To realistically simulate collisions with rigid objects (i..e. high stiffness such as 1000 Nm^{-1}), desired sampling rates are at least 200 Hz. Many state of the art haptic systems use 1000 Hz sampling rates (Mishra & Srikanth, 2000). In the following, we prepare the experimental setup and the operator positions the tool continuously colliding with a virtual wall in the virtual environment. The GUI thread or the graphics rendering loop (see Fig. 13) is tuned to execute at about 50 Hz. The average sampling rate of the haptic rendering loop can be measured on the host computer or on the NetBurner. We used the continuous haptic rendering model in order to evaluate the performance of the experimental setup.

Table 2 below shows the throughput, sampling rate and tested payload. The sampling rate is defined as how many closed-loop control cycles the host computer can complete in one second. A complete closed loop control cycle requires the host computer to receive a new set of sampled data (motor positions), compute the haptic rendering, and update the control data (voltages). When data is sent over a network, each unit transmitted includes both header information and the actual data being sent. The header identifies the source and destination of the packet, while actual data is referred to as the payload. As indicated in section 3.3, the predefined communication protocol encapsulates the motor positions (the host receives 8 bytes) and control voltages (the host transmits 16 bytes) using a payload of 24 bytes. The performance benchmarking application keeps track of the number of samples/iteration and the actual data size measured in a time interval. The mean sampling rate was the average value over 10 trials. Each trial involved an operator using the device in the virtual environment for a one-minute interval and recorded the sampling rate by using customized firmware on NetBurner. Besides the default size of the data packets, the same experiment was repeated for increasing packet sizes. This allows an observation of the system performance if additional information (such as time stamps) propagated over the networked environment. Table 2 shows the test results with the application executing on Windows XP.

Payload (bytes)			TCP		UDP	
Total	Rx	Tx	Mean Sampling Rate (Hz)	Mean Throughput (Bytes/sec)	Mean Sampling Rate (Hz)	Mean Throughput (Bytes/sec)
24	8	16	768	18432	1105	26520
28	12	16	765	21420	1063	29764
32	16	16	765	24480	1020	32640

Table 2. Communication Performance Statistics (TCP and UDP) in Windows XP.

Comparing the performance of the TCP communication protocol to the UDP, the UDP achieves the desired 1 kHz update rates. The connection-less UDP socket with less overhead outperforms the connection-oriented TCP socket for the intended application of the system. Note that as the size of the data packets increases, the mean sampling rate decreases. Table 3 shows the results by repeating the same experiment on a Linux environment.

Payload (bytes)			UDP (Windows XP)		UDP (Linux)	
Total	Rx	Tx	Mean Sampling Rate (Hz)	Mean Throughput (Bytes/sec)	Mean Sampling Rate (Hz)	Mean Throughput (Bytes/sec)
24	8	16	1105	26520	1181	28344
26	10	16	1083	28158	1161	30186
28	12	16	1063	29764	1134	31752
30	14	16	1042	31260	1108	33240
32	16	16	1020	32640	1084	34688
36	18	18	999	35964	1063	38268
40	20	20	980	39200	1039	41560
44	22	22	959	42196	1019	44836
48	24	24	941	45168	998	47004

Table 3. Communication Performance Statistics (Windows XP and Linux).

