A Joint-space Position Control-based Approach to Haptic Rendering of Stiff Objects using Gain Scheduling

Yang Wang¹ · Lei Feng¹ · Kjell Andersson¹

Received: 4 May 2021 / Accepted: 8 October 2021 / Published online: 26 October 2021
© The Author(s) 2021

Abstract

Haptic rendering often deals with interactions between stiff objects. A traditional way of force computing models the interaction using a spring-damper system, which suffers from stability issues when the desired stiffness is high. Instead of computing a force, this paper continues to explore shifting the focus to rendering an interaction with no penetration, which can be accomplished by using a position controller in the joint space using the encoders as feedback directly. In order to make this approach easily adaptable to any device, an alternative way to model the dynamics of the device is also presented, which is to linearize a detailed simulation model. As a family of linearized models is used to approximate the full dynamic model of the system, it is important to have a smooth transition between multiple sets of controller gains generated based on these models. Gain scheduling is introduced to improve the performance in certain areas and a comparison among three controllers is conducted in a simulation setup.

Keywords Haptic rendering · Gain scheduling

1 Introduction

Haptic devices are mechanisms that try to reconstruct the often missing haptic feedback in virtual reality (VR) interactions and reflect it to the user to make the whole interaction more realistic. Haptic feedback, though sounded unfamiliar, is one of the five main senses we rely on daily. Haptic technologies are everywhere from our mobile phones and laptops to highly specialized devices like dental training simulators [16, 32].

Analogous to graphic/visual rendering, haptic rendering is an additional human-computer interface for the sense of touch so that the user can perceive virtual objects through haptic channels like skin, hands, and other receptors. In this paper, we focus more on the haptic rendering of interactions between a tool and a stiff object using desktop impedance-type force feedback devices. For such applications, it is commonly accepted that the refresh rate of the haptic rendering loop should be 1000 Hz and within each iteration there are 5 sequential steps [26]:

1. sense the configuration of the device in joint space;
2. map that into workspace using forward kinematics;
3. determine whether there is a collision and where it is;
4. compute the correct interaction force/torque (F/T) to render in workspace;
5. map that into joint space F/T as reference signals for low level control;

Although interaction with soft objects can be one of the new frontiers of haptic research, stiff interaction is still a large part of the targeted rendering scenarios and quite challenging to render stably. Keeping a high refresh rate for the haptic rendering loop is one of the ways towards more stable rendering, which can be achieved by making computation more efficient. Different approaches are developed for different modeling methods. Zilles

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

✉ Yang Wang
wang5@kth.se

Lei Feng
lfeng@kth.se

Kjell Andersson
kan@kth.se

¹ Machine Design, Brinellvägen 83, KTH-Royal Institute of Technology, SE-100 44, Stockholm, Sweden

and Salisbury [36] develop the concept of god-object to determine where the interaction is. A similar concept is called “virtual proxy” [24]. This distinguishes where the tool should be, which is represented by the god-object, from where the tool actually is. The position of the god-object is solved by active constraints and it is suitable for the systems that use polyhedral representations. Ortega et al. [21] later extend the concept into a 6-DOF god-object for more complex haptic applications. As the computation gets more complex, they use asynchronous processes to compute the god-object motion and the rendering force, respectively, to meet the time requirement for each process. Otaduy and Lin [22] give another take at rendering complex objects with high polygon count. They linearize the contact model in complex contact scenarios to maximize the haptic refresh rate. Yu et al. [35] propose an extended version of the configuration-based optimization approach [31] to target sharp geometric features on objects. The approach uses a coarse-to-fine sphere tree to represent an object for faster collision detection and contact constraint formulation. It is designed for dental simulators and claimed better for rendering narrow space constraints and fine features in general.

Normally, for the whole haptic rendering loop to work, models of the device kinematics and dynamics are evaluated in real-time, which is another component to slow down the haptic rendering loop. Shortening this computation time is beneficial for stability and the methods are not limited within haptic applications. Park and Lee [23] introduce a simplified model to estimate the full model. To compensate the nonlinear dynamics and modeling errors, adaptive and robust control method are used to dynamically estimate and compensate these terms. Cheng et al. [6] experiment on a 2-DOF redundantly actuated planar parallel mechanism (PM) to compare the performance of 4 different control schemes. The device is also chosen for its relatively simple structure and the model of its dynamics can be obtained in closed-form. The comparison shows that the proposed control algorithm has smaller error in position/velocity tracking by moving a part of the computation offline to have a shorter execution time. Corbel et al. [11] propose a simplified dynamic model that is motivated by the high speed high acceleration application. The simplification is to ignore the inertia of some linkages and split their mass so that the calculation associated with these linkages can be avoided. Abdellatif et al. [2] propose a method to find a reduced set of dynamic parameters and use that to reduce computation. 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. [12] propose an model identification method and demonstrate it on a 3-DOF PM. The dynamic model is first established with base parameters and then the parameter with the largest standard

deviation is iteratively removed until the inertial matrix is positive definite. The results show that the controller based on the reduced model has less error and computational load, which enables the real-time control of the device.

Another instability factor is extra energy. Though the virtual environment is passive, extra energy can still be generated from interacting due to the sampling nature of the haptic device. Gillespie and Cutkosky [15] term the phenomenon “energy leak” and they propose passivity-based approaches to solve this. Hannaford and Ryu [17] propose time-domain passivity control with an observer-controller structure. An adaptive damping is introduced to precisely dissipate the extra energy. Ryu and Yoon [25] extend this into a memory-based passivity approach. In their case, a dedicated hardware FPGA is fast enough to log all the data needed for position versus force trajectory plot and the controller uses that to guarantee that the net energy coming from the passive virtual environment is less than 0. Note that most passivity-based controllers are considered a bit conservative as passivity is a sufficient condition for stability. Colgate et al. [10] propose “virtual coupling” as a filter to achieve stable behavior. This approach is quite popular though some studies also point out that it might lead to some “force artifacts” in force rendering [35]. Adams and Hannaford [3] later extend this approach to work for both impedance or admittance models of haptic interaction but the approach cannot handle a dynamically changing environment.

In summary, there are different ways to speed up the haptic rendering loop and cut down computations. For step 4 of force computing, stiff objects are mostly modeled as a spring-damper system and this can lead to unstable behavior if not dealt with carefully. To avoid the shortcomings of using a spring-damper system, we propose a position-control based approach as a solution to achieve a stable interaction with stiff objects [33]. A new user scenario is introduced in this paper. In this scenario, we move the tool along a straight path on a stiff flat wall while we are applying a force perpendicular to the wall. This “pressing while sliding” scenario resembles the task of polishing in dental procedures. The following research questions (RQs) will be further explored:

RQ 1: What are the benefits of using a family of piecewise linear models to develop a gain scheduling controller with smooth transitions?

RQ 2: What performance can be obtained for the user scenario “pressing while sliding”?

The rest of this paper is structured as follows: the concept of the approach is revisited in Section 2. In Section 3, the control design process is laid out with the use of piecewise linear model and gain scheduling. Section 4 shows the simulation methodology on how to setup tests to evaluate the system. Results of the simulation are shown in Section 5,

Table 1 Design parameters of TAU

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 I-column, $3L_3$	274.5 mm

followed by a discussion in Section 6. Finally, conclusions and future work are presented in Section 7.

