

Simulation of Haptics Force Law using SimMechanics and Simulink

Majid H Koul, Subir K Saha
Department of Mechanical Engineering
Indian Institute of Technology Delhi
New Delhi-16, India
majidkoul@gmail.com, saha@mech.iitd.ac.in

M Manivannan
Department of Applied Mechanics
Indian Institute of Technology Madras
Chennai-36, India
mani@iitm.ac.in

Abstract—Haptic devices are special robotic devices that convey interaction forces from the virtual reality environment to the human operator. Since the human operator holds the device for interaction, ensuring the stability of these systems is very important. In this paper we present our work in analyzing and identifying the stability bounds of these haptics systems, namely the Z-width, specifically for multi-DOF (Degrees of Freedom) systems. Multi-DOF systems are usually modelled with complex dynamic equations. We utilized the features of SimMechanics and Simulink toolboxes of MATLAB for generating the dynamic model of these systems. In addition to the basic stick modelling feature in SimMechanics, CAD model of a real 2-DOF haptic device was imported to SimMechanics environment. A virtual wall interaction with the 2-DOF haptic device was modeled using the basic modules available in Simulink. The complete haptics environment including components like quantization of encoders, sampling rate of the controller, device friction, velocity estimation and filtering was realized for a realistic haptics force law simulation. The proposed implementation enabled us to study the effect of various components like virtual damping, encoder resolution, sampling rate etc. on the performance and stability of a haptics control loop.

Keywords – *Simulation; Haptics; SimMechanics*

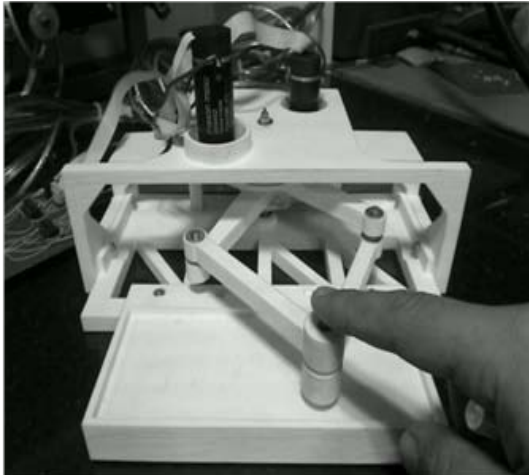
I. INTRODUCTION

Haptics refers to the technology that enables interaction with the virtual environment through the sense of touch. The sense of touch or force-feedback from the virtual environment is conveyed through a special robotic device referred to as a haptic device. Applications range from training medical residents through virtual surgery, carrying psychophysics experiments, developing haptic enabled CAD environments, games, art and education etc. A typical haptics loop involves conveying the forces of interaction from the virtual environment to the human operator. Since human operator forms a part of haptic control loop, ensuring the stability of these systems is very important. Instability of haptic systems refers to the interactions with a virtual environment that result into unintended and perceived vibrations of constant magnitude. These vibrations or oscillations of constant magnitude are also referred to as limit cycles in

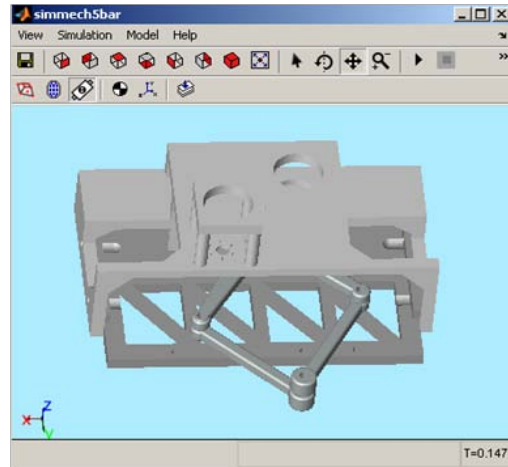
non-linear control literature. Presence of limit cycles destroys the feel of a true virtual environment that is necessary for a realistic haptic simulation. Several factors contribute to the onset of instability in these haptic systems which include sampling rate of the controller, computational delay, physical damping of the device, sensor resolution, velocity estimation, coulomb friction, etc. [1].

Typical virtual environments are modeled as lumped spring damper systems in mechanical parallel. For simulating a wall, the forces of virtual wall interaction are estimated as a function of virtual wall stiffness and the virtual wall damping. A region of values of these virtual wall parameters that can be implemented stably by a haptic device is typically referred to as Z-width in the conventional haptics literature. The Z-width of a particular haptic device/controller is typically restricted by the factors responsible for instability, as listed above [1].

