A Position-Control Based Approach to Haptic Rendering of Stiff Objects

Yang Wang, *Student Member, IEEE*, Lei Feng, *Member, IEEE*, and Kjell Andersson

***Abstract*—Conventional force rendering methods in haptic applications often suffer stability issues when simulating interactions with stiff objects such as a virtual wall. This paper argues that the emphasis in such scenarios is to minimize the penetration into the virtual wall instead of modeling the wall as a spring-damper system. Therefore, we propose an approach using a position controller to achieve better haptic rendering of the virtual wall. The proposed approach exploits model-based development tools to obtain the linear control system model without the need for an analytical model of the dynamics of the haptic device. A simulation-based performance comparison of two different controllers has been made for a 6-DOF parallel structure haptic device.**

***Index Terms*—Haptics, force rendering, kinesthetic devices, dynamic systems and control.**

I. Introduction

HAPTICS is the study of the sense of touch and it contains a wide range of interesting subtopics. This paper uses the term "haptic feedback" mostly for force feedback, which enables interactions with a virtual environment by holding a hand tool. Analogous to visual rendering, haptic rendering generates some haptic sensations, such as a mouse click, a gun recoil, or a collision. Thus, haptic devices can be found in applications range from daily mobile devices [1], [2] to specialized devices such as game controllers [3], [4], research devices [5]–[7], and dental simulators [8]–[10]. The concept of haptic feedback is also often mentioned in other related areas like tele-robotics, master-slave control, and bilateral control.

The origin of the commercial haptic device, Phantom, can be dated back to at least 25 years ago [11]. However, the solution to one of the major challenges like stably rendering contact with a stiff surface still remains to be improved before the technology is safe to use in areas like surgical robotics. The conventional haptic rendering method contains 5 steps, namely, position sensing (in joint space), position computing (in workspace), collision detection, interaction force computing, and actuators/motors reference signal computing [12]. The reference signal here is often a motor torque signal, so the subsequent motor control is basically a force control. Given the fact that there is normally no torque sensor used in the control system to monitor either the torque output from the motor or the force felt by the user, the force control is done in an open-loop fashion and the quality of the rendering is not monitored. The effect of the rendered force is to counteract the user's force so that the haptic tool will stop on the surface. The interacted virtual surface is usually modeled as a spring. Thus the rendered force applied to the haptic tool is proportional to the displacement of the tool into the surface. If the surface has high stiffness, a small displacement can result in a large force, which may make the system unstable.

Another challenge lies in the interaction between the continuous-time physical world and the discrete-time virtual world, generating extra energy when all systems are considered passive. This phenomenon is termed "energy leak" by Gillespie and Cutkosky [13]. Many effects contributing to this phenomenon, such as sensor quantization and time discretization, have been studied and passivity-based methods have been developed [14]. Colgate *et al.* [15] propose "virtual coupling" to put an upper bound to the impedance of the haptic device. It serves like a filter between the virtual world and the device; hence it sometimes distorts the forces in magnitudes and directions in an undesirable way to create artifacts. Adams and Hannaford [16] later extend this impedance-focused approach to also include the admittance model of haptic interaction. However, this method cannot easily extend to render a dynamically changing environment. Hannaford and Ryu [17] propose time-domain passivity control, which deploys an observer to monitor the extra energy in the system in real time and a variable damping to dissipate that. Later, Ryu and Yoon [18] extend this further into a memory-based passivity approach with special hardware to quickly save a huge amount of data. By recording the interacting position-force graph between the device and the environment, this method guarantees that the net energy coming from the virtual environment into the system is always less than 0. They acknowledge that the application for the method should emphasize more on stability than transparency, because the resulting behavior is still conservative.

For the implementation, keeping the haptic rendering loop operating at 1000 Hz also proves to be quite challenging. The dynamics of the device, which is frequently used in the rendering loop, can be relatively slow to compute because many nonlinear processes are involved. In addition, the dynamics of a parallel mechanism (PM) is mostly estimated by iterative or recursive

Manuscript received February 28, 2020; revised August 8, 2020 and October 16, 2020; accepted November 29, 2020. Date of publication December 14, 2020; date of current version September 16, 2021. This article was recommended for publication by Associate Editor Prof. Matteo Bianchi and Editor-in-Chief Prof. Domenico Prattichizzo upon evaluation of the reviewers' comments. *(Corresponding author: Yang Wang).*

The authors are with the Department of Machine Design, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: wang5@kth.se; lfeng@kth.se; kan@kth.se).

Digital Object Identifier 10.1109/TOH.2020.3044682

methods, which introduce error and slow down the computation further when it is hard to converge. For well-known PMs like the Stewart-Gough platform [19] or Agile-Wrist [20], a closed-form (or explicit/analytical) solution of the dynamic model can be derived, thus the evaluation is faster than using numerical methods. However, such solution relies on the symmetry of the actual mechanism and cannot be generalized to work on more general PMs. Cheng *et al.* [21] obtain the closed-form dynamic model of a 2-DOF redundantly actuated planar PM because the structure is relatively simple and has symmetry. The results for 4 different control structures verify that shortening the execution time will improve accuracy in position/velocity. They achieve that by bringing a part of the computation offline. Corbel *et al.* [22] simplify the dynamic modeling of a well-known PM, Delta, even more by making two main assumptions: the inertia of the "forearm" linkages is ignored and their mass is split and combined into their connected neighbors. This avoids calculations associated with these linkages, making the control meet its execution time requirement. This is a practical and reasonable compromise because their application is high-speed high-acceleration and shorter execution time is valued more than tracking accuracy. Instead of modeling the full dynamics, Abdellatif and Heimann [23] identify a reduced set of dynamic parameters to increase the efficiency of the identified dynamic model. The optimal set of parameters is reduced to 10 rigid-body and 14 friction-related parameters for a 6-DOF PM. In a similar fashion, Diaz-Rodriguez *et al.* [24] propose a workflow to identify a "reduced model" for a 3-DOF PM. It is an iterative process and the reduced set is achieved by checking whether the inertial matrix is positive definite and, if not, removing the parameter with the largest standard deviation. The results again verify that less computational load is important for real-time implementation. All of these methods aim to achieve a faster dynamic evaluation within a typical haptic rendering loop.

In summary, the traditional haptic rendering computes interacting force to stop the haptic tool on the surface. Passivity-based methods guarantee stability but sacrifice transparency. The methods aiming at faster rendering loop usually require some special properties in the actual mechanism or user case, making them inapplicable to other mechanisms or cases. To overcome some of the shortcomings of the previously mentioned methods, we propose a general modeling approach applicable to any structure and a position controller to regulate directly the position of the haptic tool. The idea of proposing position-controller-based haptic rendering is to render stiff interaction by avoiding penetration into the surface instead of computing force based on penetration. Noted that our study focuses on interaction force computation. Other steps in conventional haptic rendering are considered solved.

The investigation of this approach with a position controller for haptic rendering has been formulated as two research questions (RQs):

RQ 1: Can haptic rendering of stiff interaction be solved as a position controller design problem?

RQ 2: How much stiffness can this new approach achieve?

To answer these research questions, this paper is organized as follows: we briefly introduce a test case haptic device called TAU in Section II and Section III presents the intuition behind the proposed approach. The whole controller design process from plant modeling to discrete time design is shown in details in Section IV. Section V shows the methodology in conducting the simulations and motivates the choice of performance metrics to evaluate. Results for different approaches on a virtual prototype under two different scenarios are compared in Section VI, followed by a co-simulation verification in Section VII. These results are further discussed in Section VIII. Finally, conclusions are presented in Section IX along with a list of future work.

II. Description of Test Case

The test case for this paper is a desktop 6-DOF parallel-structure haptic device called TAU that has been previously developed in the group at KTH for virtual dental training [25], [26]. It is a general PM that contains a base fixed to the ground, a moving platform, and several chains that connect the two, which then create kinematic closed loops. This is opposed to serial manipulators which only have one chain or kinematic open loop. TAU has 3 kinematic chains that are not identical to each other (see Fig. 1). In each chain, the link connected to the base is always called the link 1 in that chain with length L_1 and link 2 is the one connected to the platform with length L_2. Such asymmetric structure distinguishes TAU from most well-known PMs and also results in a more complex and computationally heavy dynamic model, making it a good test case for the purpose of this paper. The base of TAU is the I-column and each chain has 2 actuators close to or fixed onto the base to minimize mass and inertia effect on the moving platform. Optical encoders are used at each active joint to measure the joint angles. The moving platform is equipped with a tool handle that emulates the surgical tool handle for the trainee and the tool center point (TCP) is defined at the center of the platform.

A. Device Specifications

The TAU haptic device is developed with the aim to provide force/torque (F/T) feedback to help dental students develop necessary motor skills for the real dental operations while training in a simulated virtual environment (VE). For this type of applications, a singularity free workspace of a 50 mm cube is enough while keeping the footprint of the device relatively small. The origin for the workspace coordinate frame is defined to be in the center of the cubic workspace. Within the workspace, the moving platform can achieve ± 10 degrees of rotation along all axes with respect to (w.r.t) the initial pose. The device itself can fit in a $250 \times 250 \times 300$ mm^3 box. A schematic of the kinematic structure with the main design parameters is given in Fig. 1 and the design parameter values is given in Table I.