2 An Approach using Position Controller

In this section, we first introduce the test case device for the paper and then, the proposed approach is recapped as it originates from previous work [33, 34].

2.1 Description of Test Case

The test case for this paper is TAU, a desktop 6-DOF PM haptic device developed at KTH as a dental training simulator I [4, 5, 29]. The design of the device is optimized in the previous work [4] to have a large enough singularity-free workspace (a $50 \times 50 \times 50 \text{ mm}^3$ cube) while maintaining a uniform performance within the whole workspace and a relatively small footprint. The whole system can be fit into a $250 \times 250 \times 300 \text{ mm}^3$ box. The design parameters can be found in Table 1 and other details can be seen in Fig. 1.

What makes TAU different from other well-studied PMs such as Delta [7] or Stewart [20] is that it has asymmetry in its kinematic structure. This is one of the reasons why we choose TAU as the test device and we will further explain this asymmetry and the challenges it opposes in Section 3.2

2.2 Recap of the Approach

For an impedance-type haptic device to render a stiff object, the traditional way is to model the virtual environment (VE) as a linear spring with high stiffness. With a high enough stiffness, the rendered spring behaves like a stiff object in real life because of the imperceptible deformation. When there is a collision and a penetration into the object, a penalty force following the Hooke's Law will push the device back out of the object. This is one of the common ways to do step 4 in the haptic rendering loop since the idea is introduced in [9], illustrated in Fig. 2a. It computes reference force signals in the Cartesian space and further converts them to joint torques signals through inverse dynamics. Here, we use the naming convention from [31],

Fig. 1 Kinematics schematics for TAU with design parameters from Table 1

where we call the actual pose of the tool that is tracked by the encoders the haptic tool and the pose that satisfies no penetration the graphic tool. The haptic tool and the graphic tool are colored green and blue, respectively, in Fig. 2.

Our approach introduces a new way to compute the rendering force [34]. A stiff object behaves similarly to a position controller in the sense that it should keep the tool on the surface of the object with no penetration. In this way, given the collision position and the current position of the tool, a position controller can compute a penalty force to eliminate the difference between the two, bringing the tool back to the surface of the object. It is a closed-loop position control in joint space.

Compared to the traditional force rendering method, the newly proposed controller has the same input/output (I/O) signals as other controllers; hence the traditional one can be easily replaced by the new controller in haptic devices, which makes it easy to use the new controller in an old system. Because now the control goal is in joint space instead of the Cartesian space, some nonlinear and complex computation is avoided, which is good for maintaining the high haptic rendering loop rate.

Fig. 2 Illustration of different control structures and model of objects: (a) traditional and (b) proposed. The blue ball symbolize the graphic tool while the green ball the haptic tool

3 Control Design

As recapped in Section 2.2, haptic rendering problem can be defined as generating the correct interaction force to create the illusion of directly interacting with virtual objects [19]. To solve the haptic rendering problem as a position control problem, the problem is first reformulated in Section 3.1. Then, for the remaining part of Section 3, we revisit the plant modeling and the discrete time control design. A linear-quadratic-regulator-based (LQR-based) full-state feedback structure is chosen because it can be optimized to prevent large control input. Given this choice, a linear time-invariant (LTI) state-space control plant is sufficient for the control design. In this paper, only key results are highlighted. A detailed version of this can be found in [34].

3.1 Equivalent Problem

With the discussion from the previous section, the haptic rendering of stiff interaction can be realized this way: given the current pose of tool handle in the Cartesian space, design a position controller that computes a motor torque reference so that the tool has minimal penetration into the stiff object when it is disturbed by the user. In this way, the user will feel like interacting with a real stiff object. Without losing generality or consistency with previous work [5], the pose in Cartesian space can be represented by a 6-by-1 vector

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

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 pose in Cartesian space corresponds to joint angles of 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

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

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. Having passive joints is also one of the reasons why the forward kinematics of a PM is hard to derive.

Accordingly, the force to be rendered here is a 6-DOF force acting on the tool in Cartesian space, defined by a 6-by-1 vector

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

and the joint torque to achieve that is defined as

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

where τ_i denotes the joint torque on joint i .

These are general definitions in kinematics and dynamics, and they can be used on both the graphic tool and the haptic tool. With these definitions, the problem becomes: given the current pose of the graphic tool $\mathbf{X}_{graphic}$ determined by collision detection as reference signal and the current pose of the haptic tool $\mathbf{X}_{haptic}$ estimated based on the $\mathbf{q}_{haptic}$ measured by joint encoders, design a position controller that determines the joint torques $\boldsymbol{\tau}$ to minimized the difference between $\mathbf{X}_{haptic}$ and $\mathbf{X}_{graphic}$.

3.2 Plant Modeling

A model of the device dynamics is needed for this phase and traditionally, the analytical form of it is obtained through complex mathematical derivation. This is non-trivial, especially in our case, as the device is a PM with asymmetric chains. The asymmetry in the mechanism poses difficulties in modeling it in the sense that each chain has

to be modeled differently. This not only is tedious but also further hinders the simplification of the equations which relies heavily on the symmetry within the mechanism. To tackle this problem and to have an approach that can be generalized to other devices, an alternative way to model the mechanism is needed. Note that unlike D-H parameterization for serial mechanisms forward kinematics [27], there are few mature methodologies to derive the dynamics for any PMs in analytical form. It mostly works on a case-by-case basis with the help of special structures or symmetry within each mechanism. Often times, the model is then simplified to some level so it can work with the chosen control design methodologies.

To speed up this phase and to develop a more general approach, we build the model of the device using Simscape, a module within the MATLAB/Simulink environment, and then obtain the targeted LTI state-space model by applying the MATLAB linearization tool. Simscape is chosen for its support for PM modeling and its ease of linearization. The approach has good generalization in the sense that it will also work on other devices with different structures.

The modeling process in Simscape is intuitive and close to how one assembles a mechanism in reality. As TAU (see Fig. 1) is a PM, the modeling starts from the base, the I-column, and continues to the end of each kinematic chains. Finally, the chains are connected by the moving platform, which forms kinematic loops. The tool handle is mounted onto the platform and the tip of the tool is aligned with the center of the moving platform. The I/O of the model is defined to match the physical machine and to have a good interface for the next step. The input $\mathbf{u}$ to the device is defined as $\boldsymbol{\tau}$ and the output $\mathbf{y} = [\dot{\mathbf{q}}^T, \mathbf{q}^T]^T$.