SimMechanics [2] has been previously used by several authors for modeling, simulation and controller analysis of robots. In [3], the authors use SimMechanics and Simulink for modeling of spatial 6-DOF parallel robot (SPR). The authors utilized the dynamic equations of motion of the SPR to build the model using basic components available in Simulink. A PID controller for the same (i.e., SPR) was further realized in Simulink. They compared the results with a similar model developed in SimMechanics to confirm the model developed in Simulink. In [4], SimMechanics was used for kinematic simulation and control of a novel 2-DOF parallel mechanism. A controller was developed in Simulink and SimMechanics to make the end-effector track a reference trajectory. Their modeling and controller tests confirm the reliability and accuracy of using SimMechanics for designing nonlinear and mechatronic systems. A MATLAB-Simulink dynamic model of the 3-DOF anthropomorphic robot manipulator along-with a robust command following control algorithm was developed in [5]. It was established that such type of control with an unambiguous correlation between the desired value and the driving torque enables an exact interpretation of the experimental results with respect to the verification of the synthesized dynamic model [5]. Hence MATLAB-Simulink, guarantees as being an essential tool for modeling, simulation and control validation of



(a) Real prototype in action



(b) CAD model imported to SimMechanics

Fig. 1. Two-DOF haptic device

robotic systems.

Several authors have studied the stability of haptic systems from the analytical and experimental point of view. Much of the work has been carried out from the network theory that establishes conservative bounds on the stable parameters of the virtual environment [1]. However, authors like Gill et al. [6] studied stability from a gross perspective (without accounting for practical hardware limitations like motor saturation, coulomb friction etc.) by performing controller analysis in the digital domain. This gross stability was utilized to refine the stability criteria established by Colgate and Brown [1]. It was established that virtual damping contributes positively towards the stability of haptic systems. Most of the work established stability bounds experimentally for an one-DOF haptic device subjected to a virtual wall interaction. Theoretical analysis of the same has been performed in digital domain requiring the knowledge of z-transform. A Simulink model of the one-DOF controller can also be developed to study the gross stability bounds of a haptic interface. However, for a multi-DOF haptic interface, stability analysis is not trivial. Since the multi-DOF haptic device is a non-linear system needing dynamic equations of motion for controller analysis, the MATLAB-Simulink environment could suitably be explored.

In this paper, we demonstrate the modeling and control of a 2-DOF haptic device in SimMechanics and Simulink. In particular, modeling aspect of the device is discussed first followed by the simulation of haptics force law for the multi-DOF system. We provide a detailed overview of the procedure that was carried out to formulate the haptic force law in Simulink. In particular, we mimic the exact virtual wall interaction that is generally achieved in experiments. Stability aspect of the controller is also discussed.

The rest of the paper is organized as follows: Section 2

discusses the mechanical modeling of a haptic device in SimMechanics; Section 3 details the basic components of a multi-DOF haptics controller; while Section 4 discusses the development of haptics controller in Simulink. Section 5 discusses the steps in simulation while Section 6 details the results obtained from simulation. Conclusions are summarized in Section 7.

II. MECHANICAL MODELING IN SIMMECHANICS

SimMechanics extends the capabilities of Simulink in modeling and simulating electro-mechanical systems. The modeling feature comes with a group of tools to specify bodies of a mechanical system along with their inertial and geometric properties, their configuration/kinematics, the coordinate system, the type of joints, initial conditions and the sensing and actuation tools to measure and initiate the motion of the system. SimMechanics automatically generates the equation of motion for the complete system at hand. Apart from the facility of modeling the mechanical system (e.g., a haptic device) in SimMechanics, an alternative method of using the exact device shape/ geometry and inertial properties of the actual device is also possible.

In this work, a CAD model of the actual haptic device developed in our laboratory was already available at hand. The actual device is shown in Fig. 1(a). These CAD files were imported to SimMechanics environment using XML file format [7], as shown in Fig. 1(b). To import the CAD model from Autodesk Inventor into Simulink/SimMechanics environment, a plug-in namely *SimMechanics Link* was installed from the Mathworks website [7]. After installation, this add-in appears as a SimMechanics link in Autodesk Inventor. Using this add-in an XML file of the CAD assembly was generated. The XML file was then imported in