B. Definitions in Kinematic and Dynamic Model

The task of haptic rendering can be treated as a simple control problem: given the TCP motion in the Cartesian space,

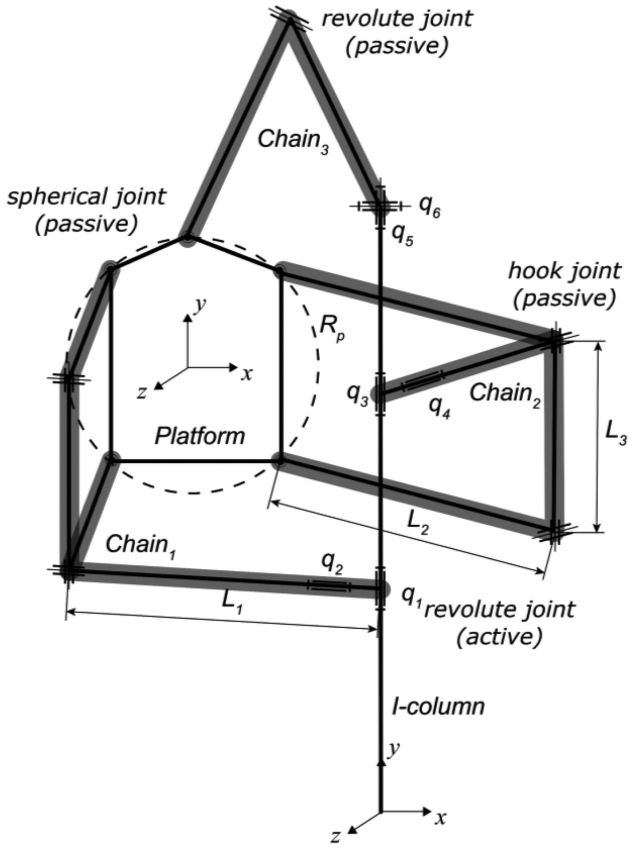

Fig. 1. Kinematic diagram of TAU with some of the design parameters from Table I. The active joints, driven by actuators, are denoted with q_1 to q_6. Other types of joints are also labelled.

TABLE I
DESIGN PARAMETERS

Parameter	Value
Radius of the platform, R_p	55 mm
Length of link 1, L_1	138 mm
Length of link 2, L_2	180 mm
Length of link 3, L_3	91.5 mm
Length of column I, $3L_3$	274.5 mm

driven by both the user and the device itself, design a controller that generates a motor torque reference so that the user feels the right haptic feedback. Without losing generality or consistency with previous work [25], the TCP pose in Cartesian space can be represented by a 6-by-1 vector

$$\mathbf{X} = \left[p_x, p_y, p_z, \alpha, \beta, \gamma\right]^T$$

where (p_x, p_y, p_z) denotes the position coordinates and (α, β, γ) the orientation angles of the tool along x, y, and z axis, respectively. The motion of the TCP is actuated by the active joints and the mapping from Cartesian space to joint space representation is the inverse kinematics. In the joint space, the pose of the device is also defined by a 6-by-1 vector

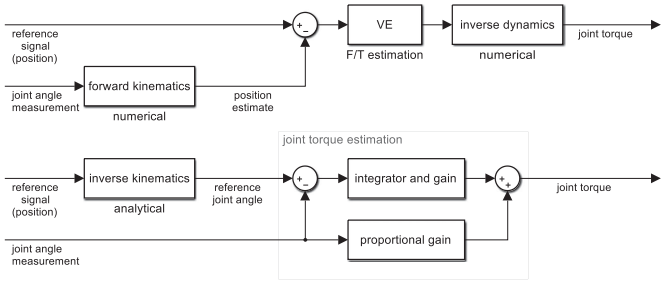

Fig. 2. Illustration of the common approach (top) and the proposed approach (bottom).

$$\mathbf{q} = \left[q_1, q_2, q_3, q_4, q_5, q_6\right]^T \tag{1}$$

where q_i denotes the angular position of joint i. These are the active joints that govern the motion of the device while the rest of the joints are passive. Accordingly, the force to be rendered here is a 6-DOF force acting on TCP in Cartesian space, defined by a 6-by-1 vector

$$\mathbf{F} = \left[F_x, F_y, F_z, T_x, T_y, T_z\right]^T$$

and the joint torque to achieve that is defined as

$$\boldsymbol{\tau} = \left[\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\right]^T \tag{2}$$

where τ_i denotes the joint torque on joint i. The mapping in between is the Jacobian matrix $\boldsymbol{J}$, which is commonly found in kinematics studies and calculated in [25].

$$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{F} \tag{3}$$

III. An Approach using Position Control

In the scenario of interacting with a stiff virtual wall, which we are using in this paper, a common approach is to model the virtual wall as a spring-damper system, as shown in the top part of Fig. 2. However, this approach can be problematic when trying to render a wall with very high or even infinite stiffness, because a small displacement can lead to huge force output.

In this paper, we continue to develop the approach presented in a previous paper [27]. We have selected a simple but representative case where we render the interaction with an ideal stiff wall with infinite stiffness as an initial step for investigating the formulated research questions. Conventional haptic rendering approaches estimate the contact forces in Cartesian space that occur during the interaction. However, our approach focuses on the positional aspects during the interaction. If the wall is ideal with infinite stiffness, the TCP cannot penetrate or deform it. Then the desired behavior of rendering a stiff wall can be achieved by designing a position controller to regulate the position of TCP fixed at the collision point once the collision happens. As indicated in Fig. 2, the signals coming into and out of the proposed controller are still the same with the traditional one but instead of estimating the interaction force in the Cartesian space, the controller

TABLE II
CONTROL REQUIREMENTS

Requirement	Value
Overshoot	< 2%
Rise time	< 0.05 s
Settling time	< 0.1 s
Joint position deviation	< 0.001 rad
TCP position deviation	< 1 mm in all xyz axis directions
TCP orientation deviation	< 0.001 rad in all xyz axes

guarantees no penetration (in position) to automatically generate the joint torque needed to achieve that.

Because the proposed position controller keeps the TCP fixed at one point with no motion, the user-applied force to the TCP should be equal but opposite to the wall-applied force, which is realized by the actuators. The TCP must not penetrate into the stiff wall. Thus, the output of a position controller is equivalent to that of a haptic rendering algorithm of a stiff wall. With this change of view, all methods related to position controller design can be applied to deal with this haptic rendering problem.

For a position controller, the control goal is to keep the TCP at a given point in the workspace and reject disturbance on the TCP. Here we consider user-applied force as disturbance instead of input to the system. The TCP shall not move if the user-applied force can be counteracted within the saturation limits of the actuators. This gives the control goal in the Cartesian space while the control algorithm is formulated in joint space since the motion of the TCP is tracked by joint encoders. Therefore, the position reference in the Cartesian space is mapped into joint space by the inverse kinematics of TAU.

The controller is developed with typical requirements for a position controller. For a step position reference, overshoot should be less than 2%, rise time less than 0.05 s, and settling time less than 0.1 s. Some of the parameters here are chosen from a pure control stand point while others are based on studies on human perception limitations [28]. Specifically, the settling time is based on the result reported in [29] that the minimal noticeable delay can be 61 ms for system delay and 132 ms for "front-end" delay. Additionally, the controller should be robust against random and sudden change of the user force. This yields the requirement that when there is F/T step disturbance at the TCP, the maximal deviation should be less than 0.001 rad for every joint. This corresponds to position deviation less than 1 mm and orientation deviation less than 0.001 rad for the TCP along all axes in workspace at the origin. These requirements are also listed in Table II.

IV. CONTROL DESIGN

The control design process can be divided into two major phases: plant modeling and discrete control design. The first phase is to model the control plant. Based on the control plant model, the second phase deals with designing the controller.

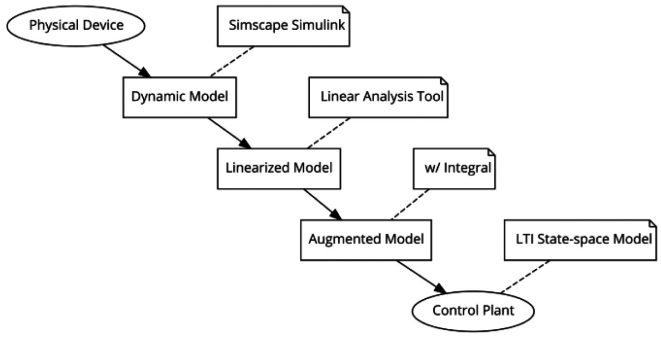

Fig. 3. Flow chart of generating control plant.

Although the two phases are in this order, the controller chosen to be designed in the second phase actually affects the requirement for modeling in the first phase. Because we have already decided the structure of our controller at this point to be an LQR-based position controller due to its advantage in penalizing large control input, the plant model we need to develop for that controller should be a linear time-invariant (LTI) state-space model [30].