The next step is linearization. With the I/O defined in the previous step, the linearized model is obtained by simply applying the Linear Analysis Tool (LAT) from MATLAB. An operating point (OP) is chosen

$$\mathbf{q}_{OP} = \mathbf{0}_{6 \times 1}, \dot{\mathbf{q}}_{OP} = \mathbf{0}_{6 \times 1}$$

and the system is linearized around that point. Here, we use $\mathbf{0}_{m \times n}$ as a shorthand notation for an $m \times n$ matrix/vector that is full of zeros. The $\boldsymbol{\tau}_{OP}$ is obtained by running a static simulation of the mechanism at the OP. For any given OP, the tool should be static at the OP under the corresponding $\boldsymbol{\tau}_{OP}$. The result of the LAT is an LTI state-space model

$$\begin{cases} \delta \dot{\mathbf{x}} = \mathbf{A} \delta \mathbf{x} + \mathbf{B} \delta \mathbf{u} \\ \delta \mathbf{y} = \mathbf{C} \delta \mathbf{x} \end{cases} \quad (1)$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ matrices are the constant coefficients in this form. Here, the state vector is $\delta \mathbf{x}$ instead of $\mathbf{x}$ because it is linearized at an OP and the result should be with respect to that OP where

$$\delta \mathbf{x} = \mathbf{x} - \mathbf{x}_{OP}$$

$\mathbf{x}$ is the state vector and $\mathbf{x}_{OP}$ is the value the state vector should be at the OP. This is also true for $\delta \mathbf{u}$, $\delta \mathbf{y}$, and a few other variables below. Note that currently, the state vector is a 12×1 vector decided by LAT. It involves states related to things like passive spherical joints in the mechanism, which can be unreliable for state feedback and impossible to measure with the current hardware. Thus, a linear transformation is needed for the model to switch to another set of desired states that are only related to the active joints, monitored by the encoders. Because $\mathbf{C}$ is invertible, (1) is transformed into

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

where

$$\begin{aligned} \delta \tilde{\mathbf{x}} &= [\delta \dot{\mathbf{q}}^T, \delta \mathbf{q}^T]^T = [(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{OP})^T, (\mathbf{q} - \mathbf{q}_{OP})^T]^T, \\ \tilde{\mathbf{A}} &= \mathbf{C} \mathbf{A} \mathbf{C}^{-1}, \tilde{\mathbf{B}} = \mathbf{C} \mathbf{B}, \tilde{\mathbf{C}} = (\mathbf{0}_{6 \times 6} \quad \mathbf{I}_{6 \times 6}) \end{aligned}$$

and $\mathbf{I}_{n \times n}$ is the $n \times n$ identity matrix.

Then, the state vector needs to be augmented with some integration of error states to eliminate the steady-state error.

$$\delta \mathbf{x}_{au} = [\delta \tilde{\mathbf{x}}^T, \delta \mathbf{x}_I^T]^T$$

These added integration of error states are

$$\delta \mathbf{x}_I = \int (\delta \mathbf{r} - \delta \mathbf{q}) dt = \int (\mathbf{r} - \mathbf{q}) dt$$

where

$$\delta \mathbf{r} = \mathbf{r} - \mathbf{r}_{OP}, \mathbf{r}_{OP} = \mathbf{q}_{OP}$$

and $\mathbf{r}$ is the reference signal in joint space.

The augmented model becomes

$$\delta \dot{\mathbf{x}}_{au} = \mathbf{A}_{au} \delta \mathbf{x}_{au} + \mathbf{B}_{au} \delta \mathbf{u} + \mathbf{B}_{ref} \delta \mathbf{r} \quad (2)$$

with

$$\begin{aligned} \mathbf{A}_{au} &= \left(\begin{array}{c|c} \tilde{\mathbf{A}}_{12 \times 12} & \mathbf{0}_{12 \times 6} \\ \hline \mathbf{0}_{6 \times 6} & -\mathbf{I}_{6 \times 6} \end{array} \right) \\ \mathbf{B}_{au} &= \left(\begin{array}{c} \tilde{\mathbf{B}}_{12 \times 6} \\ \hline \mathbf{0}_{6 \times 6} \end{array} \right), \mathbf{B}_{ref} = \left(\begin{array}{c} \mathbf{0}_{12 \times 6} \\ \hline \mathbf{I}_{6 \times 6} \end{array} \right) \end{aligned}$$

3.3 Discrete Time Control Design

The control plant in (2) is a continuous time model. It will be transformed into discrete time by the Zero-Order Hold (ZOH) method and then applied for discrete time control design.

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

where T_s is the sampling time and it is set to be 1 ms (1000 Hz refresh rate) [1, 14]. The model from (2) is transformed into

$$\delta \mathbf{x}_{au}[j+1] = \mathbf{A}_{auDT} \delta \mathbf{x}_{au}[j] + \mathbf{B}_{auDT} \delta \mathbf{u}[j] + \mathbf{B}_{refDT} \delta \mathbf{r}[j]$$

where

$$\begin{aligned} \mathbf{A}_{auDT} &= e^{\mathbf{A}_{au} T_s} \\ \mathbf{B}_{auDT} &= \left(\int_0^{T_s} e^{\mathbf{A}_{au} t} dt \right) \mathbf{B}_{au} \\ \mathbf{B}_{refDT} &= \left(\int_0^{T_s} e^{\mathbf{A}_{au} t} dt \right) \mathbf{B}_{ref} \end{aligned}$$

The e here is the natural exponential constant, also known as Euler's number. The result is a discrete time LTI state-space model control plant, denoted by a subscript of auDT.

So we formulate the control problem as an optimization problem to find the optimal gain $\mathbf{K}_{auDT}$ such that the state-feedback control law

$$\delta \mathbf{u}[j] = -\mathbf{K}_{auDT} \delta \mathbf{x}_{au}[j]$$

minimizes the quadratic cost function with a LQR-based formulation

$$\text{cost}(\delta \mathbf{u}) = \sum_{j=1}^{\infty} \left(\delta \mathbf{x}_{au}^T[j] \mathbf{Q} \delta \mathbf{x}_{au}[j] + \delta \mathbf{u}^T[j] \mathbf{R} \delta \mathbf{u}[j] \right)$$

The $\mathbf{Q}$ and $\mathbf{R}$ matrices are tuned to be

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

The actual control command sent to the motors is

$$\mathbf{u}[j] = \mathbf{u}_{OP}[j] + \delta \mathbf{u}[j] = \boldsymbol{\tau}_{OP}[j] - \mathbf{K}_{auDT} \delta \mathbf{x}_{au}[j]$$