4.2 Kinematics Verification

Other tests have been conducted using the hardware setup, the SPBS device, and a Pentium 4 PC running the software application on Windows XP. As shown in Figure 14, the measured length from the origin (centre of rotation) to the tip of the handle is 0.35m. By moving the tip of the handle to test point 1, the measured coordinate on the device in the physical workspace is (0.346, 0.050, 0.000) ± 0.001m.
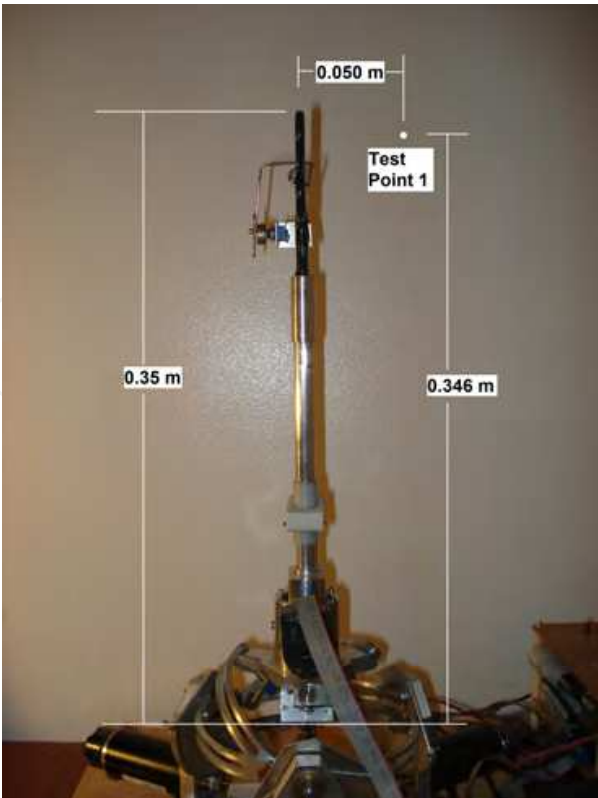
Fig. 14. Measured test point in physical workspace.

Comparing to the model rendered in the virtual environment, as the operator positioned the device to the test point, the calculated coordinate based on measured motor angles, forward kinematics, and a proper scaling of the model is (0.346, 0.050, 0.000).
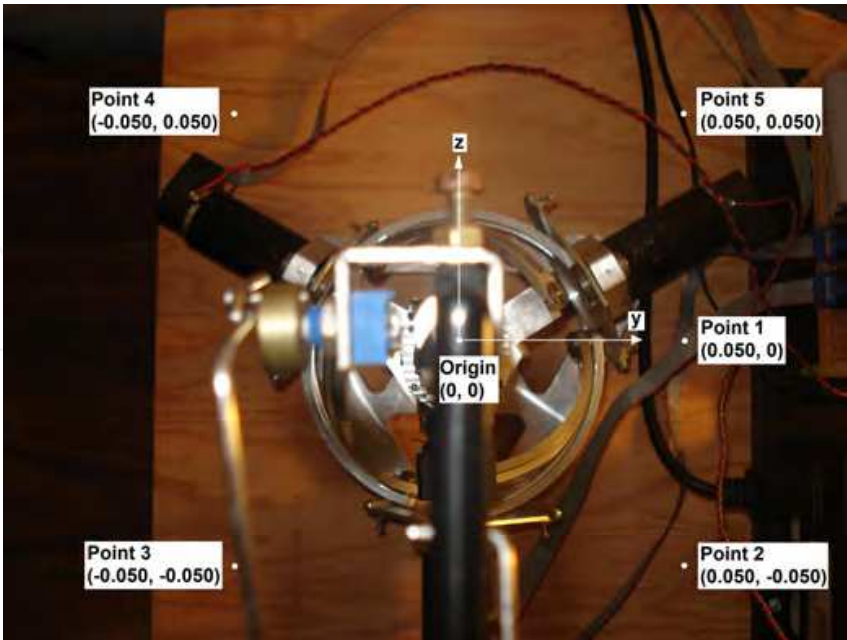

Fig. 15. Test points in physical workspace.

By adjusting the position of a camera along the positive x-axis, Figure 15 shows the top view of the device and the previous test point 1 projected onto this view plane at a distance approximately 0.35m parallel to the y-z plane. Table 4 shows the comparison between the measurements on the actual device and the calculated coordinates on the virtual model as the operator manipulated and positioned the tip of the handle to all the test points shown. Note that the experiments were conducted by the operator determining the position of the tip of the handle and estimating a home reference with all motor angles resetting to zero at the starting origin. The imperfect zero-home reference, estimated location of the handle tip, and camera displacements may introduce source of errors during the experiments.