A. Plant Modeling

Traditional modeling of control plant involves a mathematical formulation of the device dynamics. This can be tedious and prone to human error especially for a parallel mechanism. In addition, the obtained model is often highly non-linear and inapplicable as a control plant without some simplifications. Our approach avoids this step of mathematical formulation of the dynamic model. Instead, it starts by building a simulation model and then extract the LTI state-space model from it with the help of some tools and reformulation. The steps in this phase are illustrated in Fig. 3 and further elaborated as follows.

1) Development of a Dynamic Model: The first step is to develop an implicit dynamic model of TAU as an alternative to a closed-form dynamic model with explicit dynamic equations. Here we choose the commercial multi-body dynamics software Simscape, which is a modeling module within the MATLAB/Simulink environment. One of the main reasons to choose Simscape is that it supports the modeling of PMs. It also gives us a model that is compatible with the MATLAB linearization tool used in the next step.

The process of building the dynamic model in this simulation environment is rather straightforward. The way of modeling is based on the idea of building a "kinematic tree" where the root is the base of the robot and the leaf is the end-effector. This approach is common for serial mechanisms (SM), hence the term "end-effector" is used. To generalize this to PMs, we still build the PM as a tree first with all the chains unconnected as branches. This middle step is illustrated in Fig. 4, where links and joints are drawn with rectangular boxes and oval circles, respectively. All the active joints are revolute joints and all the joints without the "active" label are passive joints. In the figure, Chains 1 and 2 are only drawn once because they have similar mechanical structure. Every branch ends

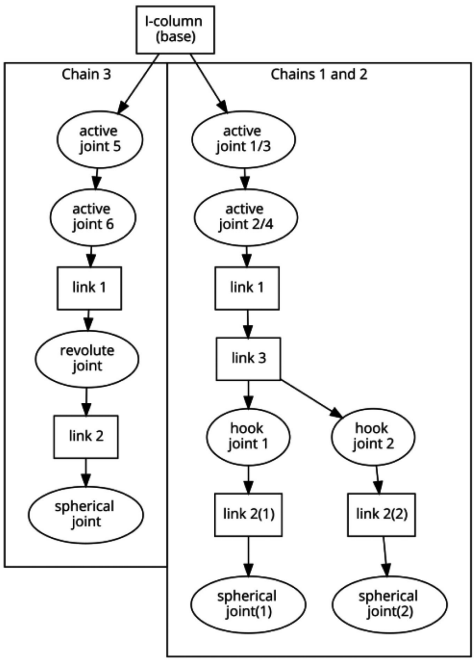

Fig. 4. An illustration of a middle step in dynamic modeling to smoothly transit from kinematic schematics to a Simscape model.

with a spherical joint as the leaf node. Then the leaf nodes are connected to the moving platform, which forms kinematic closed loops. This gives us a PM that finally resembles the kinematic structure illustrated in Fig. 1. In detailed implementation, each branch is built with defining links and joints as Simscape blocks with connectivity, geometric, and mechanical properties. The process of connecting blocks makes it easier than propagating the rigid body equation of motion from root to leaf by hand. The inputs/outputs (I/Os) are defined to match the physical system of TAU, having joint torque as input and encoder reading as output.

2) Linearization of the Dynamic Model: The second step is to linearize the dynamic model generated from the previous step at an operational point (OP). We use the Linear Analysis Tool (LAT) in MATLAB to accomplish this linearization. The OP is chosen to be in the center of the cube-shaped workspace. It should be noted that we only linearize the model at one OP for the current study. The output from the LAT is an LTI state-space model of the full dynamics, which can be expressed as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \tag{4}$$

This is just the general form of an LTI state-space model. The input to the Simscape model is $\mathbf{u} = \tau$ as defined in (2) and the output is $\mathbf{y} = [\dot{\mathbf{q}}^T, \mathbf{q}^T]^T$, where $\boldsymbol{q}$ is defined in (1). The $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ matrices are constant coefficients of the LTI model. When applying LAT to derive the linear model, the state vector $\boldsymbol{x}$ is a 12×1 vector decided by LAT.

Because there are multiple sets of states that could fully define the state of the system, LAT picks a state vector by its internal algorithm. However, the picked state vector may not be convenient for the control design because some of the states are hard to measure or observe in the physical system. This means that we need to convert the automatically generated state-space model through linear transformation to a new state vector that is directly related to the joint position and velocity.

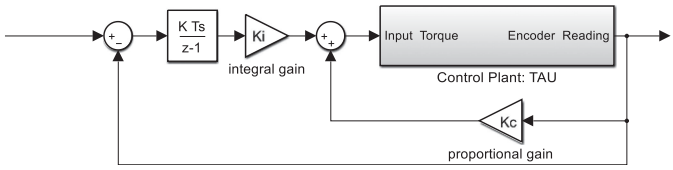

Fig. 5. Control block diagram of an LQR controller with integrator for the MIMO system.

In implementation, we check that $\mathbf{C}$ is invertible so we get $\mathbf{x} = \mathbf{C}^{-1}\mathbf{y}$.

Left multiply $\boldsymbol{C}$ to first equation of (4) and we get

$$\mathbf{C}\dot{\mathbf{x}} = \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\mathbf{u}$$

Substitute all the terms related to $\boldsymbol{x}$ and we get

$$\dot{\mathbf{y}} = \mathbf{C}\mathbf{A}\mathbf{C}^{-1}\mathbf{y} + \mathbf{C}\mathbf{B}\mathbf{u} \tag{5}$$

which also has the form of the first equation of (4)

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\mathbf{u} \\ \tilde{\mathbf{y}} = \tilde{\mathbf{C}}\tilde{\mathbf{x}} \end{cases}$$

where

$$\tilde{\mathbf{x}} = \mathbf{y}, \tilde{\mathbf{A}} = \mathbf{C}\mathbf{A}\mathbf{C}^{-1}, \tilde{\mathbf{B}} = \mathbf{C}\mathbf{B}, \tilde{\mathbf{C}} = (\, 0_{6\times6} \quad \mathbf{I}_{6\times6} \,)$$

3) Augmentation of the Linearized Model: Standard state feedback control of linear systems has difficulty to regulate the output at arbitrary reference values. To overcome the limitation, an integral action is added in the state feedback control. An illustration of the new control structure is given in Fig. 5. The integration of the error between the reference and the output is included in the control input. Similar to the integration part of the PID control, the integration of the error will ensure zero steady-state error. To adapt to the state feedback control structure, the integral action is considered as an augmented state vector.

Let $\boldsymbol{r}$ be the vector of reference values for the six joints. We define the error between the reference and the output as

$$\Delta = \mathbf{r} - \mathbf{q} \tag{6}$$

and the additional integral states as

$$\mathbf{x}_i = \int \Delta dt = \int (\mathbf{r} - \mathbf{q}) dt$$

evidently,

$$\dot{\mathbf{x}}_i = \Delta = \mathbf{r} - \mathbf{q}$$

The augmented state-space model of the system (noted with a subscript) becomes

$$\dot{\mathbf{x}}_{ad} = \mathbf{A}_{ad}\mathbf{x}_{ad} + \mathbf{B}_{ad}\mathbf{u} + \mathbf{B}_{ref}\mathbf{r} \tag{7}$$

where the augmented state vector is extended with the additional integral states

$$\mathbf{x}_{ad} = \left[\tilde{\mathbf{x}}^T, \mathbf{x}_i^T\right]^T \tag{8}$$

and the state-space model matrices are

$$\mathbf{A}_{ad} = \left(\begin{array}{c|c} \tilde{\mathbf{A}}_{12\times 12} & 0_{12\times 6} \\ \hline 0_{6\times 6} \quad -\mathbf{I}_{6\times 6} & 0_{6\times 6} \end{array}\right)$$

$$\mathbf{B}_{ad} = \begin{pmatrix} \tilde{\mathbf{B}}_{12\times 6} \\ 0_{6\times 6} \end{pmatrix}, \mathbf{B}_{ref} = \begin{pmatrix} 0_{12\times 6} \\ \mathbf{I}_{6\times 6} \end{pmatrix}$$

Up to this point, we have built a high-fidelity simulation model that captures system dynamics, linearized it to an LTI state-space model, and augmented it to eliminate steady-state error. This is our control plant that will be used in the next step to design a discrete time LQR-based position controller.

B. Discrete Time Design

The second phase is to design a controller for the control plant. As stated before, we choose to design a centralized LQR position controller because it has the advantage of considering the coupling effect among the joints and penalizing large control input. By carefully designing the weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ in (12), a balance between small control input and high accuracy can be reached to keep the motors away from saturation if possible. However, the control plant we generated in the previous phase is still in continuous time. So we need to first transform the plant model to discrete time and then design the controller in discrete time.