3.4 Gain Scheduling with Multiple OPs

So far, we revisit the approach introduced in [34] and the results with a single OP (SOP) in that paper are quite satisfying. However, we also noticed that the performance drops when the tool moves away from the SOP, which is the origin of the workspace. After a bit of tuning, we realized our performance is limited by the hardware capacity and the accuracy of the linearized model. Gain scheduling is a common nonlinear control technique [18] to design subcontrollers based on linearized models of the nonlinear plant at different OPs. This technique can be dated back to the 1960s and it also has some recent development. Over the years, it is sometimes combined with the study of linear parameter varying (LPV) systems. One of the studies related to haptics is done by Walsh and Forbes [30]. In this paper, the authors introduce a

very strictly passive gain-scheduled controller considering the passive properties of the controller. However, as most passivity-based control, it may behave too conservatively when rendering a stiff wall. In our study, as a first step to introduce gain scheduling in our control, we follow the classic gain scheduling method [18]. Therefore, a family of linearized models at different OPs is introduced to be the piecewise linear model to describe the full dynamics over the whole workspace. To achieve good performance even on the edge of the workspace, 27 ($3 \times 3 \times 3$) OPs are chosen as follows

$$\begin{aligned} {}^{OP} \mathbf{X}^{m,n,l} &= \begin{pmatrix} {}^{OP} p_x^{m,n,l} \\ {}^{OP} p_y^{m,n,l} \\ {}^{OP} p_z^{m,n,l} \\ \mathbf{0}_{3 \times 1} \end{pmatrix} = \begin{pmatrix} 25(m-2) \\ 25(n-2) \\ 25(l-2) \\ \mathbf{0}_{3 \times 1} \end{pmatrix}, \\ m, n, l &\in \{1, 2, 3\} \end{aligned}$$

where m, n, l are the OP indices in x, y, z directions, respectively. The number 25 is the half of the side length of the $50 \times 50 \times 50$ mm³ workspace and the unit is mm. Accordingly, the plant models and the controllers associated with these OPs are generated with the same procedures and $\mathbf{Q}, \mathbf{R}$ matrices. The resulting family of linear controllers serves as a single controller whose parameters are changed based on the scheduling variables. The scheduling variables are all the variables in $\mathbf{X}_{haptic}$, which is estimated based on the joint encoder readings $\mathbf{q}_{haptic}$ and forward kinematics.

A naive switching method among these local linear controllers is based on the closest OP to the current $\mathbf{X}_{haptic}$. The switching mechanism is illustrated in Fig. 3a as a two-dimensional (2D) case. The background with different shades of gray indicates different sets of control gains based on the different OPs (red dots). Note that the size of the colored dots is not relevant and is only for illustration purposes. To have some consistency with Fig. 2, the light blue ball still represents where the graphic tool is. Note that in this view, the graphic and the haptic tool coincide because this view is looking at the wall and there are no constraints on the xy -plane. The area circled by black dashed line is a zoom-in illustration of what could be the estimated trajectory when a user move the tool right along the switching boundary. The zig-zag gray solid line is a possible trajectory for the tool on a millimeter scale due to user hand tremor, sensor noise, and numeric error, prior to accounting the effect of control. This trajectory will lead to rapid switching which is not an ideal behavior for gain scheduling controllers in general. The switching is more severe when the tool is working near the intersecting point of two switching boundaries because potentially it can switch among four sets of control gains. The discontinuity in the control signals based on this switching mechanism leads to oscillations in motion, which can be sensed by the user.

Fig. 3 Illustrations of different gain scheduling schemes. The red dots are OPs on the wall. The gray backdrop indicates change of control gain. In (a) the change of control gain has clear boundaries, while in (b) the change is gradual and it is illustrated with gradually changing gray. The light blue ball is the position of the tool

(a) nearest OP

(b) interpolating

To eliminate switching boundaries, a smooth transition can be achieved by using e.g. linear interpolation based on the distances to the closest set of OPs. Although we are using a multivariate linear interpolation, the mathematics is still based on the 1D case that

$$\frac{q_v - b_{v_1}}{q_x - b_{x_1}} = \frac{b_{v_2} - b_{v_1}}{b_{x_2} - b_{x_1}}$$

where v is the value associated with x , and q_v is the value to be interpolated based on q_x . (b_{x_1}, b_{v_1}) and (b_{x_2}, b_{v_2}) are the value pairs of the break points, indicated by the superscript b . q_v can be calculated as

$$q_v = \frac{b_{x_2} - q_x}{b_{x_2} - b_{x_1}} b_{v_1} + \frac{q_x - b_{x_1}}{b_{x_2} - b_{x_1}} b_{v_2}$$

With higher dimensions, this 1D case is performed once in each direction. Because this is a 1D case, b_x is only a scalar, whereas for nD case, b_x is a nD vector

$$b_x = [b_{x_1}, b_{x_2}, \dots, b_{x_n}], b_x \in \mathbb{R}^n$$

The associated value b_v is still a scalar and the value pair for a break point becomes (b_x, b_v) . We use $f(\dots)$ to denote the associated value v with the corresponding set of x in different dimensions. A pseudo code version of the nD linear interpolation algorithm is shown in Algorithm 1.

Note that the order of the directions in which the interpolation is performed does not matter and the number of break points grows exponentially with the dimension. In the 2D case shown in Fig. 3b, the set contains 4 OPs and in 3D it will contain 8 (2^3) OPs. As also stated in the algorithm input data, for the j^{th} dimension, i_j can only take the value of either 1 or 2. This means that for all the OPs, the value for their j^{th} coordinate can only be either $b_{x_j^1}$ or $b_{x_j^2}$. Given there are n dimensions and for each dimension the value of the coordinate only has 2 choices, we can conclude that there are 2^n different OPs in the set of closest OPs for this

Algorithm 1: nD linear interpolation.

Input: 2^n OPs with the notation

$$b_x^{(i_1, i_2, \dots, i_n)} = [b_{x_1^{i_1}}, b_{x_2^{i_2}}, \dots, b_{x_n^{i_n}}],$$

$$b_v^{(i_1, i_2, \dots, i_n)} = f(b_x^{(i_1, i_2, \dots, i_n)}),$$

$$i_j \in \{1, 2\}, \forall j \in \{1, 2, \dots, n\}$$

Input: a QP with the notation

$$q_x = [q_{x_1}, q_{x_2}, \dots, q_{x_n}]$$

Output: q_v

for The $m = 1, 2, \dots, n$ dimension **do**

Short hand notation

$$c_{before} = [q_{x_1}, q_{x_2}, \dots, q_{x_{m-1}}],$$

$$c_{after} = [b_{x_{m+1}^{i_{m+1}}}, b_{x_{m+2}^{i_{m+2}}}, \dots, b_{x_n^{i_n}}]$$

c_{after} has 2^{n-m} combinations

for each different combination of c_{after} **do**

Compute

$$f(c_{before}, q_{x_m}, c_{after})$$

$$= \frac{b_{x_m^2} - q_{x_m}}{b_{x_m^2} - b_{x_m^1}} f(c_{before}, b_{x_m^1}, c_{after})$$

$$+ \frac{q_{x_m} - b_{x_m^1}}{b_{x_m^2} - b_{x_m^1}} f(c_{before}, b_{x_m^2}, c_{after})$$

end

end

algorithm. Similarly, when computing the m^{th} dimension, c_{after} has 2^{n-m} combinations. For the m^{th} dimension, the $f(\dots)$ values needed for the inner loop computation should be the results of the inner loop from the $(m-1)^{th}$ dimension. In the beginning when $m = 1$, the $f(\dots)$ values are the $b_v^{(i_1, i_2, \dots, i_n)}$ values of the break points.

Because the value is interpolated, it is depicted in Fig. 3b with a gradual changing color of gray without clear switching boundaries as opposed to Fig. 3a. The interpolated values include the controller gain matrices K_{auDT} as well as offsets at the OP such as q_{OP} and u_{OP} .