Test Point	Measured y-z coordinates on physical device (metres)	Calculated y-z coordinates on virtual model (metres)
1	(0.050, 0.000)	0.050, 0.000
2	(0.050, -0.050)	0.050, -0.049
3	(-0.050, -0.050)	-0.049, -0.049
4	(-0.050, 0.050)	-0.049. 0.051
5	(0.050, 0.050)	0.050, 0.051

Table 4. Comparison between measured coordinates and calculated coordinates

4.3 Haptic Feedback

In addition to the observation and tracking of a virtual tool, an example haptic scene is prepared for the operator to experience haptic feedback in the environment. A virtual wall is predefined in the scene prior to the experiment. Figure 16 shows the virtual environment (left) and the home position of the device model (right). The sphere rendered in the left region indicates the haptic interface point in the scene.
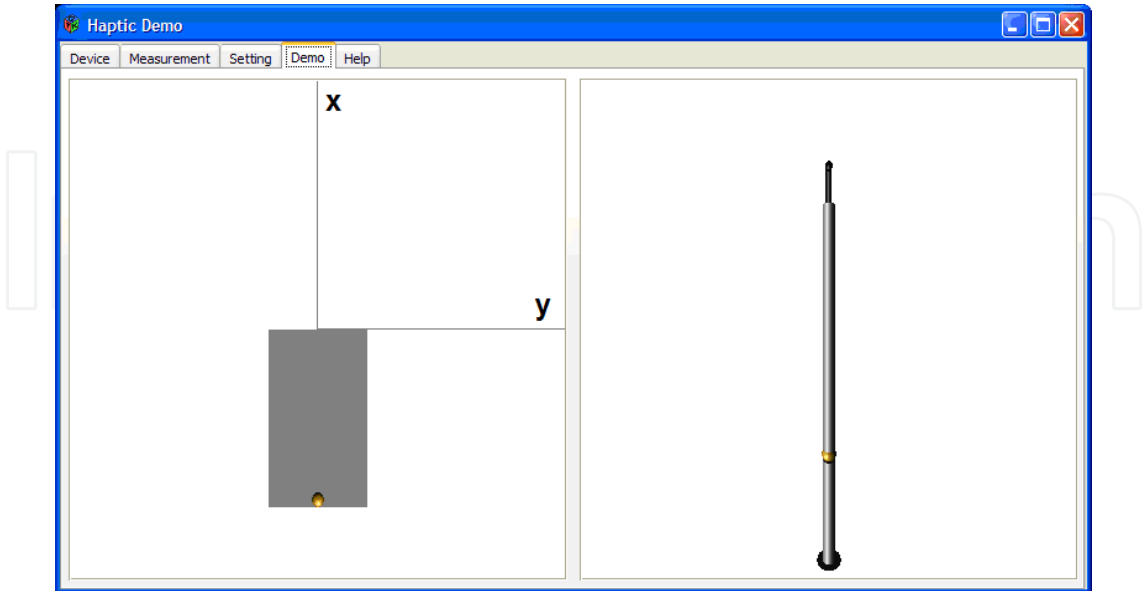
Fig. 16. Virtual environment for haptic exploration (virtual wall into page)