1) Discrete Time Control Plant: As the first step in this phase, the discrete-time state-space model of the control plant is obtained by the Zero-Order Hold (ZOH) method because it provides an exact match between the continuous- and discrete-time systems in the time domain for staircase inputs. ZOH implies

$$\mathbf{u}(t) = \mathbf{u}[j], jT_s \le t \le (j+1)T_s \tag{9}$$

where *j* is the discrete time step. Then we have

$$\mathbf{x}_{ad}[j+1] = \mathbf{A}_{adDT}\mathbf{x}_{ad}[j] + \mathbf{B}_{adDT}\mathbf{u}[j] + \mathbf{B}_{refDT}\mathbf{r}[j] \tag{10}$$

where

$$\mathbf{A}_{adDT} = e^{\mathbf{A}_{ad}T_s}$$

$$\mathbf{B}_{adDT} = \left(\int_0^{T_s} e^{\mathbf{A}_{ad}t}dt\right)\mathbf{B}_{ad}$$

$$\mathbf{B}_{refDT} = \left(\int_0^{T_s} e^{\mathbf{A}_{ad}t}dt\right)\mathbf{B}_{ref}$$

The e here is the natural constant. The subscript is extended from "ad" to "adDT" to denote that it is the augmented model in discrete time.

2) Control Problem Formulation: The second step is to design the controller in the discrete time domain. The control goal of this controller, as discussed earlier, is to keep the TCP fixed at the collision point so that the deviation off that point is minimized under random disturbance. The control problem is formulated as an infinite-horizon discrete-time LQR to find the optimal gain matrix $\boldsymbol{K}_{adDT}$ such that the state-feedback control law

$$\mathbf{u}[j] = -\mathbf{K}_{adDT}\mathbf{x}_{ad}[j] = -(\,\mathbf{K}_c \,|\, \mathbf{K}_i\,)\begin{pmatrix} \tilde{\mathbf{x}}[j] \\ \mathbf{x}_i[j] \end{pmatrix} \tag{11}$$

minimizes the quadratic cost function

$$Cost(\mathbf{u}) = \sum_{j=1}^{\infty}\left(\mathbf{x}_{ad}^T[j]\mathbf{Q}\mathbf{x}_{ad}[j] + \mathbf{u}^T[j]\mathbf{R}\mathbf{u}[j]\right) \tag{12}$$

for the discrete-time state-space model demonstrated in (10). This is calculated by first solving the associated Discrete-time Algebraic Riccati Equation [29] that is

$$\mathbf{P} = \mathbf{A}^T\mathbf{P}\mathbf{A} - \left(\mathbf{A}^T\mathbf{P}\mathbf{B}\right)\left(\mathbf{R} + \mathbf{B}^T\mathbf{P}\mathbf{B}\right)^{-1}\left(\mathbf{B}^T\mathbf{P}\mathbf{A}\right) + \mathbf{Q}$$

where $\boldsymbol{P}$ is the solution to the equation and the "adDT" subscript for $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices is dropped for keeping the equation clean. With $\boldsymbol{P}$, $\boldsymbol{K}_{adDT}$ is achieved through

$$\mathbf{K}_{adDT} = \left(\mathbf{R} + \mathbf{B}_{adDT}^T\mathbf{P}\mathbf{B}_{adDT}\right)^{-1}\left(\mathbf{B}_{adDT}^T\mathbf{P}\mathbf{A}_{adDT}\right)$$

It should be noted that in implementation, $\boldsymbol{K}_{adDT}$ is obtained by the LQR solver offered by MATLAB. The LQR function in MATLAB takes the state-space model $\boldsymbol{A}_{adDT}$ and $\boldsymbol{B}_{adDT}$ as well as the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices as input arguments and computes $\boldsymbol{K}_{adDT}$ which is later split into $\boldsymbol{K}_i$ (the gain w.r.t the integral states) and $\boldsymbol{K}_c$ (the gain w.r.t the rest of the states).

The $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are initialized as identity matrices and tuned to satisfy the control requirement listed in Table II. Based on numerous systematic tests, we get

$$\mathbf{Q} = \begin{pmatrix} \mathbf{I}_{6\times 6} & 0_{6\times 6} & 0_{6\times 6} \\ 0_{6\times 6} & 10^3 \times \mathbf{I}_{6\times 6} & 0_{6\times 6} \\ 0_{6\times 6} & 0_{6\times 6} & 10^7 \times \mathbf{I}_{6\times 6} \end{pmatrix}$$

$$\mathbf{R} = \mathbf{I}_{6\times 6}$$

The tuning is based on the objective that we need a better accuracy for position tracking than for velocity and a small penalty on high control input. Note that the relative ratio between elements in the two matrices is more important than the exact value. If $\boldsymbol{R}$, the penalty on high control input, is larger, the system cannot meet the rise time requirement. If the entries in $\boldsymbol{Q}$ w.r.t error integration are smaller, the system cannot meet the time requirement for recovering from disturbance at TCP. If the entries in $\boldsymbol{Q}$ w.r.t joint position are smaller, the system cannot meet the deviation requirement.

As a de facto standard for haptic rendering applications, the controller is designed with a sampling frequency of 1000 Hz [14], [30]. For our designed controller in Fig. 5, the fastest closed-loop pole is around 2480 rad/s, which is well below half of the sampling frequency of 6280 rad/s.

Apart from developing the proposed controller, we also implemented a baseline controller to be used as a reference for comparison with our newly proposed controller. The baseline controller used in this paper is from [32], which is also used as a comparison controller in [33]. It is an impedance controller that models the stiff wall as a hard spring with a certain stiffness. When the tool has penetration into the stiff wall, the responsive force to push the tool away from the wall is computed as the multiplication of the penetration distance and the stiffness according to Hooke's law. The responsive force on the tool is mapped to joint torques. The stiffness of the wall is set to be 6000 N/m and the reason behind the choice is explained in later sections.

V. Simulation Methodology

As we mentioned in the previous section, when we generate the control plant, we also build a Simscape model of the device, which can be used for simulation verification. This simulation result is later verified by a MATLAB-ADAMS co-simulation, which will be presented in Section VII.

To verify the advantage of the proposed approach, simulations of the TCP pushing a stiff wall are studied. Because we formulate the haptic rendering problem as a position control problem, the human force exerted on the TCP is considered a disturbance to the controller. The goal of the simulation is to see how well the controller can keep the TCP position under various disturbance forces. To capture the response to different types of disturbance, we have chosen two representative disturbance signals. One is a step force signal, representing a sudden change of human force. The other is a slow changing force signal, generated by a hand motion and a hand model, representing a disturbance closer to real user cases. Because we are rendering a flat wall parallel with xy-plane, the disturbance is always along z axis towards the wall, imitating a push on the tool to press against the wall.

The wall we are rendering is located at z $= 0$, on which we have defined a grid of uniformly distributed query points (QPs). The x and y coordinate of the QPs range from -25 mm to 25 mm with a step of 5 mm, resulting in a grid of 121 QPs on the wall (Fig. 6). We query the performance of the controller at each QP by running the same simulation repetitively to have a performance distribution over the rendered wall. The simulation is conducted as follows: the TCP is first initialized at a QP, in contact with the virtual wall but with no interaction force, which is the boundary case for collision detection. Then, a disturbance signal is applied and the response of the system is recorded and later analyzed.

Scenario 1 tests the force rendering effect under the step force disturbance. The disturbance signal is shown below

$$F_1(t) = F_{sig1}(t) + \delta_{noise1}(t) \tag{13}$$

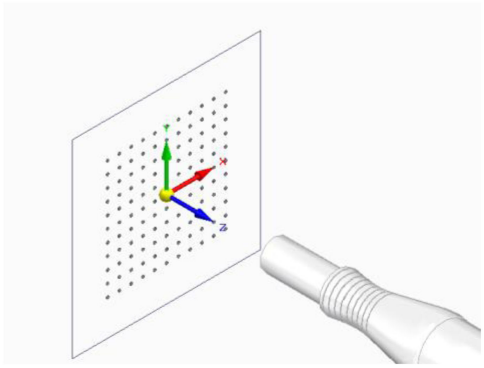

Fig. 6. Illustration of a grid of QPs in the workspace for general setup.

where

$$F_{sig1}(t) = F_{step}(t) = \begin{cases} 0, 0 \le t < 1 \\ F_c, 1 \le t \le 3 \end{cases}$$

$$\delta_{noise1}(t) = \begin{cases} 0, 0 \le t < 1 \\ N(0, 0.5), 1 \le t \le 3 \end{cases}$$

The force disturbance signal consists of two parts: a step signal and a white noise. For the step signal, there is a 1 s delay for the system to stabilize itself before the step happens at time point t $= 1$ s. The step level (F_c) is randomly generated uniformly within the range of 8 – 15 N for each run. The step signal holds for 2 s, until the end of the simulation. The total simulation time for a single run is 3 s. The disturbance is overlaid with a high frequency white noise signal. We use the notation $N(\mu, \sigma)$ to indicate that the noise follows normal distribution with a mean of μ and a standard deviation of σ. The amplitude of the white noise is roughly 10% of the step level. As stated, Monte-Carlo simulations are run at each QP with the random disturbance signal defined by (13) to check the robustness of the performance.

In Scenario 2, we disturb the system with a different type of signal and the rest of the simulation setup stays the same. Instead of having a step force disturbance, the main part of the disturbance signal is a motion signal generated by a user-hand model. The hand is modeled as a spring-damper system with stiffness (k_h) of 40 N/m and damping (b_h) of 3.6 Ns/m according to previous studies [34].