The new controller is named multi-OP position controller (MOPPC) to highlight the usage of multiple operating points (MOPs). As a comparison, we include the proposed controller in [34] and rename it to SOPPC to emphasize that it is generated from a control plant using a single OP. The baseline controller in that paper is also included as a comparison and renamed as open-loop impedance-type force controller (OLIFC).

4 Simulation Methodology

Simulations are conducted to investigate the performance of different controllers mentioned above. One of the tests frequently used for evaluating force rendering capability is to do a 1D pressing against the wall, as shown in [13, 17], where other moving directions are constrained and therefore ignored. The device only moves in the direction perpendicular to the wall and the penetration in this direction is recorded and reported. Although this test demonstrates the performance of the device at one point in space in a certain direction with a near quasi-static setup, it does not represent the performance over the whole workspace or consider the effect of motions in non-pressing directions. If put in an application sense, this test resembles more to the working conditions during tasks like drilling but gives little insight into tasks like milling or surface polishing, where a major lateral motion is on-going while a pressing force perpendicular to the motion is also present.

To mitigate this, we test the performance of TAU under “pressing while sliding” conditions in this paper. This is also found in papers like [31]. It provides another layer of verification on the performance of the device in addition to “pressing while stationary”. The object we are rendering is still a unilateral stiff wall. “Unilateral” means that the wall only constrains the motion of the tool in one direction [13]. Specifically, it prevents the tool from entering the wall. However, the tool should be free to leave the wall. A set of fixed trajectories is chosen in order to investigate how the performance varies along each trajectory. The trajectory is fed to the controller in real-time, mimicking the result from the last haptic rendering step, collision detection. Two different disturbance scenarios are also designed to represent typical disturbance in working conditions. For each scenario, 20 Monte-Carlo simulations are conducted on the full dynamic model of the device with randomly varying disturbance level and white noise. These

specifications will be further explained in the following subsections.

4.1 Query Trajectory

Eight straight line segments are chosen as a set of representative paths to cover the rendered wall as shown in Fig. 4. Then, these 8 paths are turned into 16 query trajectories (QTs) given that one path can be traversed in two different directions. The performance along all QTs should give a more comprehensive view of the system instead of the performance at one point in the workspace. Each trajectory is parameterized with two waypoints, the start and stop pose of the tool, and the traveling time. The traveling time, regardless of the length of the path, is set to 2 s. The trajectory is also guaranteed to be smooth and second order differentiable. The 16 QTs are labelled in Fig. 4 with numbers close to its starting point.

4.2 Disturbance Scenario

Two scenarios with different disturbance models are introduced to test the performance under different kinds of workload.

4.2.1 Scenario 1: Force Model

Scenario 1 resembles the step test for general control systems. A pure force signal is applied onto the tool,

Fig. 4 Illustration of all 16 QTs on the wall, labelled with number. Note that the two QTs on the same path are separated apart only for clarification. The red dots are OPs on the wall

emulating the pressing force from the user. The force signal is a step signal overlaid with a white noise as shown below

$$F_{sc1}(t) = F_{signal1}(t) + w_{noise}(t)$$

where

$$F_{signal1}(t) = F_c$$

$$w_{noise}(t) = \mathcal{N}(0, 0.5)$$

The step level (F_c) is randomly generated for each simulation uniformly distributed within the range of 8 - 15 N. The step signal holds the whole duration of the simulation. The white noise follows a normal distribution parameterized by a mean of μ and a standard deviation of σ . The notation of $\mathcal{N}(\mu, \sigma)$ is adopted to represent that. The amplitude of the white noise is roughly 10% of the step level. As stated, the same simulation will be run along each QT with a randomization in the step level and added white noise as a Monte-Carlo simulation in order to demonstrate the robustness of the performance.

4.2.2 Scenario 2: Hand Model

The disturbance of Scenario 2 is applied through a hand model. As suggested by [28], the behavior of the hand can be modeled using a spring of 40 N/m and a damper of 3.6 Ns/m, connected in parallel. Unlike Scenario 1, where a pure force signal is generated, a hand motion is designed and applied to the hand. The relative motion and the spring-damper system modeling the hand generate a disturbance on the tool, imitating how the user applies a disturbance.

Contrary to the force model, where the disturbance has a sharp edge to jump from 0 to F_c , the hand model is aimed to apply a disturbance with a smooth transition. Other than that, the idea is still the same: the disturbance should stay constant for a while to confirm if there is a steady state error. This translates into having a hand motion that moves towards the wall and then stay stationary to maintain the level of pressing force, assuming the tool also stays stationary on the surface of the wall. The motion is designed to follow a constant acceleration, constant speed (zero acceleration), constant deceleration profile so that the acceleration follows

$$a_h = \begin{cases} -a_c, & 0 \leq t < 0.45 \\ 0, & 0.45 \leq t < 0.55 \\ a_c, & 0.55 \leq t < 1 \\ 0, & 1 \leq t \leq 2 \end{cases}$$

The time period of acceleration/deceleration is chosen so that the generated disturbance has a smooth transition and little overshoot. With these constraints, a_c is calculated based on F_c and how much the hand needs to move. By designing the hand motion X_{hand} , the disturbance is

generated at run-time as a function of X_{hand} and X_{haptic} .

$$F_{sc2}(t) = F_{signal2}(t) + w_{noise}(t)$$

where

$$F_{signal2}(t) = \text{function}(X_{hand}(t), X_{haptic}(t))$$

4.3 Post-processing and Performance Metrics

Stiffness related metrics are commonly used in evaluating the performance of haptic devices [8] since it considers the level of the disturbance as well as the penetration. However, it is also a calculated value based on a simple yet important assumption that there is indeed a penetration. In the calculation, the penetration is always in the denominator. If the penetration is close to zero and positive, the calculated stiffness will be huge. It does not necessarily mean that the stiffness rendered by the device can reach that level constantly. If the penetration is exactly zero, which is the ideal case for rendering a stiff wall, the calculated stiffness is infinity. This is a bit problematic if the next step is to take the average of the values. If the penetration is negative, which by definition means that there is still a distance between the tool and the wall, the stiffness calculation loses its physical meaning because there is no collision. Due to factors like numeric error, digitization and noise in sensors, oscillations, and so on, these three cases can all occur within one simulation. For this study specifically, a stiffness higher than 1.6×10^5 N/m is considered to be cut off, as the disturbance can reach no larger than 16 N and a penetration less than 0.1 mm is considered unidentifiable by the encoders. If this value is reached, we consider the device is rendering infinite stiffness and these data points will be processed separately. This is later referred to as the cut-off criteria. Note that the criteria is met mostly because there is too little penetration. So, when the criteria is met, it is also considered to have no penetration, which corresponds to the infinite stiffness.

To maintain consistency with the previous findings in [34], minimal stiffness K_{min} along the QT is chosen for evaluating Scenario 1 and average stiffness over time K_{avg} for Scenario 2 for each simulation.