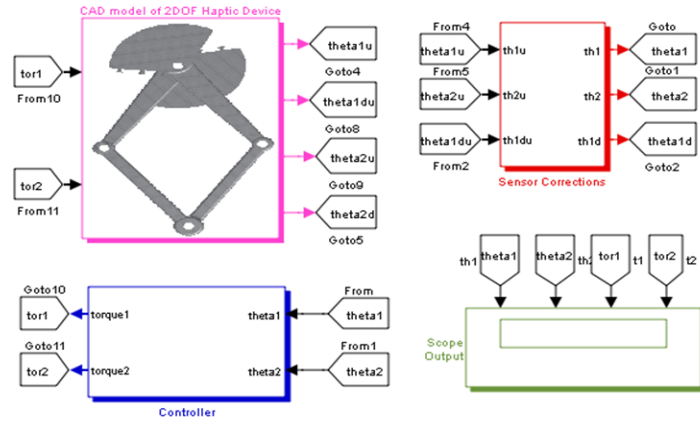


Fig. 2. Block diagram of the complete system

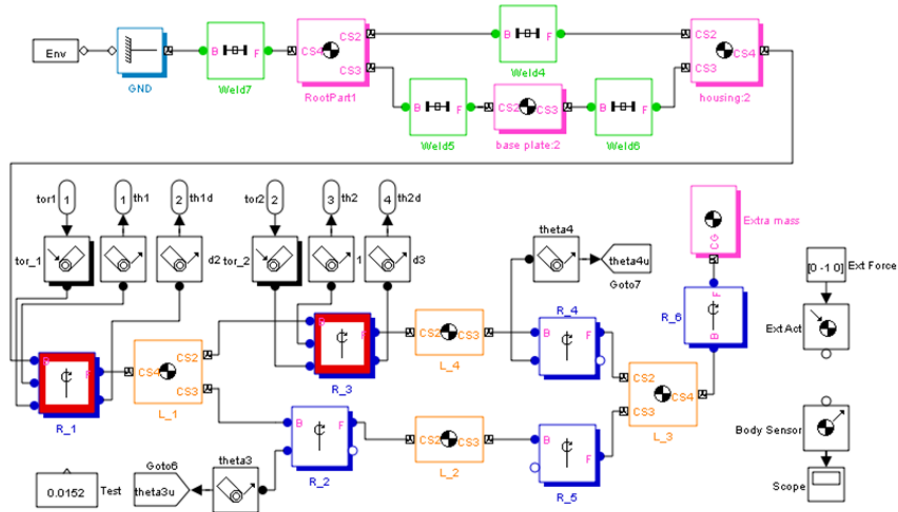


Fig. 3. Internal details of the imported CAD model

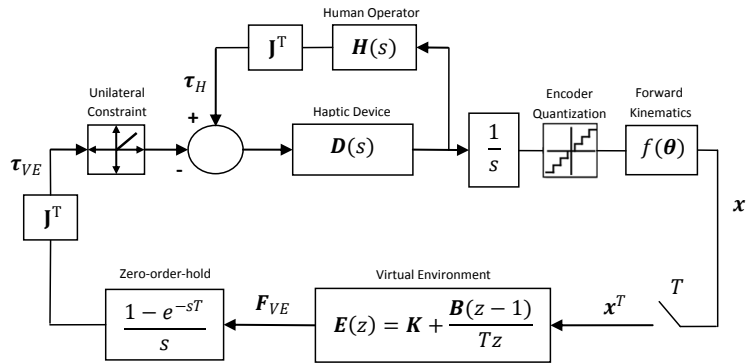


Fig. 4. The 2-DOF haptic control loop

SimMechanics environment using the `smimport('abc.xml')` command at the Matlab command prompt, where `abc` was the name of the XML file. Using this procedure, SimMechanics model of the CAD assembly was generated. The inertial and kinematic parameters of the device were automatically imported along with the graphics file. Figure 2 depicts the basic blocks of the complete system realized in SimMechanics and Simulink. The modeling consisted of two major blocks: CAD/Mechanical model of the device and the associated haptics controller. In addition, a block on sensor corrections was required to match the coordinate system/convention of the imported CAD model with the SimMechanics environment. Figure 3 depicts the internal details of the CAD model imported from Autodesk Inventor. SimMechanics automatically creates a mechanical model of the imported file utilizing the basic building blocks available in its library. These building blocks are mechanical body block, joint modules, and ground and environment modules. Environment module contains details about the effect of gravity and other physical details like the type of analysis needed, tolerance in the geometry, etc. To utilize the imported model properly, addition of sensor and actuator modules are required. A joint sensor module measures various joint variables like position, velocity, etc. in addition to the torque at desired joints. The actuator module provides the facility to impart the necessary torque or force at the required joints or at different locations of the device, respectively.