This model is introduced in the simulation by adding a spring-damper block between the TCP and the hand, shown in Fig. 7. Note here that we do not calculate F_h explicitly. The force is the result of the relative motion between the hand and TCP. That being said, the calculation of F_h should be

$$F_h = F_{spring} + F_{damper} = k_h(x_h - p_z) + b_h(\dot{x}_h - \dot{p}_z) \tag{14}$$

Then we define the motion of the hand x_h, which is determined by the final force level of the hand force F_c. Here, we reuse the step level F_c mentioned in Scenario 1 for the final force level because it is also randomly generated with the

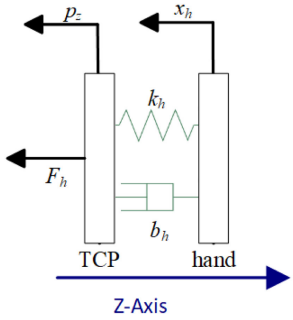

Fig. 7. Illustration of the hand model.

same range as before for the purpose of Monte-Carlo simulation. The motion of the hand is described by the following motion equations

$$a_h = \begin{cases} 0, 0 \le t < 1 \\ -a_c, 1 \le t < 1.45 \\ 0, 1.45 \le t < 1.55 \\ a_c, 1.55 \le t < 2 \\ 0, 2 \le t \le 3 \end{cases}$$
$$v_h = \int a_h dt, x_h = \int v_h dt$$

The length of acceleration/deceleration period is decided to ensure that the generated hand force changes smoothly. The calculation of these values follows the routine of trajectory planning where the assumption that the hand has a constant acceleration a_c during the time periods defined in a_h. The relative motion between the hand and the TCP induces a force acting on the TCP. This is what replaces the step force in the disturbance used in Scenario 2. It should be noted that the noise signal in Scenario 2 is only applied from 2 to 3 s, during which the hand frame is not moving anymore.

$$F_2(t) = F_{sig2}(t) + \delta_{noise2}(t) \tag{15}$$

where

$$F_{sig2}(t) = F_h$$
$$\delta_{noise2}(t) = \begin{cases} 0, 0 \le t < 2 \\ N(0, 0.5), 2 \le t \le 3 \end{cases}$$

A. Study of Responses

In order to develop some metrics to compare the response from the two controllers at different QPs, we performed some preliminary simulations. By examining results from these preliminary simulations, we realize that there are common characteristics in the responses of the controllers regardless of the choice of the QP. These characteristics will help us to better define the performance metrics.

1) Responses for Scenario 1: For the proposed controller, the TCP will go into the wall and come out in a very short time in Scenario 1, as shown in Fig. 8. Usually there is only one "dip"

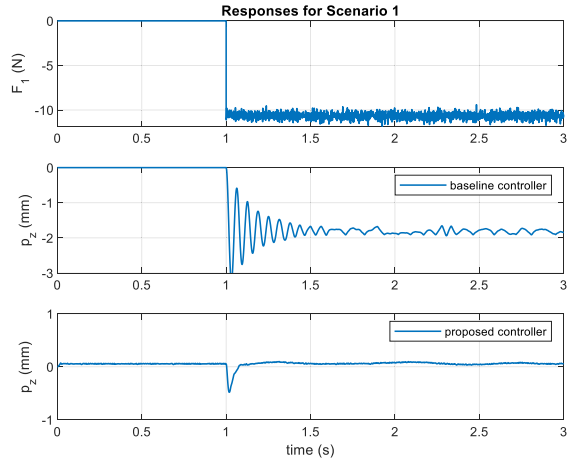

Fig. 8. The disturbance signal and the corresponding responses from the controllers for Scenario 1 at a random QP.

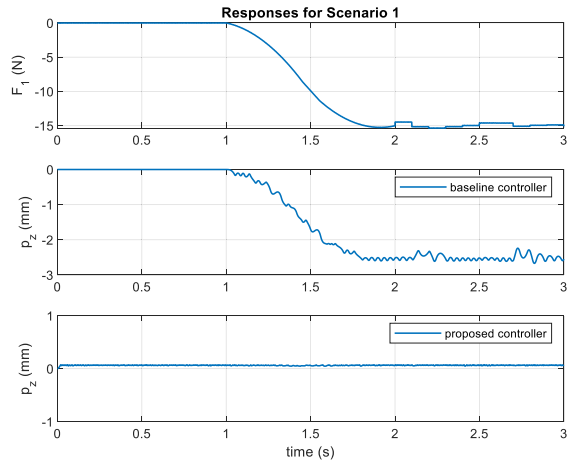

Fig. 9. The disturbance signal and the corresponding responses from the controllers for Scenario 2 at a random QP.

into the wall when the disturbance signal suddenly steps up and for the rest of the time it stays at the surface of the virtual wall.

For the baseline controller to maintain a stable response, we model the TCP with a damping of 4 Ns/m in z direction, based on the recommended stability guidelines found in [15] that

$$b_m > \frac{K_r T_s}{2} + |B_r| \tag{16}$$

where b_m is the mechanical damping of the system, K_r/B_r rendering stiffness/damping, and T_s sampling period. In theory, this should guarantee stability when rendering a pure spring wall ($B_r = 0$) with stiffness less than 8000 N/m. Consequently, we set the stiffness of the wall to be 6000 N/m for this controller and with the given damping, it shows an oscillation converging to a certain depth into the virtual wall.

2) Responses for Scenario 2: For the baseline controller, at the end of the Scenario 2 simulation, a penetration occurs because the controller treats the wall as a spring. This can be seen in Fig. 9. This time, the oscillation disappears because there is some damping in the user hand model.

For the proposed controller, the behavior is quite different. The TCP will almost stay at the surface of the virtual wall all the time unless there is a sudden huge change in the disturbance force.

B. Performance Metrics

Based on the characteristics found in these preliminary responses, we choose the following metrics to have a similar discussion across all controllers and scenarios, which enables better comparison on the performances of different controllers in different scenarios.

We consider two stages that can characterize the whole response of a controller. One is during the first impact and the other is when the system reaches steady state. The first shows the response to sudden changes and the second shows an overall performance. To better describe the performance in these stages, we define the following metrics.

First impact penetration (D_{z1}) is defined as the depth of the first penetration when a disturbance is applied. As we can see from Fig. 8, there is a penetration, probably the largest one throughout the whole simulation, corresponding to the sudden appearance of the disturbance force. Because the aim of our controller is to render a stiff wall, there should be no penetration into the wall. And because it shows the transient behavior, it can be the worst case the controller can do during the whole simulation.

First impact perceived stiffness (K_{z1}) is defined as the stiffness calculated by

$$K_{z1} = \frac{F_1(t_{z1})}{D_{z1}(t_{z1})}$$

where t_{z1} is the time point when D_{z1} is reached.

Steady-state penetration (D_{zs}): for both scenarios, there is a stage before the end that the disturbance force is not changing if the noise is ignored. This stage is when the steady state is considered to be reached. For Scenario 1, this is from 1 s to 3 s. For Scenario 2, this is from 2 s to 3 s. D_{zs} is defined as the mean penetration during this stage to average out the transient behaviors.

Steady-state perceived stiffness (K_{zs}) is defined as the stiffness calculated by

$$K_{zs} = \frac{F_c}{D_{zs}}$$

to represents the stiffness of the steady state.

All these metrics are not the final result to describe the performance of the controllers but rather to evaluate a single simulation (at one QP). Because we are running Monte-Carlo simulations for each QP with random disturbance force level for 20 simulations, the final result to describe the performance on a controller level is the mean and standard deviation of these metrics. Then, the distribution of the final result over the wall (for all QPs) will demonstrate how the performance vary on the wall.

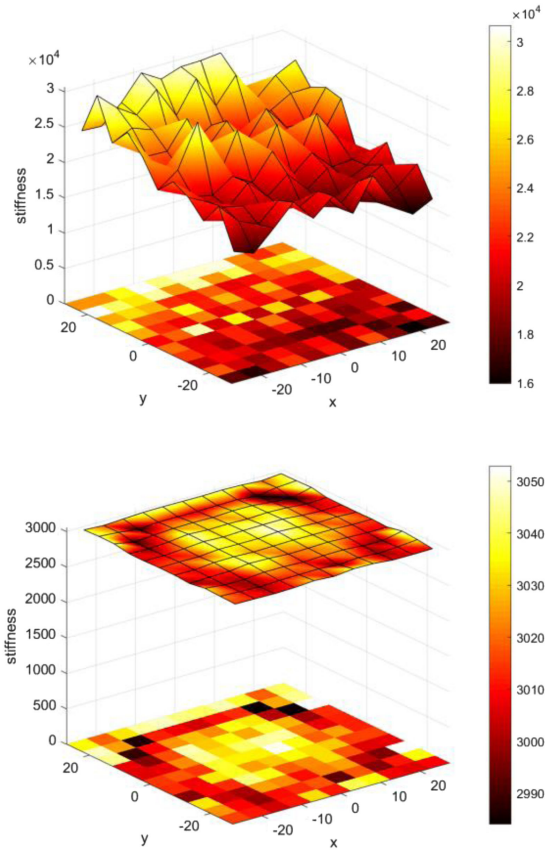

Fig. 10. The mean of K_{z1} over 20 simulations at each QP for Scenario 1 distributed over the wall (top: proposed, bottom: baseline).