K_{min} is calculated by

$$K_{min} = \min_j \left(\frac{F_{sc1}[j]}{D_z[j]} \right), \forall j \in \{1, 2, \dots, N\}$$

and it is chosen to report the worst-case scenario for Scenario 1. $D_z(t)$ is the penetration into the wall and N is the number of time steps for the simulation. Corresponding to the sharp step change in the disturbance, there is a relatively large initial penetration, as shown in Fig. 5. As the disturbance holds its constant level after the step, the penetration should be less noticeable. Because the initial penetration exists only for a short period of time compared

Fig. 5 A plot showing a typical disturbance for Scenario 1 and typical responses for all three controllers in z direction (penetration)

to the whole simulation time, the average stiffness over time will filter out this low point.

Because the disturbance in Scenario 2 has a smooth transition, as depicted in Fig. 6, the relatively large initial penetration found in Fig. 5 is not presented in Scenario 2. The slow change in disturbance value also gives controllers enough time to response, so the performance during the whole simulation time is quite consistent. Large

penetrations are more likely as a result of the noise. Hence, K_{avg} can better describe the performance along QTs.

$$K_{avg} = \frac{1}{N} \sum_{j=1}^N \left(\frac{F_{sc2}[j]}{D_z[j]} \right)$$

As stated previously, no penetration is actually the aim of the controller and we consider stiffness that meets the

Fig. 6 A plot showing a typical disturbance for Scenario 2 and typical responses for all three controllers in z direction (penetration)

cut-off criteria to have reached infinite stiffness. Infinite stiffness ratio (r_{IS}) is the ratio of the time period that the calculated stiffness meets the cut-off criteria over the total simulation time. Because the sampling frequency in discrete time is constant, the ratio is equivalent to

$$r_{IS} = \frac{N_{\text{cut-off}}}{N}$$

Again, N is the total number of time steps for the whole simulation and $N_{\text{cut-off}}$ is the number of time steps that meet the cut-off criteria. r_{IS} should be used as an auxiliary metrics when the results from the stiffness metrics are unclear. When $r_{IS} = 1$, it means that the calculated stiffness can be considered infinite throughout the whole simulation and it should be considered no penetration during the simulation. When $r_{IS} = 0$, it means that there exists a noticeable penetration throughout the whole simulation given the definition of the cut-off criteria. When r_{IS} is between 0 and 1, it just means that the controller manages to render infinite stiffness sometimes.

r_{IS} is accompanied with another auxiliary metrics called maximal off-wall distance (D_{mo}). When r_{IS} is larger than 0, D_{mo} is examined to make sure that the tool never gets too far off the wall, because that is an unwanted behavior.

These metrics will be used in the coming result and discussion sections.

5 Simulation Results

To demonstrate the value level as well as the distribution, (half) violin plots are chosen (see Figs. 7 and 8) to report

Fig. 7 The violin plot about K_{min} for each Scenario 1 simulation along all 16 QTs except for QT 6. For all subplots, the y axis is in N/m for stiffness and the x axis is the QT number. (green: MOPPC, blue: SOPPC, red: OLIFC)

Fig. 8 The violin plot about K_{avg} for each Scenario 2 simulation. For all subplots, the y axis is in N/m for stiffness and the x axis is the QT number. (green: MOPPC, blue: SOPPC, red: OLIFC)

the results corresponding to different scenarios, which can be considered as an enhanced box plot. Each violin plot includes a mini box plot with the circle as median, a rotated kernel density plot on the side, and a horizontal bar indicating the mean.

5.1 Scenario 1 with Force Disturbance

For Scenario 1, K_{min} is chosen as a worst-case scenario estimation. As shown in Fig. 7, the results from three different controllers are shown in two different subplots. The value level of K_{min} for OLIFC is quite consistent, at a value level lower than 3000 N/m. Thus, it is plotted in a separate subplot to show some details and variations in the distribution of the data. The results for both MOPPC and SOPPC are plotted in the top subplot since they are in a similar range. That being said, the results spread roughly between 3000 and 30000 according to the figure. The results for MOPPC, indicated with the color green, can be further separated by the green dashed line at 20000. Similar separation can also be found for the results for SOPPC by the blue dashed line at 9000. The separation suggests that within the results for each controller the results for the first 8 QTs are on a different level from that for the last 8 QTs. The separation will be discussed in details in the discussion section. The red dashed line at 3000 gives a sense of where the results for OLIFC might be as 3000 is loosely the upper bound of the values. The results for QT 6 are omitted for all controllers because the results for MOPPC and SOPPC meet the cut-off criteria.

5.2 Scenario 2 with Hand Disturbance

K_{avg} is the main metrics for evaluating Scenario 2. Note that based on how the post-processing is done, the data points that meet the cut-off criteria are not used for calculating K_{avg} . In Fig. 8, the results for OLIFC still show a consistent stiffness at the level of 6000 N/m, which is the stiffness set for the wall for this controller. The results for OLIFC are still plotted separately from those for SOPPC and MOPPC due to the difference in the order of magnitude. A red dashed line of 6200 is added to the top plot just to illustrate where the numbers for OLIFC will be if the results for all three controllers are plotted together. For SOPPC and MOPPC, the results for the first 8 QTs are omitted because most of them meet the cut-off criteria and should be considered infinite stiffness. These results are also accompanied with an r_{IS} , the infinite stiffness ratio, of 1 or close to 1. With this kind of r_{IS} , D_{mo} (maximal off-wall distance) ensures that the tool actually stays on the surface of the wall instead of breaking the collision and going off the wall back to free space. The D_{mo} for all simulations along all 16 QTs is about 0.2 mm for MOPPC and 0.7 mm for SOPPC, which is an acceptable distance to be considered as on the surface of the wall. Meanwhile, the maximal penetration is around 0.3 mm and 0.9 mm for MOPPC and SOPPC, respectively.

6 Discussion

From the results for both scenarios, the performance is quite consistent for OLIFC. With a slow changing disturbance, it seems to reach its predefined stiffness of 6000 N/m. However, as shown in Scenario 1, with a sudden change in the disturbance, the worst-case stiffness is only about half of the targeted value.

The choice of the value 6000 N/m for OLIFC is not arbitrary in this study. An initial guess is based on guidelines in [1]. With some initial trial-and-errors and some margins, 6000 N/m is set for a balance between less oscillation and higher stiffness. Judging from the results for the other two controllers, higher stiffness cannot be reached due to instability instead of saturation. This might explain why the performance of the OLIFC is so uniform. The targeted 6000 N/m is well within the saturation limits at any point on the wall so that it can be reached everywhere easily.

Because OLIFC treats the wall as a linear spring with a constant stiffness, there is an unavoidable steady-state error in position, hence a penetration. With the maximal disturbance of 15 N and the predefined stiffness of the wall, a steady penetration of 2.5 mm can be expected. Similar results can be found from the simulations. Because the disturbance level uniformly varies from 8 to 15 N as a part of the Monte-Carlo simulation and the penetration varies with

the disturbance level, the average number is slightly better, at around 1.9 mm for Scenario 1 and 1.5 for Scenario 2. The maximal penetration is reached in Scenario 1 with a value of 5.3 mm.