III. HAPTICS CONTROLLER

A typical controller used to simulate a haptic force law for a multi-DOF haptic device is shown in Fig. 4. It consists of a block represented by $D(s)$ that represents the device dynamics. An output of such block with torque as the input is the device velocity; hence $D(s)$ acts as admittance block also. Since human forms the part of a control loop, $H(s)$ denotes the human impedance encountered by the device. The position information is usually sensed by optical encoders which send a quantized signal based on the sensor resolution. This quantized signal is then passed on to the forward kinematics block that converts the joint angles to the end-effector position. A virtual environment block $E(z)$ computes the forces of interaction based on the penetration depth and speed of the end-effector. These forces of interaction are converted to the necessary torques at the active joints by pre-multiplying them with the transpose of Jacobian. A zero-order hold effect is produced since the positional signal in real system is sampled at T ms. A unilateral constraint block establishes the basic part of haptics force law, i.e., zero force outside the virtual wall and non-zero inside the virtual wall. For a multi-DOF virtual wall interaction, the stiffness and damping coefficients are represented in a diagonal-matrix form. Since we are concerned with a 2-DOF planar device here, our virtual wall coefficients would be a 2×2 diagonal matrix. So the forces of interaction/reaction force would also have two components. This reaction force vector is

multiplied by the transpose of Jacobian which is a 2×2 matrix. It results into a 2-dimensional vector that represents the necessary torques required at the active joints. It must be noted that the direction of the torques should be such that the device produces reaction force as desired with a virtual wall interaction. The next section details the modeling steps carried out in Simulink and SimMechanics.

IV. HAPTICS CONTROLLER IN SIMULINK

Simulink provides the facility of modeling almost every component of a haptic control system shown in Fig. 4. Figure 5 depicts the components of the controller modeled in Simulink. A joint sensor module connected to the revolute joint module of the CAD model which provides the necessary joint information like position, velocity and acceleration. We utilized the positional information of the active joints in radians and passed it through the quantization block that enabled us to mimic the effect of encoder resolution. For estimating the velocity, this position information was passed through the zero-order-hold block that enabled us to estimate the velocity at a particular sampling rate. The position information after passing through the zero-order-hold block was differentiated using the discrete differentiator module which was then filtered using the filter subsystem designed using `fdatool` in MATLAB, as shown in Fig. 6. This filtered joint-rate output was fed to the matrix multiply block where it was multiplied by the Jacobian resulting into the velocity of the device's end-effector. It must be noted that the CAD model only takes care of the inertial and kinematic parameters of the device and not the friction present in the actuator or at the joints. So we experimentally estimated the coefficient of viscous and coulomb friction in the actuator. These friction components were amplified for their effect at the device axis since the motor and device joint was non-collocated for torque enhancement. The effect of friction on the haptic device was added as a frictional torque subsystem shown in Fig. 7. In this subsystem, the velocity estimated by the joint sensors available in SimMechanics was used, since this effect is naturally added in the real system rather in a discrete way like the viscous damping of the virtual environment. Coulomb friction would act as a constant torque based on the direction of device velocity. The same was incorporated after the signum function shown in Fig. 7. A collision detection subsystem was added to realize the function of the unilateral constraint block. The procedure for collision detection would incorporate definition/location of a virtual wall in the y -direction. We utilized `if` and `elseif` subsystems available in Simulink to define the condition in which the controller would be on and off, inside and outside the virtual wall, respectively. The reaction force was estimated based on the typical haptics force law which is a function of the end-effector's position and velocity. This reaction force was converted to appropriate torques at the active joints using force-torque mapping block in Fig. 5. This force/torque mapping block performs the pre-multiplication of Jacobian matrix with the reaction force vector. The Jacobian was evaluated by writing the Jacobian matrix in the

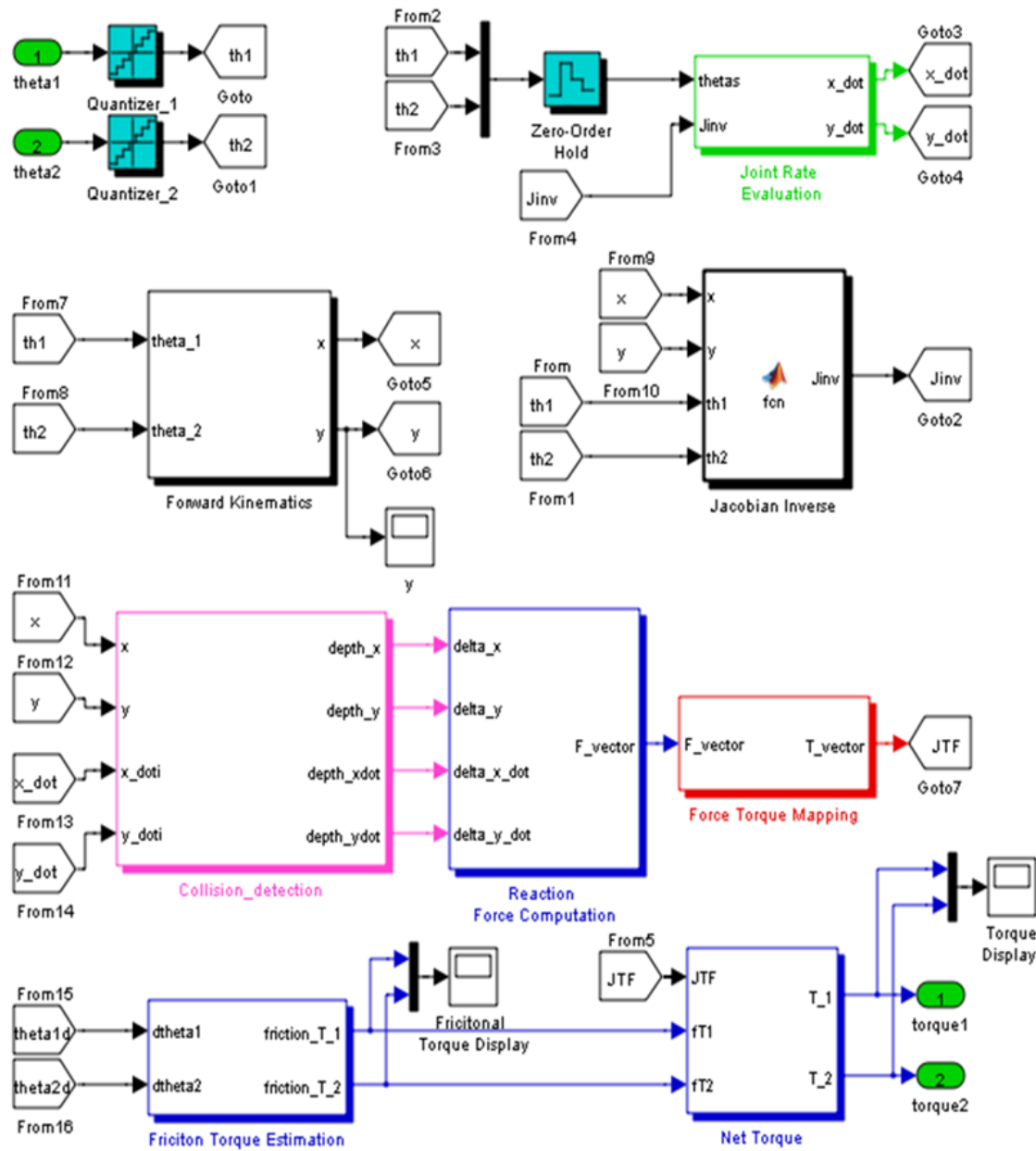


Fig. 5. Detailed controller module in Simulink

MATLAB's embedded function module that would take joint positions and end-effector positions as input. Appropriate directions of the friction torque were chosen and the reaction torque was limited to the motor saturation torque provided in the datasheets of the real motors used in the haptic device at hand.

V. SIMULATION

After the device was successfully build or imported from the CAD environment, the controller (designed in Sect. 4) shown in Fig. 5 was connected to it by incorporating a controller subsystem in the main block diagram, as shown in Fig. 2. Initially the device was placed at a particular position (non-singular region) in the vertical direction, as shown in Fig. 8. A linear visco-elastic model of a virtual wall was defined in the controller block that represented a

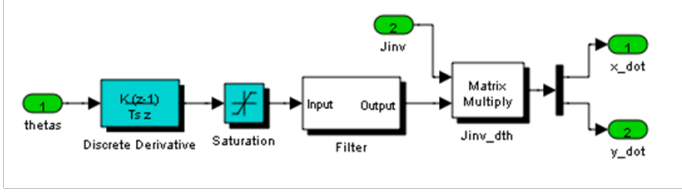


Fig. 6. End-effector velocity estimation in Simulink

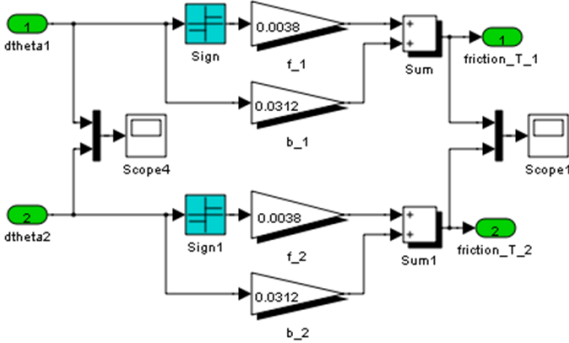


Fig. 7. Friction compensation at the active joints of the 2-DOF haptic device

physical restriction to the movement of the device in the vertical direction. These models are preferred in haptics for their computationally efficient nature, ease of modelling and stability features [8], [9]. An extra-mass (70g) was attached to the end-effector for increasing the inertial effects during the experiments. The end-effector was allowed to fall freely under gravity. This type of experiment being simple, does not require the inclusion of human impedance $H(s)$ in the controller analysis. The device fell under its own weight until the virtual wall was encountered in the horizontal plane. As the end-effector reached the wall, a reaction torque was commanded at the active joints corresponding to the reaction force at the end-effector. The reaction force was estimated using a simple haptics force law given as

$$\mathbf{f} = \mathbf{K}\mathbf{x} + \mathbf{B}\dot{\mathbf{x}} \quad (1)$$

where, \mathbf{f} denotes the vector of reaction forces at the end-effector and \mathbf{K} , \mathbf{B} are the stiffness and damping coefficient matrices of the virtual wall respectively. The terms \mathbf{x} and $\dot{\mathbf{x}}$ denote the vector of depth of penetration (inside the virtual wall) and vector of velocity of the end-effector respectively. They are given as

$$\mathbf{f} \equiv \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \quad \mathbf{K} \equiv \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}, \quad \mathbf{B} \equiv \begin{bmatrix} b_x & 0 \\ 0 & b_y \end{bmatrix}, \quad (2)$$

$$\mathbf{x} \equiv \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad \dot{\mathbf{x}} \equiv \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

The reaction force in (1) was pre-multiplied by the transpose of Jacobian matrix for the 2-DOF device, which provided the

required joint torques as

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f} \quad (3)$$

where

$$\boldsymbol{\tau} \equiv \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \text{and} \quad \mathbf{J} \equiv \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{bmatrix} \quad (4)$$

whereas \mathbf{f} is given in (2), the elements of \mathbf{J} in (4) are available in [10]. It must be noted that force \mathbf{f} in (1) and the corresponding torque $\boldsymbol{\tau}$ in (3) is zero for the case when the user is outside the virtual wall.

For stable output, the end-effector position and velocity showed damped oscillations, while for unstable output, oscillations of constant magnitude were observed. For a particular value of virtual damping coefficient, virtual stiffness could be increased until the oscillatory behavior is observed. This value is recorded as the stable stiffness at that particular value of damping coefficient. Similarly the whole Z-width for the device could be generated. The incorporation of the quantization modules, zero-order hold blocks available in Simulink enabled us to study the effect of encoder resolution and the sampling-rate of the controller on the stability of the haptic device.

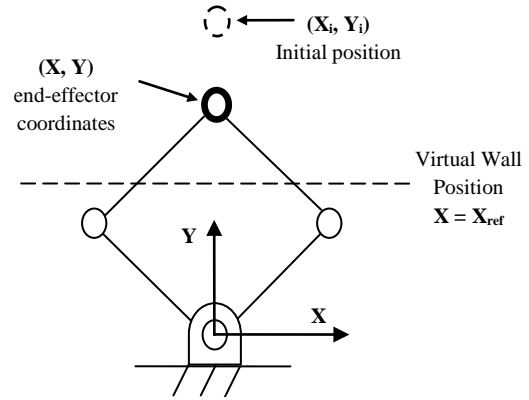


Fig. 8. Virtual Wall representation for the 2-DOF

VI. RESULTS

We plotted y-component of the position and velocity of the end-effector in each trial of the simulation. We also plotted reaction force from the virtual wall on account of being hit by the end-effector while falling under gravity. Since the motion of the end-effector was vertical and the virtual wall was also defined as obstruction along y-axis, the x-component of the reaction force was almost negligible during the complete set of trials, i.e., $f_x = 0$ in (2). Figure 9 depicts the output details of the stable virtual-wall interaction. As seen from Fig. 9, the y-position of the end-effector dropped from an initial position till the virtual wall was encountered. As the virtual wall was hit, oscillations in position and velocity were seen for around 0.3 seconds after which a stable output was achieved. The interaction took place at nearly 0.13 seconds, since the reaction torque

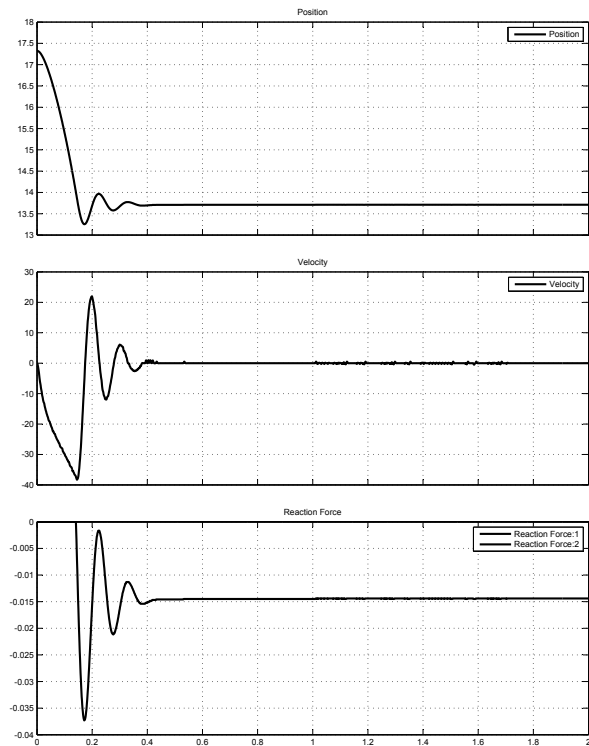
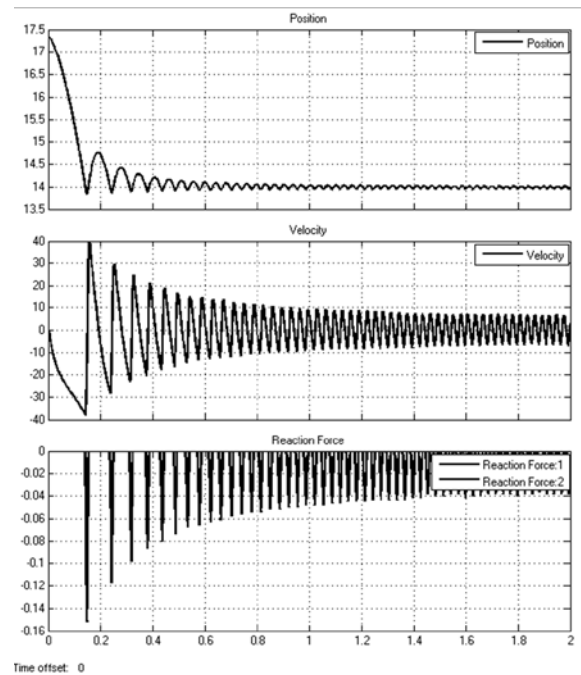


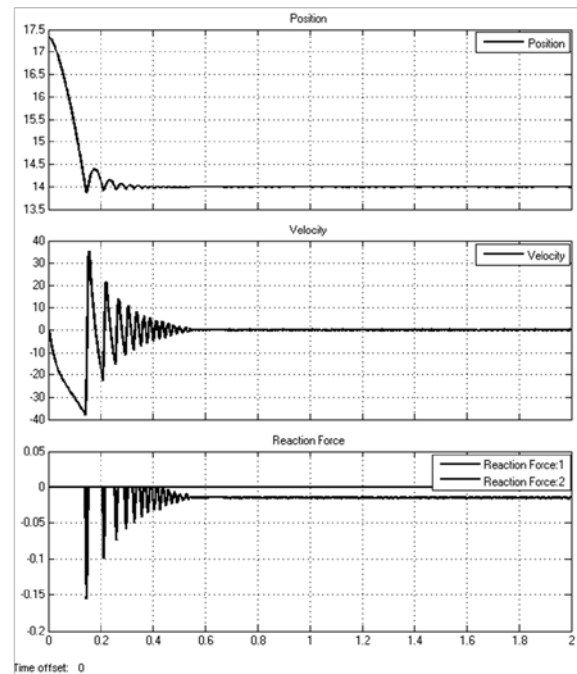
Fig. 9. Output of a stable interaction with a virtual wall

till 0.13 seconds was zero. Figure 10(a) depicts an unstable output for higher values of the virtual wall parameters. Fig. 9 and 10 depict the tests carried out for stiffness k_y only, i.e., $k_x, b_x, b_y = 0$. As the virtual stiffness of the wall k_y was increased beyond a particular value, instability was observed. This instability was seen as oscillations of constant magnitude in position, velocity and the reaction force from the virtual wall. In this case, the oscillations did not die even after 2 seconds of the simulation.

The higher magnitude of the virtual wall parameters was seen from the initial value of the reaction force in Fig. 10(a), which was very high compared to the initial value of reaction force in Fig. 9. Table 1 enlists the values of parameters used for respective simulations. With the addition of virtual damping, the stable range of virtual stiffness has increased. The effect can be seen in Fig. 10(b), where the virtual stiffness value was kept similar to the one used for the results of Fig. 10(a). However, an addition of virtual damping was made to the haptics force law. This resulted into a nearly stable interaction with the virtual wall. Hence the results are in confirmation with the theory proposed by Gill et al. [6]. Towards the end, a line diagram of the similar device built purely in the SimMechanics environment which is depicted



(a) Instable output with no virtual damping



(b) Stable output with non-zero virtual damping

Fig. 10. Effect of virtual damping on the stability of haptics controller

TABLE I. PARAMETERS OF THE VIRTUAL WALL USED IN SIMULATIONS

Parameter	Fig. 9	Fig. 10(a)	Fig. 10(b)
k_y (N/m)	0.1	1.0	1.0
b_y (Ns/m)	0.0	0.0	0.0005

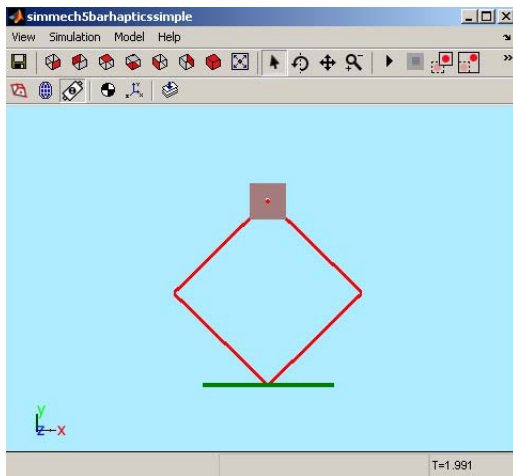


Fig. 11. Two-DOF haptic device modeled in SimMechanics

in Fig. 11. Inertial and kinematic parameters were substituted similar to the real/CAD model of the device. Simulation results with this device were same to those obtained from the imported CAD model in SimMechanics (Fig. 9 - 10).

VII. CONCLUSIONS

In this paper, we have provided a systematic way of modeling a haptic device and the associated haptics controller in SimMechanics and Simulink. In particular, we have demonstrated the development of a multi-DOF haptic controller in Simulink. A 2-DOF haptic device was imported from CAD environment and a similar device was built using the basic components of the SimMechanics toolbox.

The multi-DOF controller was tested for a particular set of gains to establish a stable interaction with a virtual wall. An unstable output for higher gains was also established. SimMechanics and Simulink were thus found to be effective tools for modeling, testing and validating the controller for a particular haptic device. Not only these tools allow us to simulate and visualize a haptic device in a realistic way, but also these software allow us to go beyond reality, i.e., we may build experimental environments according to our own imagination, using the tools available in the simulation library.

In future, we plan to test a higher DOF system as a haptic device and plan to put an in-house dynamics algorithm to compare the efficiency of the algorithm behind SimMechanics.

REFERENCES

- [1] J. E. Colgate, J. M. Brown, "Factors affecting the Z-Width of a haptic display," Proc: IEEE International Conf. Robotics and Automation, pp. 3205-3210, 1994.
- [2] The MathWorks Inc. SimMechanics User's Guide March, 2007. <http://www.mathworks.in/products/simmechanics/index.html>
- [3] Yang Chifu, Ye Zhengmao, O. O. Peter, Han Junwei, "Modeling and simulation of spatial 6-DOF parallel robots using Simulink and SimMechanics," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on , vol.4, pp. 444 - 448, 9-11 July, 2010.
- [4] Deng Wenbin, Lee Jae-Won, Lee Hyuk-Jin, "Kinematics simulation and control of a new 2 DOF parallel mechanism based on MATLAB/SimMechanics," Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on , vol.3, pp.233 - 236, 8-9 Aug. 2009.
- [5] J. Kardos, "The simplified dynamic model of a Robot manipulator," In: Proc. 18th Annual Int. Conf on Technical Computing Bratislava, 2010. http://dsp.vsb.cz/konference_matlab/MATLAB10/
- [6] J. J. Gill, A. Avello, A. Rubio, J. Florez, "Stability analysis of a 1-DOF haptic interface using the Routh-Hurwitz criterion," Control Systems Technology, IEEE Trans., vol. 12(4), pp. 583-588, 2004.
- [7] The Math Works Inc. Importing CAD models-SimMechanics for MATLAB and Simulink. <http://www.mathworks.in/products/simmechanics/description3.html>
- [8] M. A. Srinivasan, C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges." Computers & Graphics, vol. 21(4), pp. 393-404, 1997.
- [9] B. Federico, D. Prattichizzo, and K. Salisbury, "A multirate approach to haptic interaction with deformable objects single and multipoint contacts." The International Journal of Robotics Research, vol. 24(9), pp. 703-715, 2005.
- [10] M. Koul, D. Rabinowitz, S. K. Saha, M. Manivannan, "Synthesis and design of a 2-DOF haptic device for simulating epidural injection," In: Proc. 13th World Congress in Mechanism and Machine Science, Guanajuato, Mexico, pp. 1-7, 2011.