VI. Simulation Results

As discussed in the previous section, instead of having one number to describe the performance of the controller, we would like to first show distribution plots over the rendered wall for the performance metrics defined in Section V-B.

A. Scenario 1 with Force Model

With Scenario 1, the results demonstrate the response under sudden quick disturbance changes, similar to the purpose of doing a step test.

Fig. 10 shows the distribution of K_{z1} over the wall for both controllers. For our proposed controller on the top, the stiffness rendered is not uniformly distributed over the wall. Along y axis, it seems to have a tendency of increasing the stiffness when the y coordinate increases but there is no such noticeable trend along x axis. This can be the result of the TAU structure having some symmetry w.r.t x axis but not so much w.r.t y axis. To look at the result in another direction, this can be related to the saturation of joint 6, which is 2 Nm. The angle of joint 6 is denoted q_6 in Fig. 1, providing an up-down motion for Chain 3. In the previous paper [27], the researchers found that joint 6 can have a different behavior due to the possibility that most of the gravity of TAU is compensated by it, making it work closer to the saturation limit. To illustrate this possibility, Fig. 11 shows the joint torque on joint 6 for two different QPs. We use QP(x, y) to represent a QP on the wall with its xy

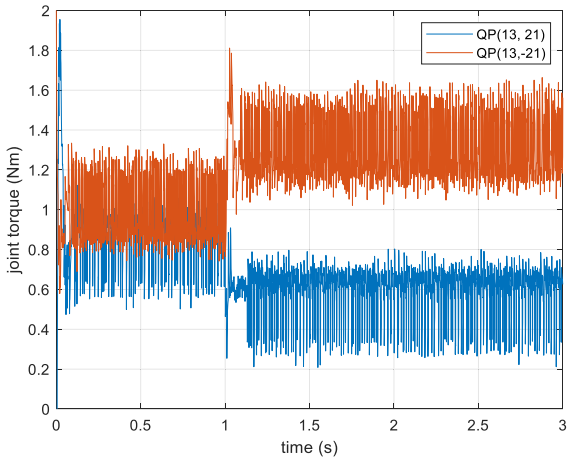

Fig. 11. The joint torque on joint 6 for different QPs.

coordinate. It is shown that even with the same level of disturbance (13 N in Fig. 11), joint 6 produces more torque to compensate when the QP is closer to the bottom of the workspace (smaller y coordinate). It is also verified that with higher F_c and larger difference in y coordinate of QP, this effect is more obvious.

As for the baseline controller, the result is fairly consistent although the rendered stiffness (around 3000 N/m) is much lower than the desired stiffness (6000 N/m). The achieved stiffness only reaches around half of the desired result because this stiffness calculation focus on the first impact which goes into the wall a bit more than it should. It is rather flat on the xy-plane compared to the figure of the proposed controller, because the motor torques are probably well within the saturation limit of the motors.

Note that the two distribution plots in Fig. 10 are not in the same scale because the calculated stiffness are not of the same magnitude. If we take a closer look at Fig. 8 for typical responses, we can see that D_{z1} for the proposed controller is much smaller than that for the baseline controller. This is why the stiffness calculated in this way differs with a magnitude of 10.

The distribution plots of K_{zs} (Fig. 12) show a similar trend, though with a higher value. This is true especially for the baseline controller as it renders a stiffness value around the desired 6000 N/m. The reason is that, when taking the average, the oscillation is cancelled out, which leads to D_{zs} smaller than D_{z1}. The result for the proposed controller is even higher and, in some sense, can be considered as approaching infinity because D_{zs} is close to zero. This also explains why the stiffness values at different QPs vary quite a lot.

B. Scenario 2 with Hand Model

With Scenario 2, the idea is to simulate the interaction between the device and the user hand, so as to test the performance of the controllers.

Compared to Scenario 1, the change of force disturbance in Scenario 2 is continuous and slow. This is closer to real interaction in the physical world and this should be an easier case for both controllers.

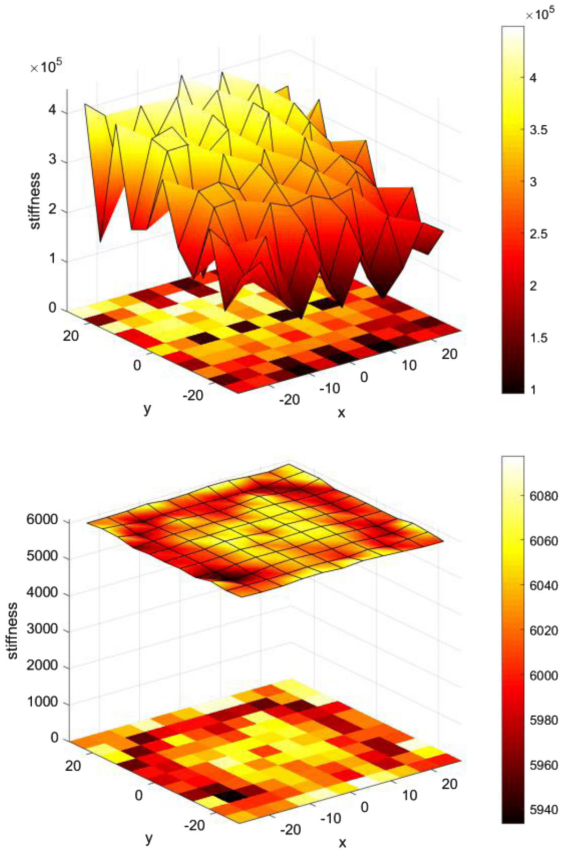

Fig. 12. The mean of K_{zs} over 20 simulations at each QP for Scenario 1 distributed over the wall (top: proposed, bottom: baseline).

Due to the nature of Scenario 2, K_{z1} is not that relevant as the impact is rather "soft". If we compare Fig. 8 and Fig. 9, it can be hard to find a first impact/dip. If we calculate K_{z1}, we can get a number but it does not show us more insight on how the controller performs because the first impact does not exist. So for Scenario 2, we focus on K_{zs}, as shown in Fig. 13.

For the baseline controller, the result is well-aligned with that for Scenario 1 although the stiffness value can be slightly lower. This small decrease can be the result of the high frequency small oscillations presented in Fig. 9. The result is still acceptable because the mean stiffness over the wall is about 6000 N/m with a standard deviation of 75 N/m.

For the proposed controller, it is again another magnitude higher, making it closer to the goal of rendering infinite stiffness. It is not so clear from the figure but even the minimal K_{zs} over the whole wall is still around 55000 N/m.

VII. Co-Simulation Verification

This section further verifies the performance of the proposed controller through a MATLAB-ADAMS co-simulation (CoS) setup to achieve more realistic simulations with a more detailed device model.

The CoS combines the best parts of both software. MATLAB/Simulink offers a simple way to build a controller while ADAMS is well-known for building and solving simulations for multi-body dynamics. How the two pieces of software

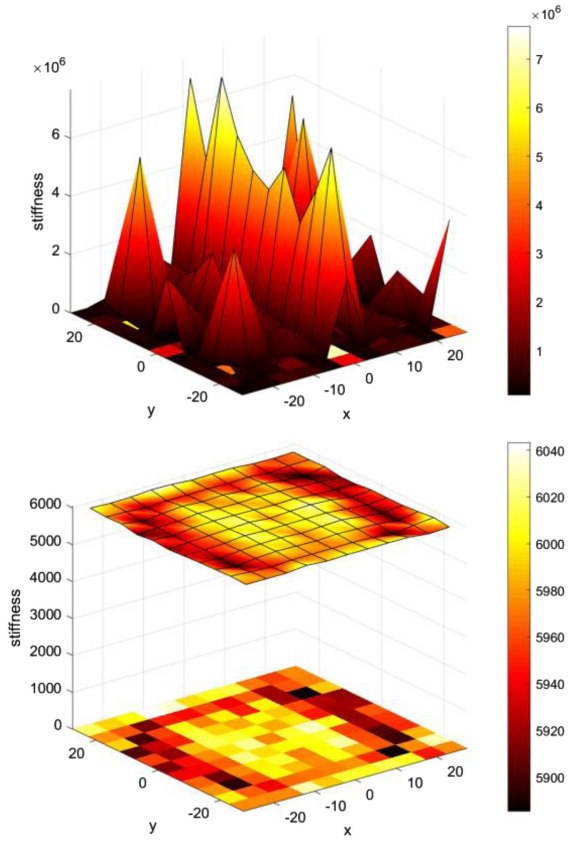

Fig. 13. The mean of K_{zs} over 20 simulations at each QP for Scenario 2 distributed over the wall (top: proposed, bottom: baseline).