The position of the wall is located at 0.020m into the page (positive z-axis) relative to the origin. The wall (rectangle) is parallel to the x-y plane. The home position of the tool (or straight up) is along the positive x-axis. Figure 16 shows the scene with the camera behind (on negative z-axis) and looking at the origin. The operator performed the experiment by moving the sphere towards the wall along the positive z-axis and colliding the sphere with the virtual wall. Figure 17 shows the position of the sphere as the operator manipulated the tool and moved the sphere accordingly. Figure 18 shows the calculated Cartesian force Fz (Fx = Fy = 0) at the haptic interface point when the sphere collided with the virtual wall.
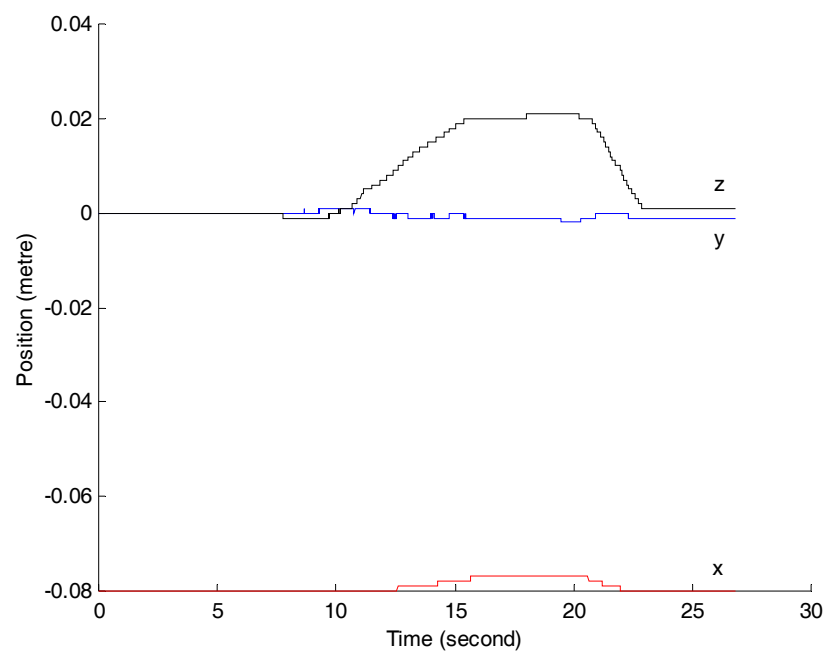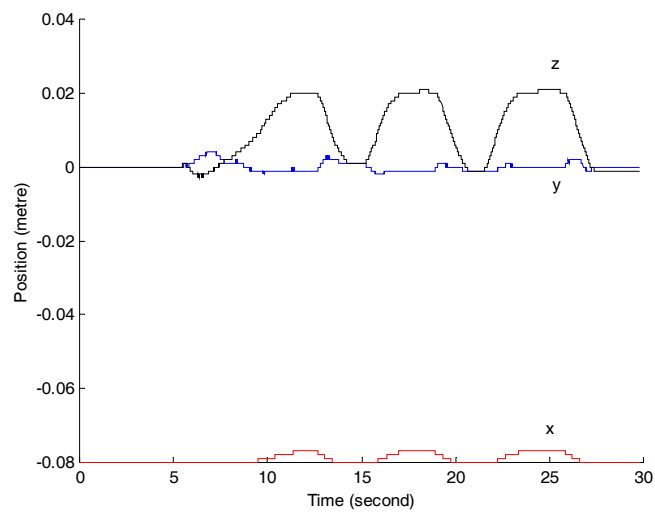
Fig. 17. Position of the sphere with respect to the reference coordinate system.
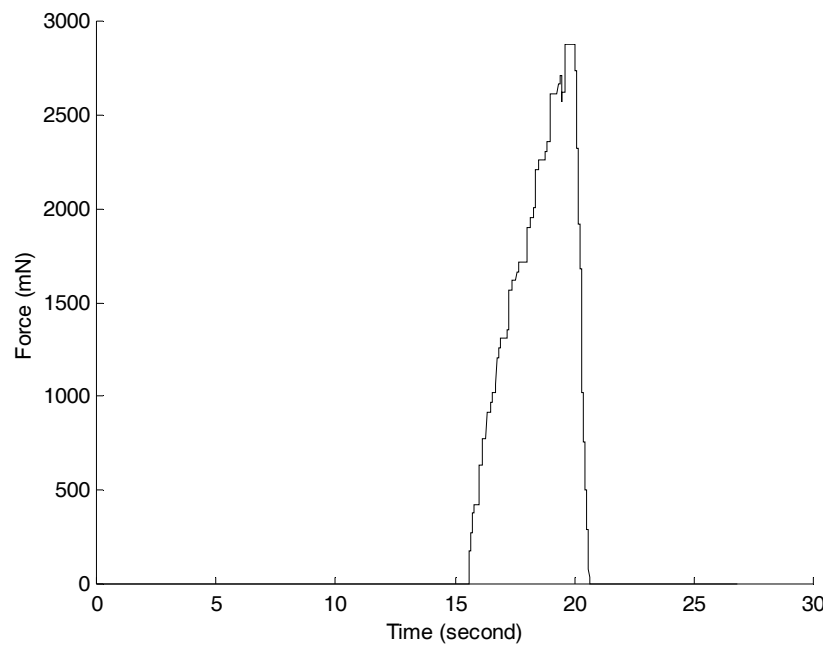
Fig. 18. Cartesian force at the haptic interface point along the z-axis.
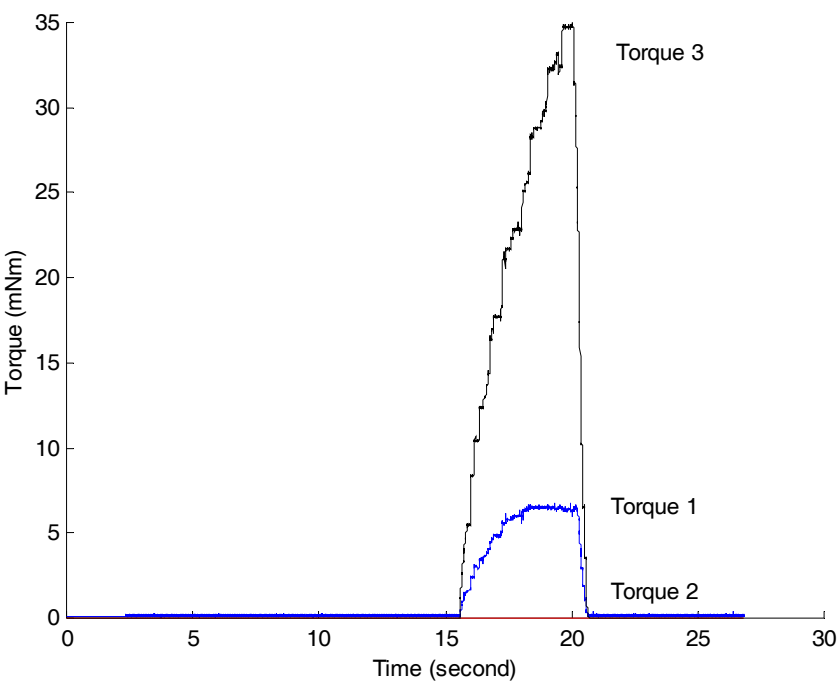
Fig. 19. Measured torque versus time plot.