For MOPPC and SOPPC, many things are in common between the two when compared to OLIFC. They are both ambitious in trying to render the wall with the highest stiffness achievable for the physical device. The results are quite similar as shown in Figs. 7 and 8 with a magnitude of 10^4 in N/m. In comparison with each other, MOPPC seems to have a better performance across all scenarios and QTs, except for QT 10 in Scenario 2. The improvement can be interpreted as the benefit of having a more accurate model. With gain scheduling, the controller is using a set of more appropriate gain that is specifically designed for the current neighborhood. The exception on QT 10 can be caused by the cut-off criteria. The r_{IS} for SOPPC in this scenario ranges from 0.56 to 0.74 while that for MOPPC ranges from 0.80 to 0.88. This means that more data points with high values are cut off and taken out in the calculation for Fig. 8. If they are added back, MOPPC can outperform SOPPC along QT 10.

In the result section, we raise the notion that the performance for the first 8 QTs seems better than that of the last 8 QTs for both MOPPC and SOPPC. This is quite clear with the dashed lines in Fig. 7. The results for first 8 QTs are omitted in Fig. 8 because they are much higher than that of the last 8 QTs. This difference indicates the QTs can be divided into 2 groups. Our hypothesis is that this division can be based on the position of starting point of the QTs, more specifically, whether the starting point is on the upper or lower half of the wall. Based on this criterion, the QTs can be regrouped into 3 groups:

1. QT 1 - 4 and 6 - 8 (upper half);
2. QT 9 - 12 and 14 - 16 (lower half);
3. QT 5 and 13 (boundary cases)

The starting point is on the upper or The boundary of upper and lower half of the wall is the x axis ($p_y = 0$), where QT 5 and 13 are on. This division is close to the division discovered in the figures. The reason behind this division can be that a good initialization is important for the performance along the whole QT. In the previous study [34], a tendency of the upper half of the wall showing a better performance was identified. Combining these findings, we can assume that the performance along a QT tends to be similar to the performance at the starting point. It maintains the performance that way even when the tool later moves to a point that is queried to show a different performance.

According to this hypothesis, most of the first 8 QTs are from the first group so their performance metrics is higher than the last 8 QTs which are mostly from the second group. The performance for QT 5 and 13 can be considered slightly

different because they actually belong to the third group. The results for QT 6 and 14 are higher than the rest of their group, respectively. That is probably enhanced by the mechanical structure of the device as Chain 3 has a revolute joint instead of a universal joint for a passive joint and yz-plane is considered a symmetric plane for the device. Note that the hypothesis can be over-fitting the data because the number of QTs in different groups are limited.

7 Conclusion and Future Work

In this paper, we have revisited our proposed approach to solve the problem of rendering stiff interaction as a position controller design problem and extended this approach with gain scheduling so that the performance drop in the lower half of the wall can be improved. The haptic rendering case is still the interaction with a stiff wall. The performance improvement is investigated through Monte-Carlo simulations under new “pressing and sliding” conditions. From the results and discussions, we can conclude that the proposed method is able to render a stiffness that is on the magnitude of 10^4 N/m on TAU device stably. This is a huge improvement against the traditional OLIFC as both guidelines and tests suggest that the stable stiffness it can render is on the level of 10^3 N/m. Note that similar conclusions are also reported in [13] that they also achieve high stiffness beyond stability limits suggested by papers like [1, 10]. Compared to the previous study [34], the new scenario of sliding investigate performance under a different condition which resembles more to the task of polishing in dental procedures instead of drilling. The performance is consistent with previous findings in magnitude but it proves hard to estimate the performance along the trajectory only with the knowledge of the performance at each point on the trajectory. Instead, the performance at the starting point of the trajectory gives a good estimate of the performance along the whole trajectory. With the newly proposed MOPPC, the transition between areas using different control gains proves to be smooth. Compared to SOPPC, MOPPC shows significant improvement in both scenarios by adopting classic gain scheduling.

Currently, the proposed approach can be used as an alternative solution to deliver high stiffness in rendering stiff objects instead of the traditional spring-damper controller. Nonetheless, it is still at its early phase of development and further research is needed to explore the full potential of the approach being an all-purpose haptic rendering approach, suitable for deformable objects. Furthermore, we expect that more complex scenarios, which require more accurate modeling and faster execution, will reveal more advantages of our proposed approach.

Author Contributions **Yang Wang:** Conceptualization, Methodology, Software, Investigation, Formal Analysis, Visualization, Writing - Original Draft. **Lei Feng:** Supervision, Writing - Review and Editing. **Kjell Andersson:** Supervision, Writing - Review and Editing.

Funding Open access funding provided by Royal Institute of Technology. Yang Wang would like to thank China Scholarship Council (CSC) for supporting his Ph. D. study. Lei Feng is financially supported by KTH XPRES.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abbott, J., Okamura, A.: Effects of position quantization and sampling rate on virtual-wall passivity. *IEEE Trans. Robot.* **21**(5), 952–964 (2005). <https://doi.org/10.1109/TRO.2005.851377>
2. Abdellatif, H., Heimann, B.: Advanced model-based control of a 6-dof hexapod robot: a case study. *IEEE/ASME Trans. Mechatron.* **15**(2), 269–279 (2010). <https://doi.org/10.1109/TMECH.2009.2024682>
3. Adams, R., Hannaford, B.: Stable haptic interaction with virtual environments. *IEEE Trans. Robot. Autom.* **15**(3), 465–474 (1999). <https://doi.org/10.1109/70.768179>
4. Ahmad, A., Andersson, K., Sellgren, U.: An optimization approach toward a robust design of six degrees of freedom haptic devices. *J. Mech. Des.* **137**(4). <https://doi.org/10.1115/1.4029514> (2015)
5. Ahmad, A., Khan, S., Anderson, K.: Kinematics and dynamics of a novel 6-DoF TAU haptic device. In: 2011 IEEE International Conference on Mechatronics, pp. 719–724, Istanbul (2011). <https://doi.org/10.1109/ICMECH.2011.5971209>
6. Cheng, H., Yiu, Y.K., Li, Z.: Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Trans. Mechatron.* **8**(4), 483–491 (2003). <https://doi.org/10.1109/TMECH.2003.820006>
7. Clavel, R.: DELTA, a fast robot with parallel geometry. In: Burckhardt, C.W. (ed.) *Proc of the 18th International Symposium on Industrial Robots*, pp. 91–100. Springer-Verlag, New York (1988)
8. Colgate, J., Brown, J.: Factors affecting the Z-Width of a haptic display. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3205–3210, San Diego, CA, USA (1994). <https://doi.org/10.1109/ROBOT.1994.351077>

9. Colgate, J., Grafing, P., Stanley, M., Schenkel, G.: Implementation of stiff virtual walls in force-reflecting interfaces. In: Proceedings of IEEE Virtual Reality Annual International Symposium, pp. 202–208 (1993). <https://doi.org/10.1109/VRAIS.1993.380777>
10. Colgate, J.E., Stanley, M.C., Brown, J.M.: Issues in the haptic display of tool use. In: Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, vol. 3, pp. 140–145, Pittsburgh (1995). <https://doi.org/10.1109/IROS.1995.525875>
11. Corbel, D., Gouttefarde, M., Company, O., Pierrot, F.: Towards 100G with PKM. Is actuation redundancy a good solution for pick-and-place? In: 2010 IEEE International Conference on Robotics and Automation, pp. 4675–4682, Anchorage (2010). <https://doi.org/10.1109/ROBOT.2010.5509921>
12. Díaz-Rodríguez, M., Valera, A., Mata, V., Valles, M.: Model-based control of a 3-DOF parallel robot based on identified relevant parameters. IEEE/ASME Trans. Mechatron **18**(6), 1737–1744 (2013). <https://doi.org/10.1109/TMECH.2012.2212716>
13. Desai, I., Gupta, A., Chakraborty, D.: Rendering stiff virtual walls using model matching based haptic controller. IEEE Trans. Haptics **12**(2), 166–178 (2019). <https://doi.org/10.1109/TOH.2019.2893926>
14. Duriez, C., Dubois, F., Kheddar, A., Andriot, C.: Realistic haptic rendering of interacting deformable objects in virtual environments. IEEE Trans. Vis. Comput. Graph. **12**(1), 36–47 (2006). <https://doi.org/10.1109/TVCG.2006.13>
15. Gillespie, R.B., Cutkosky, M.R.: Stable user-specific haptic rendering of the virtual wall. In: Proceedings of the ASME International Mechanical Engineering Congress and Exhibition, vol. 58, pp. 397–406 (1996)
16. Gottlieb, R., Baechle, M.A., Janus, C., Lanning, S.K.: Predicting performance in technical preclinical dental courses using advanced simulation. J. Dent. Educa. **81**(1), 101–109 (2017). <https://doi.org/10.1002/j.0022-0337.2017.81.1.tb06252.x>
17. Hannaford, B., Ryu, J.H.: Time-domain passivity control of haptic interfaces. IEEE Trans. Robot. Autom. **18**(1), 1–10 (2002). <https://doi.org/10.1109/70.988969>
18. Khalil, H.K., Grizzle, J.W.: Nonlinear Systems, vol. 3. Prentice Hall, Upper Saddle River, NJ (2002)
19. Lin, M.C., Otaduy, M.: Haptic Rendering. Algorithms, and Applications. A K Peters/CRC Press, Foundations (2008). <https://doi.org/10.1201/b10636>
20. Oftadeh, R., Aref, M.M., Taghirad, H.D.: Explicit dynamics formulation of stewart-gough platform: a newton-euler approach. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2772–2777, Taipei (2010). <https://doi.org/10.1109/IROS.2010.5653157>
21. Ortega, M., Redon, S., Coquillart, S.: A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. IEEE Trans. Vis. Comput. Graph. **13**(3), 458–469 (2007). <https://doi.org/10.1109/TVCG.2007.1028>
22. Otaduy, M.A., Lin, M.C.: A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. IEEE Trans. Robot. **22**(4), 751–762 (2006). <https://doi.org/10.1109/TRO.2006.876897>
23. Park, H., Lee, J.: Adaptive impedance control of a haptic interface. Mechatronics **14**(3), 237–253 (2004). [https://doi.org/10.1016/S0957-4158\(03\)00040-0](https://doi.org/10.1016/S0957-4158(03)00040-0)
24. Ruspini, D.C., Kolarov, K., Khatib, O.: The haptic display of complex graphical environments. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97, pp. 345–352. ACM Press/Addison-Wesley Publishing Co., Los Angeles, CA, USA (1997). <https://doi.org/10.1145/258734.258878>
25. Ryu, J.H., Yoon, M.Y.: Memory-based passivation approach for stable haptic interaction. IEEE/ASME Trans. Mechatron. **19**(4), 1424–1435 (2014). <https://doi.org/10.1109/TMECH.2013.2285380>
26. Salisbury, K., Conti, F., Barbagli, F.: Haptic rendering: introductory concepts. IEEE Comput. Graph. Appl. **24**(2), 24–32 (2004). <https://doi.org/10.1109/MCG.2004.1274058>
27. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics. 2nd edn. Springer Handbooks Springer International Publishing, New York (2016). <https://doi.org/10.1007/978-3-319-32552-1>
28. Speich, J.E., Shao, L., Goldfarb, M.: Modeling the human hand as it interacts with a telemanipulation system. Mechatronics **15**(9), 1127–1142 (2005). <https://doi.org/10.1016/j.mechatronics.2005.06.001>
29. Sun, X., Sellgren, U., Andersson, K.: Situated design optimization of haptic devices. Procedia CIRP **50**, 293–298 (2016). <https://doi.org/10.1016/j.procir.2016.04.096>
30. Walsh, A., Forbes, J.R.: A very strictly passive gain-scheduled controller: theory and experiments. IEEE/ASME Trans. Mechatron. **21**(6), 2817–2826 (2016). <https://doi.org/10.1109/TMECH.2016.2565387>
31. Wang, D., Zhang, X., Zhang, Y., Xiao, J.: Configuration-based optimization for six degree-of-freedom haptic rendering for fine manipulation. IEEE Trans. Haptics **6**(2), 167–180 (2013). <https://doi.org/10.1109/TOH.2012.63>
32. Wang, D., Zhao, X., Shi, Y., Zhang, Y., Hou, J., Xiao, J.: Six degree-of-freedom haptic simulation of probing dental caries within a narrow oral cavity. IEEE Trans. Haptics **9**(2), 279–291 (2016). <https://doi.org/10.1109/TOH.2016.2531660>
33. Wang, Y., Feng, L., Andersson, K.: A new approach to haptic rendering by position control. In: 2019 IEEE 15th International Conference on Control and Automation (ICCA), pp. 148–153 (2019). <https://doi.org/10.1109/ICCA.2019.8899956>
34. Wang, Y., Feng, L., Andersson, K.: A position-control based approach to haptic rendering of stiff objects. IEEE Trans. Haptics: 1–1. <https://doi.org/10.1109/TOH.2020.3044682> (2020)
35. Yu, G., Wang, D., Zhang, Y., Xiao, J.: Simulating sharp geometric features in six degrees-of-freedom haptic rendering. IEEE Trans. Haptics **8**(1), 67–78 (2015). <https://doi.org/10.1109/TOH.2014.2377745>
36. Zilles, C.B., Salisbury, J.K.: A constraint-based god-object method for haptic display. In: Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, vol. 3, pp. 146–151, Pittsburgh, PA, USA (1995). <https://doi.org/10.1109/IROS.1995.525876>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yang Wang received the B.S. degree in Mechanical Engineering and Automation from Beijing Jiaotong University in 2014 and M.Sc. in Eng. degree in Robotics from Johns Hopkins University in 2016. He is currently pursuing a Ph.D. degree in Machine Design at KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include Haptic Robotics, Medical Robotics, Mechanical Design, and Dynamics and Control.

Kjell Andersson is a full professor in Machine Design at the department of Machine Design, KTH Royal Institute of Technology. He has over 20 years' experience in research in modeling, simulation and optimization of dynamical behavior of mechanical systems. 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.

Lei Feng (M07) 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. In 2012, he joined the Mechatronics and Embedded Control System Division, KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently an Associate Professor. 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.