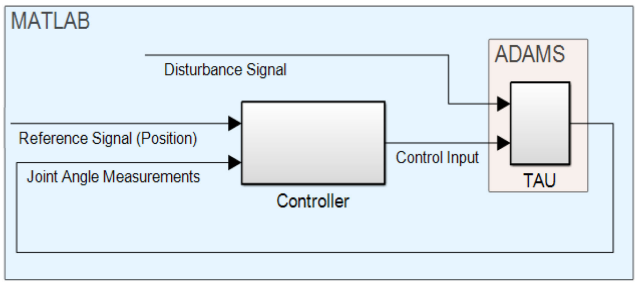

Fig. 14. The framework for MATLAB-ADAMS co-simulation.

interface with each other is demonstrated in Fig. 14. The model of TAU is modeled in ADAMS and exported to MATLAB as an S-function. This S-function only provides the interface for the 2 pieces of software to communicate with each other so that solving dynamic equations at each time step happens in ADAMS. The controller and the disturbance model are implemented in MATLAB. The control input from the controller and the disturbance force are input to ADAMS. The joint angle measurements from ADAMS are fed back to the controller in MATLAB to close the loop. Another motivation for CoS is that it can demonstrate some challenges when we turn from simulation to real experiments because the ADAMS model is more accurate w.r.t the real device. The TAU device model for Simscape-based simulation is only accurate in kinematics but the mass and inertia distribution of

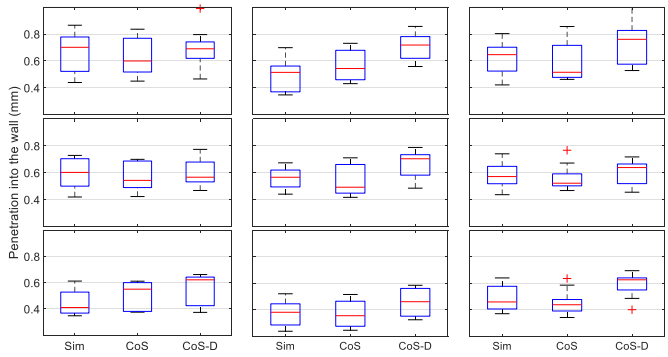

Fig. 15. Box plots of penetration into the wall at 9 QPs under force model disturbance for Sim, CoS, and CoS-D.

the device is simplified to simple geometry and defined using lumped models. The model in ADAMS, on the other hand, is built based on the CAD models used in manufacturing the real device.

The setup for CoS follows the methodology in Section V. We only shrink the size of the simulation to have 9 (3-by-3) QPs on the wall and the simulation is repeated 10 times instead of 20. The total number of the co-simulations is 90. The current implementation for CoS is quite time-consuming as one single 3-second simulation will take about 40 minutes to finish on a modern desktop computer.

Apart from that, we introduce a random discrete time delay between the controller and the ADAMS model and run the CoS again. The goal is to see whether our controller can deal with some possible delays due to computation and communication. The random delay is an integer times of sampling time, uniformly distributed from 0 to 3.

A. *Control Requirements Verification*

Given the CoS results, whether the performance still meets the control requirements as listed in Table II is the first thing to verify. The most important one in the table is that the TCP position deviation in z axis direction, in other words, the penetration, should be less than 1 mm. From this point on, we refer to the Simscape simulation as Sim, co-simulation CoS, and co-simulation with random delay CoS-D. The largest penetration achieved by our proposed controller during Sim, CoS, and CoS-D for the force model disturbance is shown in Fig. 15. The performances from Sim and CoS look similar and are verified by t-test (0.05 as significant level) with the smallest p-value larger than 0.1. The penetration for CoS-D looks comparable with that from CoS or slightly worse, which shows that random delay does have a negative effect on the performance and our controller can handle it to some extent. The results for the hand model disturbance look similar to Fig. 15 and the overall largest penetrations for both scenarios are well within 1 mm when there is no random delay in the system, which meets the control requirement. For CoS-D, only 3 out of 90 results show penetration larger than 1 mm but they are less than 1.1 mm. More detailed values can be found in Table III and the CoS-D numbers are shown in the brackets

TABLE III
LARGEST PENETRATION ACROSS ALL 9 QPS

Results	Largest Penetration (mm)
Sim with force model	0.87
CoS with force model	0.86 (1.09)
Sim with hand model	0.18
CoS with hand model	0.17 (0.23)

TABLE IV
EXTREME VALUES FOR METRICS FROM SIM AND COS

Results	Values (N/m)	
	Sim	CoS
Scenario 1 Lowest K_{z1} for p.c.	17174	16743 (13367)
Scenario 1 Higheset K_{z1} for b.c.	3112.6	5360.2
Scenario 1 Lowest K_{zs} for p.c.	92387	34306 (54634)
Scenario 1 Higheset K_{zs} for b.c.	6180.5	6302.2
Scenario 2 Lowest K_{zs} for p.c.	65002	22463 (20299)
Scenario 2 Higheset K_{zs} for b.c.	6202.1	6342.4

as a special case for CoS. Note that only the data for the corresponding 9 QPs from the Sim data are extracted for comparison with the CoS and CoS-D results.

B. Performance Verification

Now that the controller meets the control requirement even when applied to the ADAMS model, we further verify that our proposed controller still works better than the baseline controller. The metrics introduced in Section V-B are used here again. The result for CoS shows a variation from that for Sim but our proposed controller out-performs the baseline controller by a significant amount. Table IV shows that even the lowest value for given metrics across all QPs achieved by the proposed controller (p.c.) is significantly higher than the highest value achieved by the baseline controller (b.c.) regardless of the scenario. Again, the CoS-D results are shown in brackets behind the corresponding CoS tests. However, we should clarify here that high values might be sensitive to small changes in penetration so they can get quite noisy and vary a lot. This effect is visible in figures like Fig. 10.

VIII. DISCUSSION

For the approach proposed in this paper for haptic rendering, we have chosen to render the contact with a stiff wall as a first initial test case. This wall is positioned at $z = 0$ and parallel to the xy-plane and enables us to study the performance in solely z axis direction.

Our proposed controller is compared against a baseline controller which models the stiff wall as a spring-damper system [33]. The goal of our controller is to render an infinite-stiffness wall. This cannot be reached in real experiments so the result of trying to reach this goal becomes render a wall as stiff as the saturation limits allow. Most existing controllers are designed to render a pre-defined stiffness. Due to this difference, the baseline controller can never render an infinite stiffness but only what it is designed for. In addition, there is also an upper bound of stiffness that can be successfully rendered for a given device, which is governed by stability and saturation. In this paper, we have designed the baseline controller for TAU with the goal of rendering a stiffness of 6000 N/m fulfilling requirements on stability and time response. It should be noted that increasing the mechanical damping of the system would improve this number but it is out of the scope of this paper. Our focus is to design a controller with a given system while the mechanical damping is a system property instead of a control parameter that we can tune.

The control ideas behind the controllers are also different. Instead of treating a stiff wall as a spring-damper system and rendering by matching the desired interaction force, we propose a position controller that prevents positional deviation (penetration) into the wall. The force reference is computed as a result of achieving this. This change in controller seems to enable more stable interactions when rendering a high stiffness wall even when mechanical damping is low in the system. This is verified by the simulation as there is no mechanical damping in the model of TAU and it is still possible to render high stiffness with stability when using the proposed controller. This also makes sense because the stability of the position is achieved by position feedback instead of mechanical damping in the system to dissipate the extra energy. Similar findings can also be seen in [33], [35] that the highest stiffness their proposed controller can stably achieve is higher than the upper bound calculated from the guidelines (16) in papers like [14].

Apart from the controller design, this study is carried out on a device other than the well-studied ones like Phantom [33], or even KUKA [35]. The device is a desktop 6-DOF parallel haptic device called TAU [25] previously developed at KTH that has novel asymmetric mechanical structure, rarely studied in the literature. Our approach avoids the complicated task of deriving the dynamic model from mechanics principles and the obtained linear model is efficient for real-time control. Although the proposed approach is only applied to this specific TAU device, there is no obvious limitations for the approach to be generalized to other PMs, or even serial ones.

Compared with other 1-DOF haptic rendering studies, we aim to get a general understanding of the performance over the whole workspace instead of alone one specific line. In this initial study, we have performed 121 (QPs) $\times$ 20 simulations to evaluate the performance in 1D by interacting with a stiff wall.

To verify the results from Sim, we carry out CoS as an alternative to real experiment. With different simulation environment and device model, we are happy to see that the performance still meets the control requirements. Although the results from CoS are not consistent with that from Sim, it is reasonable and the advantage of the proposed controller over the baseline controller is preserved quite well. With a random delay introduced into the system, the performance of the co-simulation should be one step closer to the real experiment. Although we did not specifically

design a dedicated module to handle delay, the results show that the proposed controller can still meet the control requirement most of the time and maintain a similar performance level w.r.t the results without delay.

IX. Conclusions and Future Work

In this paper, we have presented a new approach to solve the problem of rendering stiff interaction as a position controller design problem. This allows us to apply mature position controller design methodologies to solve this haptic rendering problem. With rendering a stiff wall as our initial case and control objective, the design focus shifts from position accuracy to disturbance rejection to better serve the new control task. A Monte-Carlo simulation is conducted to test the robust performance of the proposed controller and a CoS is followed to provide some verification of the previous simulation results. We can conclude from these initial investigations that the proposed method shows acceptable performance in simulation which indicates that it can be an alternative method for haptic rendering. For the selected test scenario of rendering a stiff wall, the proposed controller can render a stiffness at the level of 10^4N/m. As demonstrated, this result should be quite robust to disturbance force, human user behavior, and time delay in the system. Due to the position feedback in the position controller, the mechanical damping of the system is not a huge factor of limiting the maximal stably rendered stiffness of a given system, which opens new possibilities for trade-offs between stability and transparency of a haptic device.

It should be noted that our proposed controller is only for stiff interaction at the moment and it can be used as an alternative controller for that. As for other problems in haptic rendering like collision detection, we consider those out of the scope of our study and solved well by other parts of the overall haptic rendering algorithm. It should also be quite easy for our proposed controller to work with other pre-existing rendering approaches as our controller does not require additional sensors and the interface with the whole structure is similar as Fig. 2 implies.

In the future, we want to expand this approach and explore the rendering of more complex virtual objects. We also plan to verify our findings in this paper in real experiments on the actual haptic device.

Acknowledgment

Yang Wang would like to thank Chinese Scholarship Council (CSC) for supporting his Ph. D. study. Lei Feng is financially supported by KTH XPRES.

References

[1] "Haptics - user interaction - iOS - human interface guidelines - apple developer." Accessed on: Dec. 8, 2019. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/haptics/

[2] "Mouse and trackpad - user interaction - macos - human interface guidelines - apple developer." Accessed on: Dec. 8, 2019. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/macos/user-interaction/mouse-and-trackpad/

[3] A. L. Guinan, N. A. Caswell, F. A. Drews, and W. R. Provancher, "A video game controller with skin stretch haptic feedback," in *Proc. IEEE Int. Conf. Consum. Electron.*, Berlin, Jan. 2013, pp. 456–457, doi: 10.1109/ICCE.2013.6486973.

[4] L. Kim, W. Park, H. Cho, and S. Park, "A universal remote control with haptic interface for customer electronic devices," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 913–918, May 2010.

[5] "3D Systems phantom premium," 3D Systems. Accessed on: May 3, 2019. [Online]. Available: https://www.3dsystems.com/haptics-devices/3d-systems-phantom-premium

[6] R. Q. van der Linde and P. Lammertse, "HapticMaster – a generic force controlled robot for human interaction," *Ind. Robot Int. J.*, vol. 30, no. 6, pp. 515–524, Jan. 2003.

[7] W. McMahan and K. J. Kuchenbecker, "Haptic display of realistic tool contact via dynamically compensated control of a dedicated actuator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, Oct. 2009, pp. 3170–3177.

[8] "Simodont dental trainer - home," Simodont. Accessed on: Dec. 8, 2019. [Online]. Available: https://www.simodontdentaltrainer.com/

[9] "DentSim is the industry leader in augmented reality dental simulation helping dental students around the world improve their skills.," Image Navigation. Accessed on: Apr. 24, 2019. [Online]. Available: https://image-navigation.com/home-page/dentsim/

[10] "Voxel-Man dental." Accessed on: Apr. 24, 2019. [Online]. Available: https://www.voxel-man.com/simulators/dental/

[11] J. K. Salisbury, "Design of haptic interfaces," *J. Acoust. Soc. Am.*, vol. 92, no. 4, pp. 2438–2438, Oct. 1992.

[12] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: Introductory concepts," *IEEE Comput. Graph. Appl.*, vol. 24, no. 2, pp. 24–32, Mar.-Apri. 2004.

[13] R. B. Gillespie and M. R. Cutkosky, "Stable user-specific haptic rendering of the virtual wall," in *Proc. ASME Dyn. Syst. Control Div.*, pp. 397–406, vol. 58, 1996.

[14] J. J. Abbott and A. M. Okamura, "Effects of position quantization and sampling rate on virtual-wall passivity," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 952–964, Oct. 2005.

[15] J. E. Colgate, M. C. Stanley, and J. M. Brown, "Issues in the haptic display of tool use," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, pp. 140–145, 3–Feb, 1995.

[16] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 465–474, Oct. 1999.

[17] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 1–10, Feb. 2002.

[18] J. Ryu and M. Yoon, "Memory-based passivation approach for stable haptic interaction," *IEEEASME Trans. Mechatron.*, vol. 19, no. 4, pp. 1424–1435, Aug. 2014.

[19] R. Oftadeh, M. M. Aref, and H. D. Taghirad, "Explicit dynamics formulation of stewart-gough platform: A newton-euler approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2772–2777.

[20] E. Abedloo, A. Molaei, and H. D. Taghirad, "Closed-form dynamic formulation of spherical parallel manipulators by gibbs-appell method," in *Proc. 2nd RSI/ISM Int. Conf. Robot. Mechatronics (ICRoM)*, Oct. 2014, pp. 576–581.

[21] H. Cheng, Y. K. Yiu, and Z. Li, "Dynamics and control of redundantly actuated parallel manipulators," *IEEEASME Trans. Mechatron.*, vol. 8, no. 4, pp. 483–491, Dec. 2003.

[22] D. Corbel, M. Gouttefarde, O. Company, and F. Pierrot, "Towards 100G with PKM. Is actuation redundancy a good solution for pick-and-place?," in *Proc. IEEE Int. Conf. Robot. Automat.*, no. section IV, May 2010, pp. 4675–4682.

[23] H. Abdellatif and B. Heimann, "Advanced model-based control of a 6-DOF hexapod robot: A case study," *IEEEASME Trans. Mechatron.*, vol. 15, no. 2, pp. 269–279, Apr. 2010.

[24] M. Díaz-Rodríguez, A. Valera, V. Mata, and M. Valles, "Model-based control of a 3-DOF parallel robot based on identified relevant parameters," *IEEEASME Trans. Mechatron.*, vol. 18, no. 6, pp. 1737–1744, Dec. 2013.

[25] A. Ahmad, S. Khan, and K. Anderson, "Kinematics and dynamics of a novel 6-DoF TAU haptic device," in *Poc. IEEE Int. Conf. Mechatronics*, Apr. 2011, pp. 719–724.

[26] A. Ahmad, K. Andersson, and U. Sellgren, "An optimization approach toward a robust design of six degrees of freedom haptic devices," *J. Mech. Des.*, vol. 137, no. 4, Apr. 2015, Art. no. 042301-14.

[27] Y. Wang, L. Feng, and K. Andersson, "A new approach to haptic rendering by position control," in *Proc. IEEE 15th Int. Conf. Control Automat.*, Edinburgh, Jul. 2019, pp. 148–153.

[28] S. J. Lederman and R. L. Klatzky, "Haptic perception: A tutorial," *Atten. Percept. Psychophys.*, vol. 71, no. 7, pp. 1439–1459, Oct. 2009.
[29] A. J. Doxon, D. E. Johnson, H. Z. Tan, and W. R. Provancher, "Human detection and discrimination of tactile repeatability, mechanical backlash, and temporal delay in a combined tactile-kinesthetic haptic display system," *IEEE Trans. Haptics*, vol. 6, no. 4, pp. 453–463, Oct. 2013.
[30] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed.. New York, NY, USA: Dover Publications, 2011.
[31] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, "Realistic haptic rendering of interacting deformable objects in virtual environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 1, pp. 36–47, Jan. 2006.
[32] C. R. Carignan and K. R. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *Haptics-e, Electron. J. Haptics Res.*, vol. 2, no. 2, pp. 1–14, Feb. 2000, [Online]. Available: http://www.haptics-e.org
[33] I. Desai, A. Gupta, and D. Chakraborty, "Rendering stiff virtual walls using model matching based haptic controller," *IEEE Trans. Haptics*, vol. 12, no. 2, pp. 166–178, Apr. 2019.
[34] J. E. Speich, L. Shao, and M. Goldfarb, "Modeling the human hand as it interacts with a telemanipulation system," *Mechatronics*, vol. 15, no. 9, pp. 1127–1142, Nov. 2005.
[35] H. Singh, D. Janetzko, A. Jafari, B. Weber, C.-I. Lee, and J.-H. Ryu, "Enhancing the rate-hardness of haptic interaction: Successive force augmentation approach," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 809–819, Jan. 2020.

Yang Wang (Student Member, IEEE) received the B.S. degree in mechanical engineering and automation from Beijing Jiaotong University, Beijing, China in 2014 and the M.Sc. in Eng. degree in robotics from Johns Hopkins University, Baltimore, MD, USA, in 2016. He is currently working toward the *Ph.D.* degree in machine design with the KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include haptic robotics, medical robotics, mechanical design, and dynamics and control.

Lei Feng (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronics from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2001, respectively, and the *Ph.D.* degree in electrical and computer engineering from the Systems Control Group, University of Toronto, Toronto, ON, Canada, in 2007. Since 2012, he has been an Associate Professor with the Mechatronics and Embedded Control System Division, KTH Royal Institute of Technology, Stockholm, Sweden. His main research interests include novel mechatronic systems enabled by additive manufacturing, energy management control of mechatronic systems, autonomous driving, verification and control synthesis of cyber-physical systems, and supervisory control of discrete-event systems.

Kjell Andersson is currently a Full Professor in machine design with the Department of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden. He has more than 20 years of experience in research in modeling, simulation, and optimization of dynamical behavior of mechanical systems. His previous research involves a variety of application areas such as haptic devices, forestry machines, wheel loaders, and power take-off transmissions for wave energy converters.