Figure 19 shows the measured torque versus time plot. The torque was measured by the application of the current monitor output (signals available from the servo amplifiers) and the integration of a low-pass filter circuit and the ADC IC on the DAS. Three axes of the motors were monitored for the torque output during the experiment. The results show the torque experienced by the operator holding the tool and repeatedly colliding with a virtual wall. As seen in the above plots, the period during which the sphere position is at the threshold, i.e. when the spring model is in effect, the force started to increase as the operator attempted to push the sphere further onto the wall. Figure 19 shows the decomposition of the torque into three motor torques felted by the operator. In this example, actuator 1 and actuator 3 were operating in order to generate the reaction force. Using the same virtual scene, the following plots show the experimental results as the operator moved the sphere back and forth from the origin to the wall experiencing a greater reaction force while attempting to penetrate the sphere further into the wall.

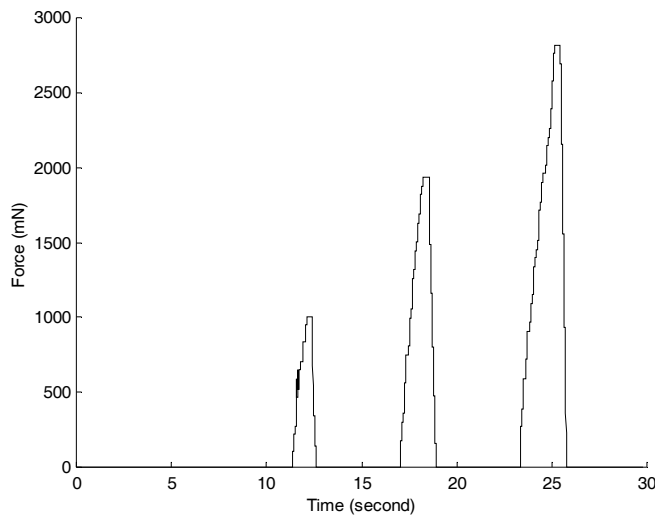Fig. 20. Position of the sphere with respect to the reference coordinate system.

Fig. 21. Cartesian force at the haptic interface point along the z-axis.
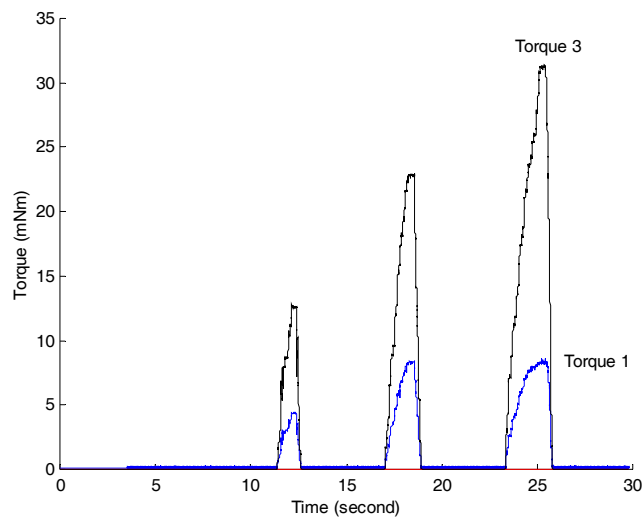

Fig. 22. Measured torque versus time plot.

4. Summary

This chapter presented the design, modelling and hardware integration of a haptic device. The design of the haptic device is based on the notion of the hybrid spherical mechanism which consists of both a passive and active spherical joints. The passive joint is responsible for supporting the static load and user interaction forces whereas the active joint is responsible for creating the haptic feedback to the user. Closed-form solution for kinematic analysis and force mapping of the device is presented. A novel distributed computational platform is also proposed. The platform exploits the notion of scalability and modularity in the design. Performance of the closed loop system is presented in the context of interacting with a rigid environment and achieving a high sampling rate using either the UDP or the TCP communication protocols.

5. References

Angeles, J. & Gosselin, C. (1990). Singularity analysis of closed-loop kinematic chains, *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281-290, 1990

Birglen, L.; Gosselin, C.; Pouliot, N.; Monsarrat, B. & Laliberté, T. (2002). SHaDe, a New 3-DOF Haptic Device, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 166-175, 2002

Boudreau, R., Darenfed, S. & Gosselin, C. (1998). On the Computation of the Direct Kinematics of Parallel Manipulators Using Polynomial Networks, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 28, no. 2, pp. 213-220, 1998

Buttolo, P., Oboe, R. & Hannaford, B. (1997). Architectures for Shared Haptic Virtual Environments, *Computers & Graphics*, vol. 21, no. 4, pp. 421-429, 1997

Craver, W. (1989). *Master Thesis: Structural Analysis and Design of a Three-Degree-Of-Freedom Robotic Shoulder Module,* The University of Texas at Austin, 1989

Gosselin, C. & Hamel, J. (1994). The Agile-Eye: a High-Performance Three-Degree-Of-Freedom Camera Orienting Device, *Proc. IEEE Int. Conf. Robotics and Automations*, vol. 1, pp. 781-788, 1994

Li, T. & Payandeh, S. (2002). Design of Spherical Parallel Mechanisms for Application to Laparoscopic Surgery, *Robotica*, pp. 133-138, 2002

Ma, A. & Payandeh, S. (2008). Analysis and Experimentation of a 4-DOF Haptic Device, *16th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 351-356, March 2008

Mishra, R. & Srikanth, S. (2000). GENIE – An Haptic Interface for Simulation of Laparoscopic Surgery, *Intelligent Robots and Systems*, vol. 1, pp. 714-719, 2000

Advances in Haptics

Edited by Mehrdad Hosseini Zadeh

Haptic interfaces are divided into two main categories: force feedback and tactile. Force feedback interfaces are used to explore and modify remote/virtual objects in three physical dimensions in applications including computer-aided design, computer-assisted surgery, and computer-aided assembly. Tactile interfaces deal with surface properties such as roughness, smoothness, and temperature. Haptic research is intrinsically multi-disciplinary, incorporating computer science/engineering, control, robotics, psychophysics, and human motor control. By extending the scope of research in haptics, advances can be achieved in existing applications such as computer-aided design (CAD), tele-surgery, rehabilitation, scientific visualization, robot-assisted surgery, authentication, and graphical user interfaces (GUI), to name a few. Advances in Haptics presents a number of recent contributions to the field of haptics. Authors from around the world present the results of their research on various issues in the field of haptics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds