Modeling and Control of a 5-DOF Parallel Continuum Haptic Device

Margaret Koehler, Thor Morales Bieze, Alexandre Kruszewski, Allison Okamura, Christian Duriez

Modeling and control of a 5-DOF parallel continuum haptic device

Margaret Koehler, Thor Morales Bieze, Alexandre Kruszewski,
Allison M. Okamura, Christian Duriez

***Abstract*—In this paper, we propose a new continuum robotics approach for haptic rendering and co-manipulation. This approach is illustrated using a robotic interface with 6 motorized fixed axes connected by deformable beams, in parallel, to an end-effector with 5 degrees of freedom (DOF). Apart from the rotation of the motors, this design has no articulation, and the motion of the end-effector is achieved by deformation of the beams. The flexible beams are equipped with bending sensors, and the motors have encoders. We use a non-linear finite element mechanical model of the robot based on a mesh of beam elements that is computed in real time at 20 Hz. The bending sensors are incorporated into the model which allows us to obtain an accurate estimate of the force exerted by the user on the end-effector. The model enables a new methodology for calculating the workspace of the continuum haptic device. The model also is propagated to a higher frequency loop (500 Hz) which performs sensing and control of the robot at high rates, using an admittance-type control to command new positions of the actuators. We show that this control methodology allows haptic rendering of virtual walls that are stiffer than the natural stiffness of the robot. Finally, we demonstrate use of the device for simple co-manipulation tasks.**

I. Introduction

Robots with a parallel structure offer advantageous properties for robot design, including low on-board mass, high rigidity with a good distribution of forces, and precise and fast positioning. These properties make parallel robots particularly suitable for many applications, ranging from fast pick and place tasks to mobile platforms for flight simulators to haptic devices. On the other hand, the fabrication of parallel robots is not straightforward because they usually contain more articulations compared to serial robots and thus require precise mechanical realization. Moreover, the workspace analysis and control methods are usually more sophisticated. In recent years, the field of soft robotics has revisited the design of robots, in particular using deformation as the principle of motion, instead of using joints between rigid bodies. In this study, we propose a continuum parallel structure as a new design principle for a robot (or haptic device), the advantage being the simplicity of fabrication. We also present a methodology for workspace analysis, sensing the effector position, and designing the control. Our approach is implemented on a robot with 6 motors and 5 DOF of output, and we evaluate its use for haptic rendering. The robot is shown in Figure 1.

A. M. Okamura and M. Koehler are with Stanford University, CHARM Lab. T. Morales Bieze, A. Kruszewski, and C. Duriez are with University of Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France.

Corresponding author: Christian Duriez
christian.duriez@inria.fr

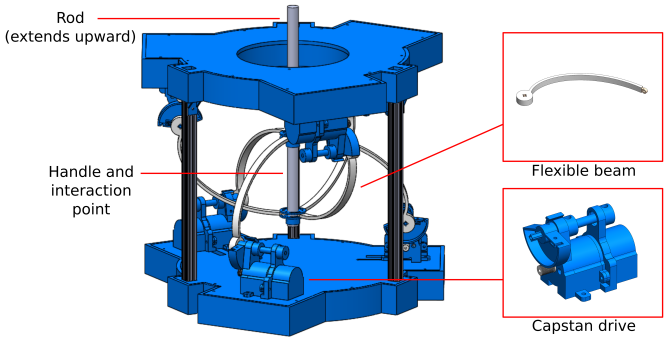


Fig. 1. The 5-DOF deformable haptic device. The device consists of six flexible beams, each of which is rotated at its base by a motor via a capstan drive. The flexible beams are rigidly attached to the handle, with their attachment centered around the controlled interaction point. Flex sensors are mounted on each beam for position feedback. The clip closest to each motor attaches a flex sensor rigidly, while the other two clips allow the flex sensor to slide while constraining it to move with the beam. A motor controller board (not shown) manages low-level position control of the motors. The device is modeled and controlled using a SOFA simulation ([1]). Reflective markers were attached to the top of the handle to collect data on the handle position and orientation for evaluation (not used for control).

The device renders force feedback in three translational and two rotational DOF. The device is controlled to render stiff constraints and free space using a quasi-static model of the device.

Kinesthetic haptic devices provide force feedback to a user interacting with a virtual environment, with applications in medical training, rehabilitation, and computer-aided design ([2]). Most kinesthetic haptic devices have been composed of rigid components ([3]). Deformable transmissions have been proposed to decrease friction, ease fabrication, and provide some benefits in control ([4], [5]).

In [6], a compliant mechanism haptic device based on joint deformations is calibrated with a procedure that is dependent on external sensing to cancel the stiffness forces of the device. This calibration requires sampling the full workspace, rather than using a model with a few parameters. In [7], we presented a 2-DOF force feedback device based on a compliant structure and a model-based procedure to cancel the natural stiffness of the device and render another stiffness using backdrivable motors. Unlike this previous work, in this paper we provide 5-DOF force-feedback and integrate deformation sensing on the device via flex sensors. This gives information on the interaction force of the user and eliminates the need for backdrivable motors. Also, the control for this device uses a model of the device at high rates, rather than relying on the

stiffness mapping for control between model updates, as was done in [7].

The paper is organized as follows. First, we provide a short review of the related work. Then we introduce the 5-DOF device, presenting its mechanical design and deriving a mechanical model of the device, in particular the modeling of the compliance. We perform an original workspace analysis based on a rendering of an isotropic force on all points of the workspace in simulation. We then demonstrate the integration of the flex sensors into the model of the device based on a device-specific sensor calibration. We present the algorithm used to perform haptic rendering, in particular the multi-rate implementation of the inverse models. Finally, we present the experimental results that validate the proposed sensing and rendering methodologies.

This paper leverages a novel haptic interface design and FEM beam modeling to make the following contributions:

- a proposed method for workspace analysis of the haptic interface based on the model
- a method for integrating flex sensors into the model which allows for estimation of both the position of and force exerted on the end-effector
- a method for multi-rate haptic control which allows for haptic rendering of simple virtual environments

These methods have been implemented and demonstrated on a new continuum parallel robot design with 6 motors and 5 degrees of freedom, and are also applicable to other device designs.

II. Related work

The design and control of our 5-DOF device builds on extensive previous work in kinesthetic haptic devices, which provide force feedback that can be sensed in the joints and muscles of a user in response to the user's movement in space. Relevant background on the design and control of these devices is presented below.

Kinesthetic haptic devices are classified into two categories, impedance-type and admittance-type ([8]). Arata et al. provide a summary of devices and their types and structures ([3]). Impedance-type devices measure the user's position, apply this displacement to a virtual environment to determine an interaction force to render, and apply this force via the end-effector. Typically, these devices are designed to have low mechanical impedance (e.g., low mass and low friction), and the impedance of the device itself is neglected in the control. Some controllers do seek to compensate for the mechanical impedance of the device, as was described by Hogan in [9]. Admittance-type devices allow for higher mechanical impedance in the device itself. For example, they can include geared, non-backdrivable motors. Such devices incorporate a force sensor at the end-effector and use this measured force in the virtual environment to obtain a target position. The actuators then drive the device to match this target position, masking the inherent device dynamics. Impedance-type devices tend to be less expensive and simpler to control than admittance-type devices, since they do not require a force sensor and do not use a nested position controller. On the other hand, admittance-type devices are better at rendering stiff constraints and can typically output higher forces than impedance-type devices.

A variety of factors contribute to the performance of a kinesthetic haptic device, and these factors have been well-studied for rigid haptic devices. The dynamic range of a haptic device can be measured in terms of the Z-width, which describes the range of impedances that can be rendered. Factors including time delay, discretization, and sensor quantization all contribute to the maximum impedance range that can be stably rendered ([10], [11], [12], [13]). Damping has been found to be critical for device passivity, though passivity is usually regarded to be too conservative of a constraint for haptic device control ([10], [14]).

In this work, we propose a new type of design based on a parallel continuum robot. A set of deformable beams transmit forces from the actuator to the end-effector. The inclusion of deformable components in a kinesthetic haptic device has been proposed previously for a variety of potential benefits. Using deformable elements can be thought of as offloading some of the stiffness rendering from the actuation to the device structure itself. Series elasticity, in which an elastic spring element is put between the end-effector and the actuator, has been the primary approach to using deformation in haptic devices. In [5], the authors used a series elastic actuator for haptic feedback to improve the free-space rendering capabilities of a haptic device at high frequencies. The high-frequency response of a standard device is dominated by the open-loop dynamics of the system, in particular the high mass of the actuators (which grows with the square of the frequency). The spring between the user and the components of the system that are high mass (in particular, the actuators) results in a rendered high-frequency impedance that is dominated by the stiffness of the series elastic component. Thus, by including the series elastic element, the user feels the impedance of the spring only, and not the mass of the actuators at high frequencies.

For 1-DOF series elastic haptic devices, which have been studied in [15], [16], [17], the maximum stiffness that can be rendered stably in an impedance-type control scheme is the intrinsic stiffness of the series elastic component. Stiffness control for a 6-DOF deformable parallel robot based on a Cosserat rod model was demonstrated in [18]. As in that work, we use a model of the device to actively compensate for the deformation of the device to render stiffness higher and lower than the inherent device stiffness.

In addition to these control benefits, deformable components have other advantages. Elastic elements can be used as a means of force-sensing, since the deformation of an elastic element corresponds to the forces that are applied to it ([4]). Acting as a low-pass filter, elastic components can shield expensive motors and gears from damage by shock loads. The use of flexure joints has been suggested as a way to achieve friction-free joints for a haptic device, but the stiffness of the joints must be compensated for ([6]).

Parallel continuum structures have been used before in the design of flexible robots due to their inherent compactness, stability ([19], [20]) and increased payload capacity when compared to non-parallel structures. These characteristics make

this type of configuration a good candidate for the design of very small robotic devices with strict precision requirements ([21]), particularly suitable for minimally invasive surgery applications ([22]) and other small scale manipulation tasks. As in the case of flexible robots in general, modeling and control of parallel continuum robots present some challenges related to their elastic behavior; however, rod theory ([18], [23]) provides a suitable framework to address these challenges.

The deformable design of our device also allows for easier fabrication than typical rigid haptic devices with a similar number of DOFs. In order to reduce the inertia of the haptic device, it is typical to fix the motors in the base of a device, so that the weight of the motors does not contribute to the inertia of the device. For serial devices, the transmission of force from the actuator to the individual joint is achieved via cables routed through the linkages of the device, which requires careful mechanical design and challenging assembly. Parallel devices tend to permit fixed actuation more naturally, but can require complex mechanisms to allow the desired motions (and only those motions) as well as high-quality bearings for smooth motion. Both cable and parallel mechanisms can introduce slop in the device. Compliant mechanisms allow for reduced part count, friction-free motion, and lightweight mechanisms ([24]). Our device is made of continuously deformable beams that can be 3D printed and are the only hardware component required between each actuator and the handle of the device. Compliant mechanisms also enable miniaturization, which has been demonstrated by a 3-DOF device with folding joints ([25]). On the other hand, compliant mechanisms typically undergo significant strain and can suffer from material fatigue. The stiffness obtained at the end-effector of a compliant mechanism is affected by material and structural properties, which limit positioning accuracy and speed of motion. End-effector position sensing, which is important for haptic control, is usually achieved for rigid devices via encoders that measure the joint angles, but sensing for deformable devices is more challenging. Deformable robots typically use either contactless external sensing (either optical or electromagnetic) or flexible embedded sensors ([26]). Sensors embedded in the structure must be able to withstand large deformations and potentially stretch, depending on the application. Sensors have been proposed based on optical waveguides ([27]) and liquid conductors ([28], [29]). In this work, we use flex sensors to sense the position of the end-effector of the device based on a model that maps the deformation of the sensors to the position of the end-effector. Flex sensors have previously been used to perform feedback control of soft bending robots (e.g., [30]) and are appealing for their commercial availability.

III. Device Design

The 5-DOF haptic device, shown in Figure 1, consists of six flexible beams connected to a rigid handle. The base of each beam is attached to a capstan drive which rotates the most proximal node of the beam. This proximal node is rigidly constrained in all directions except that of the actuated rotation. The distal end of each beam is rigidly attached to the handle which the user holds to interact with the virtual

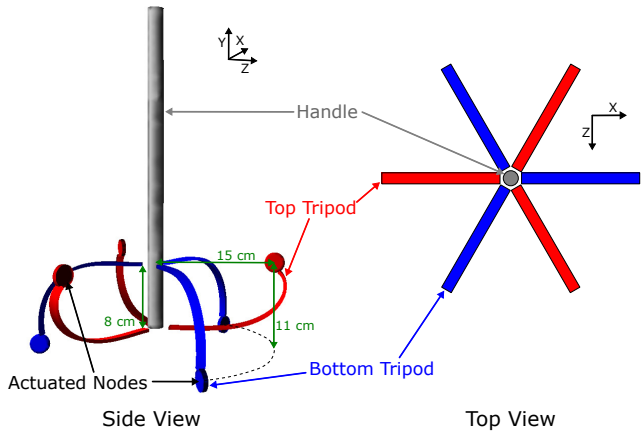


Fig. 2. The device consists of two flexible beam tripod structures attached to the handle and the rigid frame. This schematic shows the bottom (blue) and top (red) tripods as well as the coordinate frame that will be used in the rest of the paper. The parameters shown in the side view, consisting of the distance between handle attachment points, radius to actuated node, and vertical distance between the actuated nodes, are easily adjustable using the device frame and could be used for design optimization. The top view shows how the beams are distributed around the handle.

environment. The six beams are arranged into two tripod-like structures, as shown in Figure 2. The motors of the bottom tripod structure (blue) are affixed to the bottom platform of the frame. The beams attached to these bottom motors are attached above the interaction point of the handle. The top tripod structure (red) is inverted, with the motors attached to the top platform of the frame, and the beams attached below the interaction point of the handle. The top tripod structure is rotated 60 degrees relative to the bottom tripod structure so there is a beam every 60 degrees, alternating attachment above and below the interaction point.

Each beam has a 10 mm by 4 mm cross section and is 3D printed using Ninjaflex material (NinjaTek). The actuated nodes were separated by a height of 11 cm, and the radius to the actuated nodes was 15 cm. The beams were attached 4 cm above and below the end effector. Since the frame is highly adjustable, these spacing parameters could be varied in future work to change the pre-load of the beams and the dynamics of the system. The rest shape of the flexible beam is shown in Figure 3. The rest shapes of the beams themselves could also be modified to achieve different device properties.

A resistive flex sensor (Spectra Symbol, 3.75 in. active length) was attached to each beam by custom clips. Because the flex sensor was inextensible and was not placed along the neutral axis of the beam, only the proximal end of the sensor was firmly attached to the beam. The other clips were used to constrain the sensor along the beam while still permitting the sensor to slide along the beam as it moved. The flex sensors were placed in the region expected to experience the largest variation of bending, but they were not precisely placed. Instead, the sensor fitting procedure described in Section VI allowed for variability in sensor position across the beams.

The proximal node of each beam was rotated by a motor

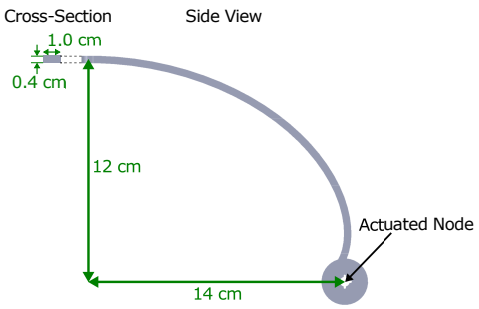


Fig. 3. The rest shape of the flexible beam and its dimensions, including its cross-section.

and capstan drive with a capstan ratio of 6.42. A custom motor controller board (Microchip dsPIC33 digital signal controller) controlled the six brushed DC motors (Mabuchi RS-455PA) in position at 1 kHz. The motor position controller was designed to have a settling time of 0.022 seconds. Encoders on the motors allowed for motor position sensing of 0.1 degrees. Communication between the motor controller board and the computer was managed via a CAN bus. The rate for the high-rate control loop, which was implemented in SOFA ([1]), was 500 Hz.

Reflective trackers were attached to points on the top half of the handle to collect position and orientation data of the handle during movements using an OptiTrack motion capture system (NaturalPoint). This data is used for evaluation, not for sensor fitting or within the control loop.

The handle is controlled in five DOFs, three translations and two rotations. The rotation in the direction of the handle is not controlled, and we assume that the user does not apply torques in that direction. Because we use six motors to control five DOFs, the device is redundant. In order to choose between the infinitely-many possible configurations that result in a particular end-effector position, the energy of deformation of the actuator nodes is minimized. This provides a unique solution to the inverse position control problem. Due to its structure, the device is very stiff in that uncontrolled direction, so the workspace is relatively narrow in the uncontrolled direction. This can also be understood by considering the two tripods separately. Each tripod can control three translational DOFs, but the two tripods are coupled to the same rigid body, in particular with a fixed distance between them. This constraint effectively removes one of the DOFs.

IV. Mechanical Modeling

In this section we present the mechanical model used in the control algorithm. This is based on previous work by [31], which is implemented in SOFA ([1]). SOFA admits a variety of different material models and discretizations. First we will describe the components used to model the mechanics of this specific device and then we will discuss the general constraint-based formulation for solving the model.

A. FEM beam model

The design of this soft parallel robot is essentially 6 curved deformable rods arranged in a simple lattice form: each rod connects one actuator to the handle. This kind of deformable structure is commonly modeled in numerical mechanics using a beam assembly. Given the workspace of the robot, we have opted for a non-linear formulation compatible with large displacements (the model takes into account the geometrical non-linearities related to the deformation, but the behavior law of the material remains linear). This beam model is based on that described in [32] and [33]. This type of modeling based on nonlinear beams has already been used for a continuum robot model in [34].

Cosserat models would be another candidate for modeling this device. In most Cosserat models used in robotics, reduced coordinates (often based on local strain) are used, like in articulated structures. These reduced coordinates are particularly adapted to linear structures. Then, for parallel structures, like in rigid parallel robots, a technique for closing the kinematic chains is employed (like Lagrange multipliers). With assembled FEM beams, the coordinates are the frames at the extremity of the beams. These coordinates are absolute and the only difference between linear or parallel structures for the robot will be the profile of the assembling matrix (band matrix for linear structures, sparse matrix for parallel structures). Of course the number of coordinates in the model is generally higher with FEM, but very efficient sparse matrix solvers exist which allow for a real-time computation. This feature is important in our case since our control strategy is based on a real-time simulation of the model. The continuum model of each arm of the haptic device is based on beam elements, with each node connecting the beam elements represented as a 6-DOF frame. Each beam represents a small part of the structure and is parameterized by 2 frames $\mathbf{X}_1, \mathbf{X}_2 \in SE(3)$. Each of the the 6 curved rods of the robot are discretized using 14 beam elements. The distribution of nodes in one of the curved rods is shown in Figure 6.

A linear stress-strain relationship is assumed (implicitly assuming a small material deformation), but the model allows for large displacements by updating the rotation matrix used to map the element stiffness matrix to the global stiffness matrix at each time step. The local frame is defined by the first node of the beam element. The mass and stiffness matrices are derived from continuum mechanics under the assumption of straight beam elements, so we distribute more nodes and elements in the part of the beam that is more highly curved so that this approximation is more accurate.

The local element mass and stiffness matrices are defined in [33] (pages 79 and 294 for stiffness and mass, respectively). Local element mechanics are then combined in the global mass and stiffness matrices based on a transformation matrix between each element and the global frame. The constraints on the structure occur at each attachment point to the base. While the device is free to rotate around these points in the direction of the motor axis, all other directions of this node are fixed. The beams are rigidly attached to the handle (there are no joints that allow for any rotations).

By applying the assembly method used in finite elements, we generate a mechanical model in which position vector $\mathbf{q} = \{\dots \mathbf{X}_i \dots\}$ gathers all the node frames (84 nodes) and the frame of the handle (the dimension of $\mathbf{q}$ is 510).

The beams were modeled with an elastic modulus of 76.4 MPa and a mass density of 1040 kg/m^3. This mechanical model could be improved by more carefully measuring the stiffness and mass properties resulting from the particular material and 3D printing process. The added stiffness and mass due to the flex sensor was not modeled, and the mass of the handle was modeled as a point mass of 8 g.

B. Constraint-based formulation

The generalized coordinates of the nodes used to simulate the device are defined as $\mathbf{q}$. Since the mass of the robot is small compared to the stiffness and to simplify the modeling, we have opted for a quasi-static approach. In static equilibrium, the external and internal forces are in equilibrium for each node of the mesh:

$$\mathbf{0} = \mathbf{M}\mathbf{g} - \mathbf{f}(\mathbf{q}) + \mathbf{H}^T\boldsymbol{\lambda} \tag{1}$$

where $\mathbf{M}$ is the mass matrix used to compute the gravity forces, $\mathbf{f}(\mathbf{q})$ is the vector of internal forces due to the stiffness of the material (dependent on the configuration), and $\mathbf{H}^T(\mathbf{q})\boldsymbol{\lambda}$ is the force due to constraints on the device, which include actuators and the end-effector. The $\mathbf{H}^T$ matrix defines the directions of actuator or end-effector forces. It is a sparse matrix with non-null values only on nodes of the mesh where those forces apply. $\boldsymbol{\lambda}$ defines their magnitudes. It is assumed that $\mathbf{H}$ is constant during a time step, and this direction, which may be dependent on configuration, is recomputed at each time step.

To understand the kinematics, let's consider a slight change of the force $\boldsymbol{\lambda}$ by $d\boldsymbol{\lambda}$. This will create a small motion $d\mathbf{q}$ of the FEM mesh nodes. Let's consider the difference between two static equilibria described in equation (1), one taken at position $\mathbf{q} + d\mathbf{q}$ and one at position $\mathbf{q}$:

$$\mathbf{f}(\mathbf{q} + d\mathbf{q}) - \mathbf{f}(\mathbf{q}) \approx \underbrace{\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}}_{\mathbf{K}(\mathbf{q})} d\mathbf{q} = \mathbf{H}^T d\boldsymbol{\lambda}. \tag{2}$$

We introduce the dual variables to $\boldsymbol{\lambda}$, named $\boldsymbol{\delta}$, which correspond to the movements in the same space. For our robot, around a static equilibrium point of the model, $d\boldsymbol{\delta}$ represents an angular displacement of the motors and the translations and rotations of the end-effector:

$$d\boldsymbol{\delta} = \boldsymbol{\delta}(\mathbf{q} + d\mathbf{q}) - \boldsymbol{\delta}(\mathbf{q}). \tag{3}$$

Around this equilibrium point, the torques exerted by the actuators and the forces and torques on the end effector must follow the principle of virtual work:

$$d\mathbf{q}^T\mathbf{H}^T d\boldsymbol{\lambda} = d\boldsymbol{\delta}^T d\boldsymbol{\lambda}, \tag{4}$$

which leads to $\mathbf{H} = \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{q}}$, and

$$d\boldsymbol{\delta} = \mathbf{H}d\mathbf{q} = \mathbf{H}\mathbf{K}(\mathbf{q})^{-1}\mathbf{H}^T d\boldsymbol{\lambda}. \tag{5}$$

Equation (5) provides the linearization around a current position $\mathbf{q}$ of the static behavior of the structure, condensed in the constraint space (composed of actuator and effector spaces). As the constraints are coupled by these non-linear equations, their position in the constraint space is also linearized in the solving process:

$$\boldsymbol{\delta} = \mathbf{H}\mathbf{K}(\mathbf{q})^{-1}\mathbf{H}^T d\boldsymbol{\lambda} + \boldsymbol{\delta}_0(\mathbf{q}) \tag{6}$$

with $\boldsymbol{\delta}_0(\mathbf{q})$ that represents the constraint position if the constraint force does not change over the time step ($d\boldsymbol{\lambda} = 0$). Then the effect of a force in the constraint direction j on the constraint direction i can be written as $\mathbf{W}_{ij}$ and is computed as follows:

$$\mathbf{W}_{ij} = \mathbf{H}_i\mathbf{K}(\mathbf{q})^{-1}\mathbf{H}_j^T. \tag{7}$$

This matrix is homogeneous to compliance, the inverse of stiffness. In [35] it has been shown that one can obtain the equivalent to a Jacobian matrix $\mathbf{J}$ for a deformable robot, that would describe the effect of a motion of the actuators $d\boldsymbol{\delta}_a$ on the motion of the effectors $d\boldsymbol{\delta}_e$:

$$d\boldsymbol{\delta}_e = \mathbf{W}_{ea}\mathbf{W}_{aa}^{-1} d\boldsymbol{\delta}_a = \mathbf{J} d\boldsymbol{\delta}_a. \tag{8}$$

Equations (5) and (8) provide small systems of condensed equations to describe the motion of the robot around the current position $\mathbf{q}$ of the robot and have been detailed in previous papers, in particular to compute a control based on inverse kinematic model of the deformable robots [31]. In these inverse kinematic models, a quadratic program (QP) is used to solve for the position or force of the actuator required to achieve a goal position of the end effector.

The novelties concerning the modeling introduced in this paper are the following:

- We will use this condensed model in an optimization function to compute the workspace of the robot as a haptic interface, including calculating the maximum isotropic force available for haptic rendering at any point in the space.
- We will expand the constraint space to obtain a condensed relationship between the measurements of the bending sensors, the motor encoders, and the position of the effector.
- We will assume that the deformations are slow enough so that the update of the $\mathbf{K}(\mathbf{q})$ and $\mathbf{H}(\mathbf{q})$ matrices can be done on the basis of a computation of the $\mathbf{q}$ positions at around 20 Hz (solution of (1)). Based on this assumption, we rely on the condensed models to update the effector displacements (configuration estimation) and the motor control (inverse kinematic model) at high frequencies to meet the needs of haptics.

V. Workspace Analysis

In this section, we compute the workspace of the device and develop a means of evaluating the force output of the device throughout that workspace. The actuators may be limited in both force and displacement. For this particular device design, displacement constraints on the actuators come into effect due to the capstan drives, while motor torque was not a limiting factor. However, as the device stiffness increases, the force capabilities of the actuators will overtake the displacement limits as the most limiting constraints on workspace and force

output, since more actuator force will be required to overcome the inherent device stiffness.

In rigid robots the reachable workspace is constrained only by the displacement of the actuators. The static force output at each point in that workspace is constrained only by the torque limits of the actuators (ignoring gravity compensation). In contrast, for a deformable device, the force and displacement of both the actuator and end-effector are coupled. This means that the workspace and the static force output are constrained by *both* the torque and displacement constraints on the actuation. This adds constraints to both workspace and force computations, relative to what is needed for a rigid device. Additionally, the device is redundant in that six motors are used to control five end-effector DOFs (three translation and two rotation).

A. Implementation

First, we define the reachable workspace as the set of configurations the robot can reach with no force on the end-effector. We evaluate the static workspace of the device by sampling the space. Using the inverse solver in simulation, we test whether each goal point can be reached, given a certain tolerance. If the simulated end-effector of the robot reaches the goal point within the tolerance, that point is considered reachable, and we evaluate the maximum isotropic static force and torque that can be output at that point using a polytope approach similar to that described in [36].

The maximum isotropic static force is the maximum force that can be applied in every direction while maintaining the current position. Due to their difference in dimension, force and moment results are often presented separately. Yoshikawa provides definitions of two force performance metrics for rigid robots ([37]). For force performance in the *strong sense*, potential force outputs are considered assuming that the moments are constrained to be zero. Similarly, for moment performance in the *strong sense*, we consider only potential moment outputs when the force output is constrained to 0. For force (moment) performance in the *weak sense*, in contrast, we consider all possible output forces (moments) given the actuation constraints, including those that also produce moments (forces) at the end-effector. The performance in the *strong sense* is stricter in that it requires that only forces be produced without also generating moments. For deformable robots, this distinction can be considered by choosing whether the robot must be constrained in position in the directions other than that for which it is being controlled. In the following, we consider the maximum isotropic forces and torques in the *strong sense*, requiring that the end-effector position remains the same in all directions and assuming that no forces/torques are applied in the directions not included.

We are interested in computing the maximum isotropic force output at the end-effector at a given position. We use the reduced compliance formulation to evaluate this. Note that this is a linearization of the mechanics about the configuration with no forces applied at the end-effector. As the actuators apply forces and deform the robot, this linearization will no longer apply, but it will still give a reasonably good idea of the force capabilities of the device and how those vary throughout the workspace.

To determine the maximum isotropic force, we want to solve the problem below.

$$\begin{aligned} \text{maximize} \quad & r \\ s.t. \quad & \forall \, \|\bar{\boldsymbol{\lambda}}_e\|_2 \leq r \; \exists \, d\boldsymbol{\lambda}_a \text{ such that} \\ & 0 = \mathbf{W}_{ee} d\boldsymbol{\lambda}_e + \mathbf{W}_{ea} d\boldsymbol{\lambda}_a + \boldsymbol{\delta}_{e,0} \\ & \boldsymbol{\delta}_a = \mathbf{W}_{ae} d\boldsymbol{\lambda}_e + \mathbf{W}_{aa} d\boldsymbol{\lambda}_a + \boldsymbol{\delta}_{a,0} \\ & \boldsymbol{\lambda}_{a,min} \leq d\boldsymbol{\lambda}_a + \boldsymbol{\lambda}_{a,0} \leq \boldsymbol{\lambda}_{a,max} \\ & \boldsymbol{\delta}_{a,min} \leq \boldsymbol{\delta}_a \leq \boldsymbol{\delta}_{a,max} \end{aligned} \tag{9}$$

Here, $\bar{\boldsymbol{\lambda}}_e = -d\boldsymbol{\lambda}_e - \boldsymbol{\lambda}_{e,0}$ is the force applied by the robot on the user who holds the effector ($\boldsymbol{\lambda}_e$ is the force applied by the user on the robot), and $\boldsymbol{\lambda}_a = d\boldsymbol{\lambda}_a + \boldsymbol{\lambda}_{a,0}$ represent forces applied by the actuators on the robot. $\boldsymbol{\lambda}_{e,0}$ and $\boldsymbol{\lambda}_{a,0}$ represent the forces on the corresponding constraints at the beginning of the step used for the linearization.

The first constraint equation in Equation 9 constrains the movement of the end-effector relative to the goal point. If only position (or orientation) is to be constrained, then only those DOFs need be included. For our analyses, we constrain all directions. The second constraint equation determines the displacement of the actuator due to both actuator and end-effector forces. The two sets of inequality constraints constrain the torques ($\boldsymbol{\lambda}_a$) and displacements ($\boldsymbol{\delta}_a$) of the actuators.

To simplify the problem and clarify its structure we can rearrange it by eliminating $\boldsymbol{\delta}_a$ and solving for $\bar{\boldsymbol{\lambda}}_e$ using the equality constraints. Then the problem reduces to:

$$\begin{aligned} \text{maximize} \quad & r \\ s.t. \quad & \forall \, \|\bar{\boldsymbol{\lambda}}_e\|_2 \leq r \; \exists \, d\boldsymbol{\lambda}_a \text{ such that} \\ & d\boldsymbol{\lambda}_{a,min} \leq d\boldsymbol{\lambda}_a \leq d\boldsymbol{\lambda}_{a,max} \\ & d\boldsymbol{\delta}^*_{a,min} \leq \mathbf{W}^*_a d\boldsymbol{\lambda}_a \leq d\boldsymbol{\delta}^*_{a,max} \\ & \bar{\boldsymbol{\lambda}}_e = \mathbf{W}_{ee}{}^{-1}(\mathbf{W}_{ea} d\boldsymbol{\lambda}_a + \boldsymbol{\delta}_{e,0}) - \boldsymbol{\lambda}_{e,0} \end{aligned} \tag{10}$$

with

$$\begin{aligned} & d\boldsymbol{\lambda}_{a,min} = \boldsymbol{\lambda}_{a,min} - \boldsymbol{\lambda}_{a,0} \\ & d\boldsymbol{\lambda}_{a,max} = \boldsymbol{\lambda}_{a,max} - \boldsymbol{\lambda}_{a,0} \\ & \mathbf{W}^*_a = \mathbf{W}_{aa} - \mathbf{W}_{ae}\mathbf{W}_{ee}{}^{-1}\mathbf{W}_{ea} \\ & d\boldsymbol{\delta}^*_{a,min} = \boldsymbol{\delta}_{a,min} - \boldsymbol{\delta}_{a,0} + \mathbf{W}_{ae}\mathbf{W}_{ee}{}^{-1}\boldsymbol{\delta}_{e,0} \\ & d\boldsymbol{\delta}^*_{a,max} = \boldsymbol{\delta}_{a,max} - \boldsymbol{\delta}_{a,0} + \mathbf{W}_{ae}\mathbf{W}_{ee}{}^{-1}\boldsymbol{\delta}_{e,0} \end{aligned} \tag{11}$$

The matrix $\mathbf{W}^*_a$ can be thought of as the total compliance of the actuator, taking into account the rigid constraint we impose on the end-effector. $\mathbf{W}_{ee}$ is positive definite and therefore invertible. The constraints in Equation 10 represent a projected polytope. The inequality constraints define a polytope in the actuator space within which the actuators can operate. This optimization problem is then projected into the end-effector space. With a redundant robot, the number of actuators exceed the number of end-effectors, so the projection equation represents a reduction in dimension as well. In order to project the polytope into the lower dimensional end-effector space, we first convert the system of inequalities (or half-spaces, also known as the halfspace representation or H-rep) into a vertices representation (V-rep) (using cddlib and pycddlib,

([38], [39])). We can then apply the linear transformation to the vertices. The projected polytope will then be the convex hull of the these transformed vertices, and this hull can be represented as a system of inequalities (again using the double description method implemented in the libraries above). Then we have a system of inequalities that define the possible end-effector forces, $\bar{\boldsymbol{\lambda}}_e$. In order to find the most limiting end-effector force direction, we can test each of these inequalities to determine the distance between the plane it defines and the origin. The minimum distance between the constraints and the origin defines the maximum magnitude force that can be applied by the end-effector on a user or the environment in all directions. Since we limit our analysis to reachable points, we are assured that the origin is within the constrained set.

B. Workspace Results

Results for the translational workspace are shown in Figure 4. These plots show results for two material stiffnesses: one is the current device stiffness and the other is less stiff by a factor of 5. Three results are shown for each device stiffness, one with unlimited actuator torques, which was assumed in our rendering, and two other actuator torque limits, 0.04 Nm and 0.01 Nm. From these results, we can see that higher forces are possible with a stiffer device when there are no actuator torque constraints to consider. Without actuator torque constraints, the limiting factor in force output is the permitted displacement of the actuators. For a stiffer device, it is possible to generate more force within a given displacement. Since the displacement constraints were the same for all tests shown, a higher force output is possible with the stiffer device.

The workspaces of the two devices with different stiffnesses are similar when actuator torques are not constrained, but these workspaces are not identical, due to the impact of gravity. Both devices were modeled as having the same mass. The workspace of the stiffer device is more closely centered around the zero vertical position, since the stiffer device is less impacted by the force of gravity. As actuator torque constraints are introduced, the workspace of the stiffer device decreases dramatically in size. Further, the possible force output within that workspace decreases for both devices (it is more noticeable in the plots for the stiffer device due to the force scale). When the actuator torques are severely restricted, the stiff device workspace becomes extremely small, since the actuators cannot overcome the inherent parallel stiffness of the device.

The workspace and the maximum isotropic torque for the two rotational DOFs is shown in Figure 5. These consider rotations that are reachable while the translational DOFs are constrained to the origin. A higher stiffness allows for higher torque output given the same constraints, but the workspace for a stiff device shrinks quickly with actuator torque constraints. In the case without actuator torque limits, the stiffer device has a larger workspace because it is less affected by gravity (the actuators do not have to move as much to compensate for gravity to keep the device at $y = 0$).

The actuator torque limits shown in these plots are fairly aggressive (more limiting than the current actuator and capstan combination, which were limited to 0.3 Nm) in order to demonstrate the impact of increasing stiffness.

VI. Deformation Sensing

The deformable nature of the device means that sensing of the actuator position does not lead directly to knowledge of either the end-effector position or force, which is important for either an impedance- or admittance-type haptic rendering algorithm. It is possible to infer the position and force of the end-effector using the mechanical model and knowledge of both the control force and position of the actuation. However, this relies heavily on having precise parameters for the mechanical model. In that case, the maximum stiffness that can be rendered is limited to the inherent stiffness of the device when the motors are fixed ([7]), since the actuator cannot sense motion at the end-effector until that motion has been transmitted through the stiffness of the device. The actuation cannot compensate for this deflection in advance. Adding sensing of the end-effector position allows us to compensate for this deflection, allowing us to achieve higher stiffness than is possible with only actuator sensing. By incorporating additional position sensing in the device, we can achieve a more robust stiffness rendering since we are less reliant on the mechanical model. In this section, we describe the calibration procedure for the sensors so that they can be integrated in the model for end-effector force and position sensing.

A flex sensor is attached to each of the deformable beams, as shown in Figure 6. The flex sensor is rigidly attached at its base and passes through guides that maintain it in contact with the beam while allowing it to slide relative to the beam as the beam deforms from its rest position. The sensor remains in contact with the beam throughout the deformation.

In order to use the sensors as constraints for solving the inverse sensing problem, we need to determine a function that maps sensor values to the DOFs of the beam. The sensors have different electrical offsets and are not carefully calibrated, so there is no a priori knowledge of the function between the deformation of the sensor and the signal it produces, other than that the signal should generally increase as the curvature of the beam increases. We fit a linear model of the relative beam node angles by first collecting data using the device and a direct simulation and then fitting the model using optimization.

A. Data Collection and Pre-Processing

To fit a function of the beam node DOFs to the sensor values, we first actuate the device through a sequence of motor positions. We define a minimum and maximum motor angle, and develop a trajectory that visits every combination of maximum and minimum positions in a random order, linearly interpolating between points. We simultaneously apply these motor trajectories (time sequence of motor positions) to the physical device and to a direct simulation of the device. The direct simulation of the device uses the commanded motor position as a constraint and integrates the rest of the device nodes according to the mechanics equations. From the device, we record the voltage signal of each sensor (as measured via a 12 bit ADC). From the simulation, we record the orientation

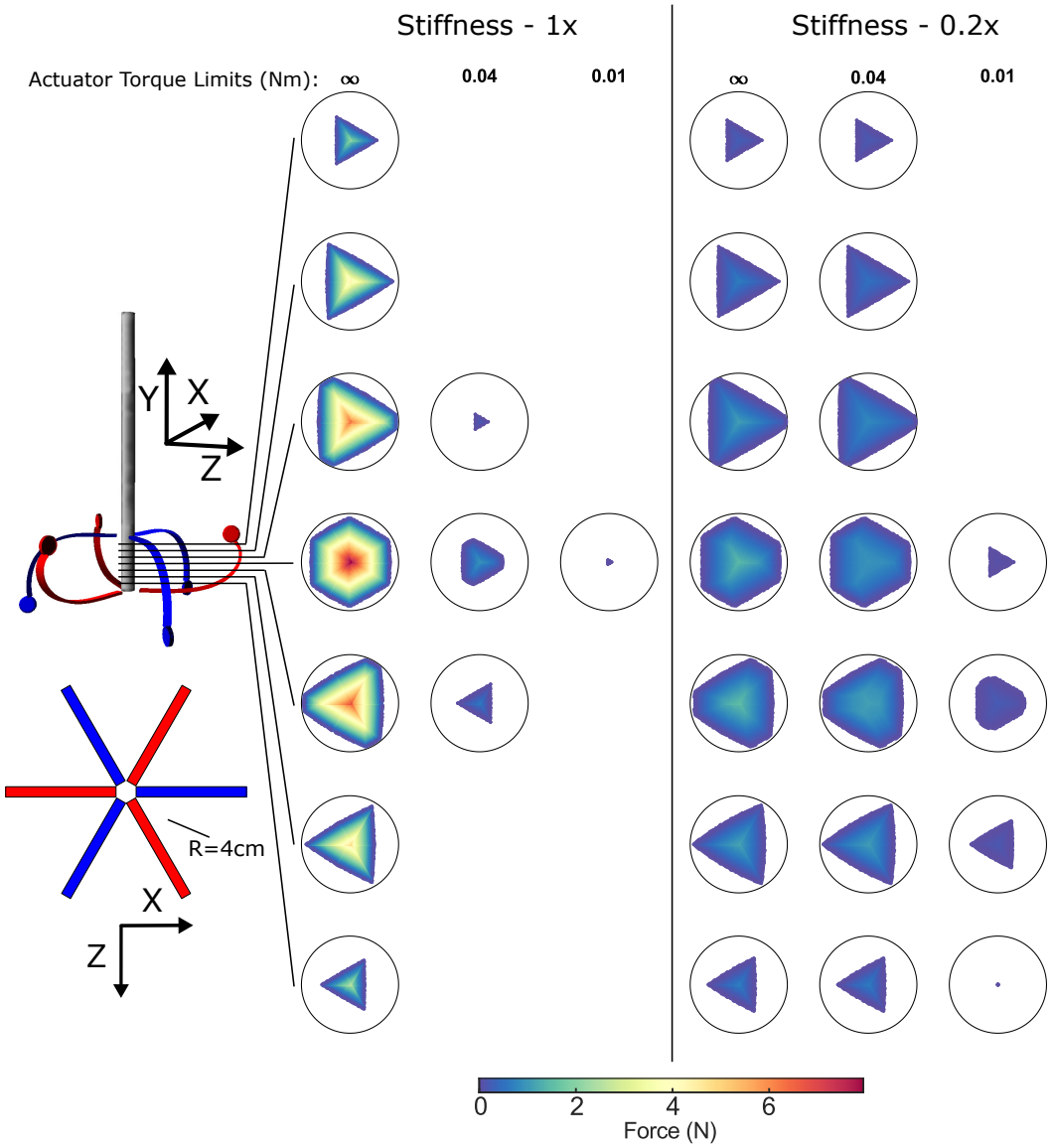


Fig. 4. The maximum computed isotropic force in the strong sense (for the x, y, and z directions) is shown throughout the translational workspace under different actuation torque limits. One stiffness (1x) uses the Young's modulus from the actual device. The other (0.2x) has a Young's modulus five times less stiff. The actuation displacement limits are constant. The workspace is represented with horizontal slices, spaced 1 cm apart vertically. Reachable points in the workspace are colored with their maximum isotropic force. Each black circle has a radius of 4 cm. $\theta_x = \theta_z = 0$ for all positions. Stiffer beams allow for higher forces but reduced workspace under torque constraints, while the workspace is more influenced by gravity for less stiff beams.

of the nodes of the beam in the region containing the sensor, as well as the orientation of the node directly actuated by the motor (see selected nodes in Figure 6).

We assume that the deformations caused by the motor trajectories will be similar to those caused by a combination of motor torques and end-effector forces and torques applied by the user. This is reasonable because of the parallel structure of the device. In particular, a force (or displacement) applied by one motor to the structure will lead to deformation in the other beams as well because of their coupling through the rigid handle. This is in contrast to a series elastic device, in which static forces or displacements from the actuation alone will not lead to deformation of the elastic element. Since we are relying on the parallel structure to transmit displacement and produce a wide range of beam deformations in order to calibrate the device, it is important to directly command the motor positions, rather than specifying end-effector positions and using the inverse simulation to control the device because we specifically want to produce deformation in the beams. The inverse position rendering algorithm minimizes the actuator energy assuming no forces are applied to the end-effector, which will lead to less deformation of the beams, even if the same end-effector positions were reached. In other words, it is important that the motors at times work against each other so that larger forces are applied to the beams, leading to larger deformations to mimic those that occur when a user applies forces to the end-effector.

We also assume that the time response of the sensor to

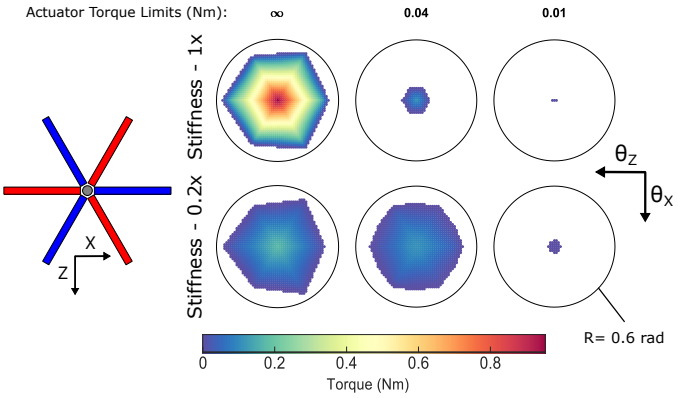


Fig. 5. The maximum computed isotropic torque in the strong sense (for the θ_x and θ_z directions) is shown throughout the rotational workspace for two different device stiffnesses under different actuation torque limits. One stiffness (1x) uses the Young's modulus from the actual device. The other (0.2x) has a Young's modulus one fifth of that value. The actuation displacement limits are held constant. Reachable angles in the workspace are colored with their maximum isotropic torque. Each black circle has a radius of 0.6 rad. $x = y = z = 0$ for all positions.

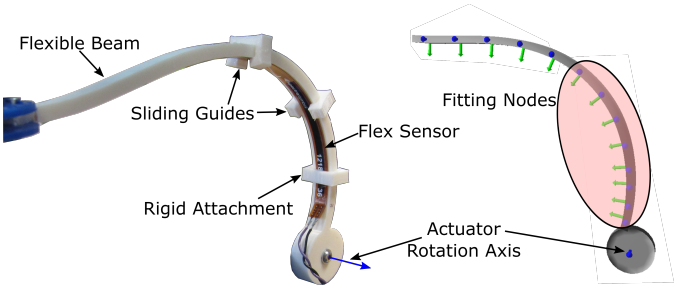


Fig. 6. A flex sensor is attached to each beam via clips (left) which either rigidly attach the sensor (for the end nearest the actuated node) or constrain it to slide along the beam as the beam bends. The highlighted nodes of the direct model (right) were used for fitting the flex sensor values to the beam deformation. These nodes include the active length of the flex sensor.

changes in shape is sufficiently fast that the electrical dynamics of the sensor can be ignored with minimal error. Thus, we assume that the sensor gives a true measurement of shape, with no time-dependent characteristics. The sensor value is assumed to be a function only of node orientation and not velocity or higher order terms.

In order to define a function that predicts the sensor value based on node orientations, we first define a set of basis functions that will be candidates for our fitting procedure. We limit these candidate basis functions to those for which we can define the derivative with respect to the node positions, since this will be essential for the projection of the mechanics matrix into sensor space (computing the $\mathbf{H}$ matrix). We are interested in sensing the primary bending direction of the beam, both because we expect this to have the most deformation and because this is the primary sensing direction of the flex sensor (as opposed to twisting or other rotations). To this end, we compute the displacement of the rigid frame of each node in the coordinate frame of the actuator node corresponding to that node's beam. The actuator node is the node rotated by the motor via the capstan. We then take the component of that displacement in the direction of the actuator node rotation axis. We compute this function at every time step for each node. Thus, for each beam, we obtain a time series of this node rotation data, which will be written as:

$$^{\text{time}}\alpha_{\text{nodeIndex}} \tag{12}$$

B. Sensor Function Fitting

Using the data collected from the simulation and sensors during the motor trajectory phase, we determine the coefficients for a linear function of the node rotations computed in the pre-processing phase, α_n, to sensor values, v, via a series of optimizations. We use ℓ_1-norm regularization to encourage sparsity and to select the most relevant nodes to use in the fit sensor function. This regularization helps to avoid overfitting the data, which reduces sensitivity to noise in the calibration phase.

For each beam, we construct the following matrix and vectors with n being the number of nodes observed and m being the number of observations (time steps) in the calibration,

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 1 & {}^0\alpha_1 & {}^0\alpha_2 & \cdots & {}^0\alpha_n \\ 1 & {}^1\alpha_1 & {}^1\alpha_2 & \cdots & {}^1\alpha_n \\ & & \vdots & & \\ 1 & {}^m\alpha_1 & {}^m\alpha_2 & \cdots & {}^m\alpha_n \end{bmatrix} \\ \mathbf{v^T} &= \begin{bmatrix} {}^0v & {}^1v & \cdots & {}^mv \end{bmatrix} \\ \mathbf{c^T} &= \begin{bmatrix} c_0 & c_1 & \cdots & c_n \end{bmatrix}. \end{aligned} \tag{13}$$

Matrix $\mathbf{A}$ consists of the node rotations from the direct simulation for each node at each time point in the direction of the actuator rotation axis for that beam, computed as described in the previous section. A constant term is also included to account for the constant offset of the sensor. Vector $\mathbf{v}$ is the vector of measured sensor values over time, and $\mathbf{c}$ is the vector of coefficients to be fit. Note that $m \gg n$.

We perform the following optimization using CVXPY ([40], [41]).

$$\min_{\mathbf{c}} \quad \|\mathbf{Ac} - \mathbf{v}\|_2 + \gamma\|\mathbf{c}\|_1. \tag{14}$$

The first term minimizes the error between the linear model and the sensed values. The second term encourages sparsity by putting a cost on each coefficient. From this optimization, we obtain a sparsity pattern for the linear fit, essentially selecting which nodes to use. We then repeat the optimization using only those selected nodes (denoted with a bar) and without the regularization term:

$$\min_{\bar{\mathbf{c}}} \quad \|\bar{\mathbf{A}}\bar{\mathbf{c}} - \mathbf{v}\|_2. \tag{15}$$

We repeat this process over a range of γ values and select the best fit with three or fewer coefficients (including the constant coefficient) as the constraint based on this fit was found to have the best behavior in the inverse simulation.

An example fit for one of the sensors is shown in Figure 7.

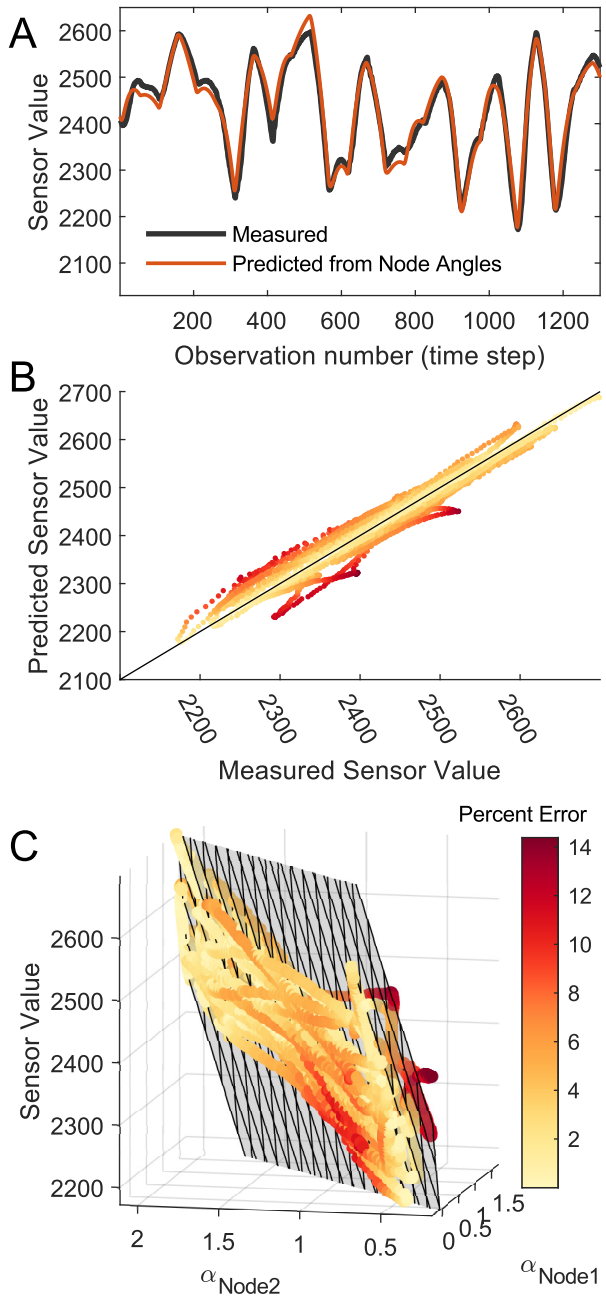


Fig. 7. The results of the sensor function fitting procedure are shown for a single sensor. A. The measured sensor value (12 bits raw ADC output) and the predicted sensor value from the simulated node angles are shown over part of the sensor calibration sequence as the motors are moved through their trajectories. B. The predicted sensor values match the measured sensor values well over the range of sensor values. Colors correspond to the error of the prediction from the measured value as a percentage of the total range of the sensor. C. The sensor value as a function of the node values. The two rotation functions (rotation in the direction of the actuator rotation, α_{Node1} and α_{Node2}) and a constant offset are used to fit the sensor values. The particular nodes used for the different sensors vary and are chosen by the fitting optimization. The fit function is shown as the grey plane. Each time step serves as an observation, and the measured sensor value is plotted.

C. Sensor Constraint for Modeling

To use the mechanical modeling scheme presented in Section IV-B, in which the mechanics are projected into the constraint space, we need to determine the projection matrix $\mathbf{H} = \frac{\partial \boldsymbol{\delta}_i}{\partial \mathbf{q}}$ for the sensor function relative to the global coordinates. Since the sensor function is the linear combination of particular DOFs transformed into the actuator frame, we can use the adjoint transformation matrix between the actuator node frame (a) and the global frame (g), ${}^g\mathbf{T}^a$, to define $\mathbf{H}$,

$$\mathbf{H} = \sum_{i}^{\text{nodes}} {}^g\mathbf{T}^a [0, 0, 0, 0, 0, c_i]^T. \tag{16}$$

The vector in the equation shows that we only consider the DOF corresponding to the z rotation direction, with c_i being the coefficient for the ith node. As with the beam mechanics, the $\mathbf{H}$ matrix is constructed first in local coordinates and then transformed to global coordinates.

In general, since the reference frame that we are measuring the orientation in is moving (since it is the actuated node) we would need to consider how the rotation of the actuator changes due to the constraint forces as well. Since we are only considering the rotation with respect to the rotation axis which is considered to be rigidly fixed, though, we do not need to account for this variation.

D. Discussion

From the plot in Figure 7, we see that the node rotations used in matrix A to fit the sensor function are highly correlated, since they move in a relatively narrow band. This makes sense because it is likely that both of them experience rotations due to their actuator. The sensor function is a linear combination of these two node rotations, and this correlation shows why it is important to consider a function of multiple nodes, rather than a single node, in order to detect bending and not just rotation due to the actuator on that beam.

The sensor fit is subject to a few sources of error. The mechanical mounting of the sensor relies on the sensor being able to freely and consistently slide along the beam, so the beam surface must be low friction and smooth so that the distal end of the sensor does not catch, causing the sensor to buckle away from the beam. The sensor calibration is also dependent on the model providing an accurate representation of the beam deformations. The sensor calibration we perform is open-loop in that we do not correlate the sensor functions that we determine to the actual end-effector positions. We compared the end-effector position predicted in the simulation to that measured by the OptiTrack motion capture system. While not directly measuring the accuracy of the nodes used for sensing, it is reasonable to assume comparable performance in position and orientation data across the device. The forward simulation matched the actual device positions quite well, with mean absolute errors less than 0.15 cm in each translational direction and less than 0.015 radians in each rotational direction.

VII. End-Effector Force and Position Sensing

In this part, we show that we can estimate the position and force at the end effector using the model. This estimation is

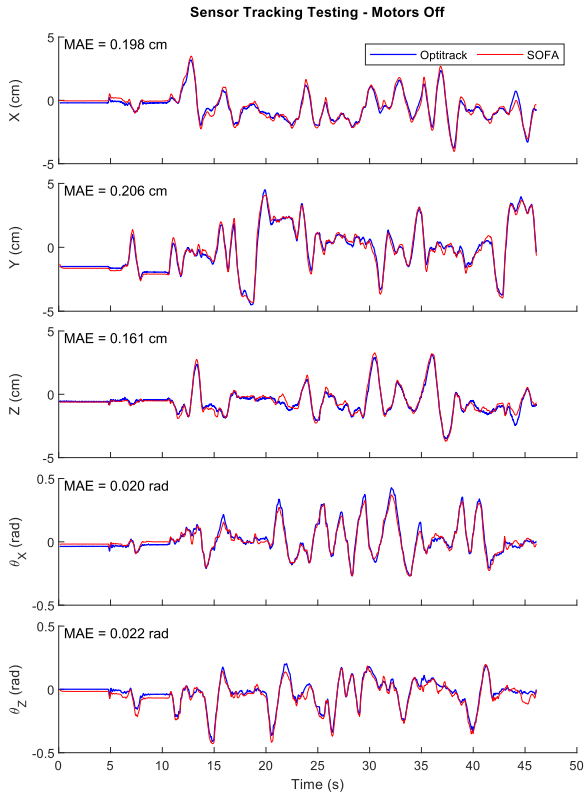


Fig. 8. Sensing using the inverse simulation (SOFA) vs the actual end-effector pose from the OptiTrack system when the motors are off (free to rotate). Both the sensor voltage and the actuator position are used to estimate the position of the end-effector. With the motors off, the device is relatively free to move. MAE is Mean Absolute Error.

based on the bending measurements of the sensors and the angular encoders on the motors and has been validated by comparison to external measurements of position and forces. Furthermore, for the haptic feedback, we need the position measurement of the end effector at high frequency. To do this, we performed a multi-rate approach to estimation. To achieve device control at rates sufficient for haptic rendering, the control loop, and thus the sensing, consists of two threads: one runs at a high rate and interacts with the robot, sensing and providing actuator positions commands at 500 Hz, while the other runs at a relatively low rate (about 20 Hz) and provides the reduced linear model to the high rate thread.

We will use quadratic programs (QPs) to solve particular constraint equations of the form of Equation 6 for both sensing and haptic rendering. The low rate SOFA thread provides the inputs to two high rate QPs, one for sensing and one for rendering. They are linearized about the same configuration of the robot, but there are differences in the $\boldsymbol{\delta}^0$ terms, in particular because the defined goal configurations are different. The compliance matrices, $\mathbf{W}_{ij}$ are defined by the method described in Section IV-B. In this section, we will define the sensing QPs and show results validating the sensed end-effector position and forces.

The sensing constraint equation is as follows:

$$\begin{bmatrix}\boldsymbol{\delta}_e\\ \boldsymbol{\delta}_a\\ \boldsymbol{\delta}_s\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{ee} & \mathbf{W}_{ea}\\ \mathbf{W}_{ae} & \mathbf{W}_{aa}\\ \mathbf{W}_{se} & \mathbf{W}_{sa}\end{bmatrix}\begin{bmatrix}d\boldsymbol{\lambda}_e\\ d\boldsymbol{\lambda}_a\end{bmatrix} + \begin{bmatrix}\boldsymbol{\delta}_e^0\\ \boldsymbol{\delta}_a^0\\ \boldsymbol{\delta}_s^0\end{bmatrix} \tag{17}$$

with

$$\begin{aligned}\boldsymbol{\delta}_e^0 &= \mathbf{x}^0 - \mathbf{x}_{\text{ref}}\\ \boldsymbol{\delta}_a^0 &= \boldsymbol{\theta}^0 - \boldsymbol{\theta}_{encoder}\\ \boldsymbol{\delta}_s^0 &= \mathbf{v}^0 - \mathbf{v}_{sensor}.\end{aligned} \tag{18}$$

Here and throughout this section, e refers to end-effector DOFs (five total, three translation and two rotation), a refers to actuator DOFs (six rotation, one for each motor), and s refers to sensor DOFs (six functions, one for each beam-mounted flex sensor). For the high rate loop, the terms $\boldsymbol{\theta}^0$ and $\mathbf{v}^0$ are the encoder positions and sensor values based on the robot configuration used to construct the QP in the low-rate SOFA loop. The term, $\mathbf{x}_{\text{ref}}$ is a constant reference position (placed at the geometric center of the robot).

A. Low Rate Sensing

Within the low rate SOFA loop, a sensing QP is solved to update the full state of the robot. This is used to give a new model for use in the high-rate sensing and rendering loops. With that QP, we solve for the motor and end-effector forces that minimize the error between the sensed encoder and sensor values and those that would be predicted by the model:

$$\min_{d\boldsymbol{\lambda}_a, d\boldsymbol{\lambda}_e} \quad \frac{1}{2}\begin{bmatrix}\boldsymbol{\delta}_a\\ \boldsymbol{\delta}_s\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\delta}_a\\ \boldsymbol{\delta}_s\end{bmatrix} \tag{19}$$

which is equivalent to

$$\begin{aligned}\min_{d\boldsymbol{\lambda}_a, d\boldsymbol{\lambda}_e} \quad & \frac{1}{2}\begin{bmatrix}d\boldsymbol{\lambda}_a\\ d\boldsymbol{\lambda}_e\end{bmatrix}^T\begin{bmatrix}\mathbf{W}_{aa} & \mathbf{W}_{ae}\\ \mathbf{W}_{sa} & \mathbf{W}_{se}\end{bmatrix}^T\begin{bmatrix}\mathbf{W}_{aa} & \mathbf{W}_{ae}\\ \mathbf{W}_{sa} & \mathbf{W}_{se}\end{bmatrix}\begin{bmatrix}d\boldsymbol{\lambda}_a\\ d\boldsymbol{\lambda}_e\end{bmatrix}\\ & + \begin{bmatrix}\boldsymbol{\delta}_a^0\\ \boldsymbol{\delta}_s^0\end{bmatrix}^T\begin{bmatrix}\mathbf{W}_{aa} & \mathbf{W}_{ae}\\ \mathbf{W}_{sa} & \mathbf{W}_{se}\end{bmatrix}\begin{bmatrix}d\boldsymbol{\lambda}_a\\ d\boldsymbol{\lambda}_e\end{bmatrix}.\end{aligned} \tag{20}$$

After solving this QP, $d\boldsymbol{\lambda}_a$ and $d\boldsymbol{\lambda}_e$ are applied to the system and the configuration is updated for the next low rate step. The sensing problem is underactuated because eleven forces, six from the motors and five from the end-effector, are used to control twelve positions, six motor positions and six sensor readings. Thus there is a unique solution to the sensing QP. This implicitly assumes that no torques are applied to the device in the θ_Y direction.

B. High Rate Sensing

The high rate sensing is very similar to the low rate sensing, except that we update the values of $\boldsymbol{\delta}_a^0$ and $\boldsymbol{\delta}_s^0$ using new encoder and sensor values, $\boldsymbol{\theta}_{encoder}$ and $\mathbf{v}_{sensor}$. Note that the compliance matrices are not updated at high rates.

From the solution of the QP, we have the sensed force at the end-effector:

$$\boldsymbol{\lambda}_{e,sense} = d\boldsymbol{\lambda}_e + \boldsymbol{\lambda}_{e,0}. \tag{21}$$

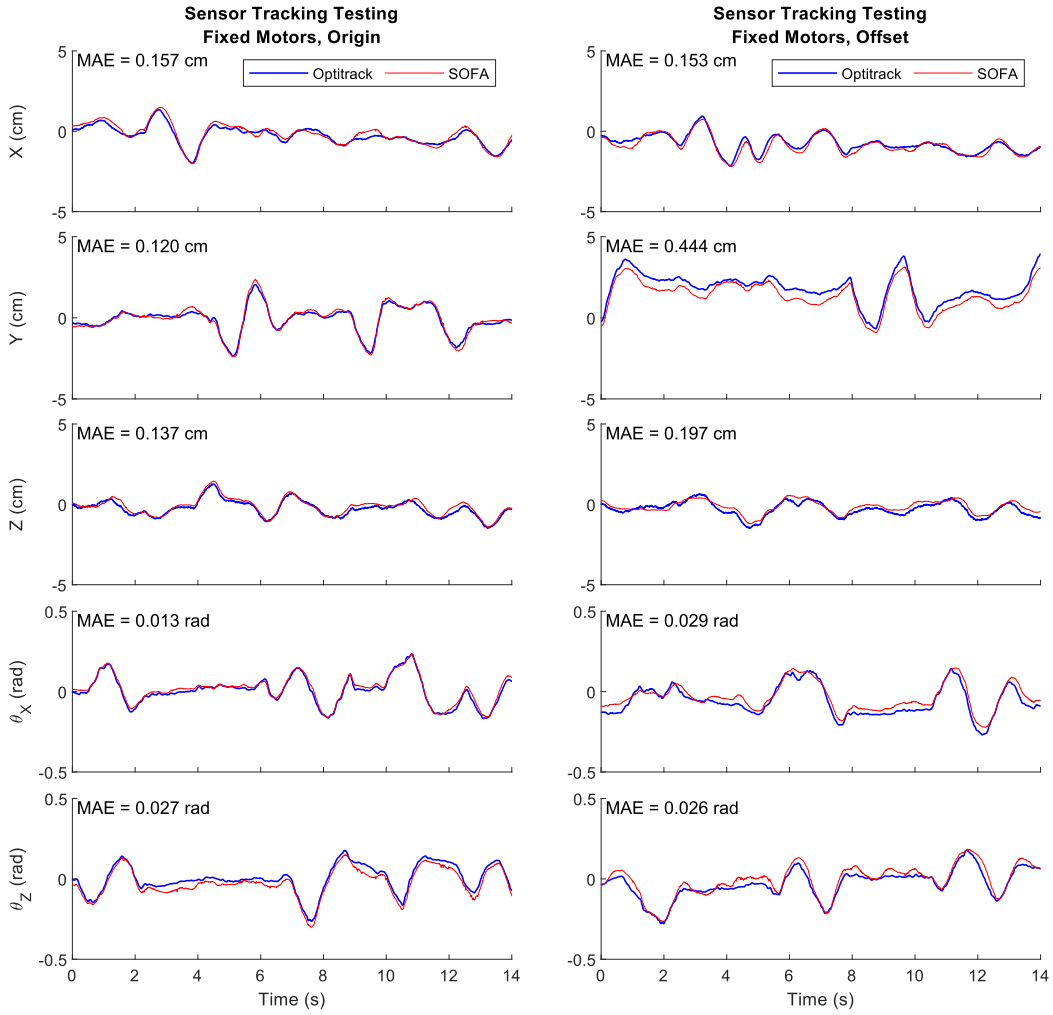


Fig. 9. Sensing using the inverse simulation (SOFA) vs the actual end-effector pose from the OptiTrack system when the motors are holding a fixed position, defined as the motor position to achieve either the origin of the workspace (left) or an offset position (right) with no forces applied to the end effector. Both the sensor voltage and the actuator position are used to estimate the position of the end-effector. The device is fairly stiff when the motors are held in place, so smaller displacements were used. With the motors commanded to a fixed position, almost all of the sensing is performed via the flex sensors. The end-effector sensing is more accurate around the origin than in the offset position. MAE is Mean Absolute Error.

We can then compute the sensed position of the end-effector, which will be important in the virtual environment, as follows:

$$\mathbf{x}_{sense} = \begin{bmatrix} \mathbf{W}_{ea} & \mathbf{W}_{ee} \end{bmatrix} \begin{bmatrix} d\boldsymbol{\lambda}_a \\ d\boldsymbol{\lambda}_e \end{bmatrix} + \mathbf{x}^0. \tag{22}$$

C. Results

To validate the position sensing, an OptiTrack motion sensing system was used to track the end-effector position while it was moved by hand under two different motor conditions: no motor torque and constant motor position. Results are shown in Figures 8 and 9. We achieve fairly good sensing across the workspace both in the case when the motors are off (free to rotate, so some of the motion comes from rotation of the motors as well as from the deformation of the beams) and when the motors are rendering a constant fixed position (so all sensing comes from the flex sensors). The mean absolute error (MAE) ranges from 2 to 5.5% of the total workspace for both translation and rotation. The mean absolute error of the calibrations for the sensors was 4.6% of its range, so much of the error in position sensing is likely due to error in the sensor fit.

After validating the accuracy of the approach on the robot position sensing (with a high frequency refresh) we also validated the ability of this approach to measure the force applied by the user on the robot using deformation sensing, encoder position, and the model. This important feature was validated by connecting a UR3 Robot with an OnRobot force torque sensor to the end-effector as shown in Figure 10. The UR3 Robot moved the end-effector through a defined trajectory that included translation and rotation while the haptic device was rendering a virtual environment. Results are

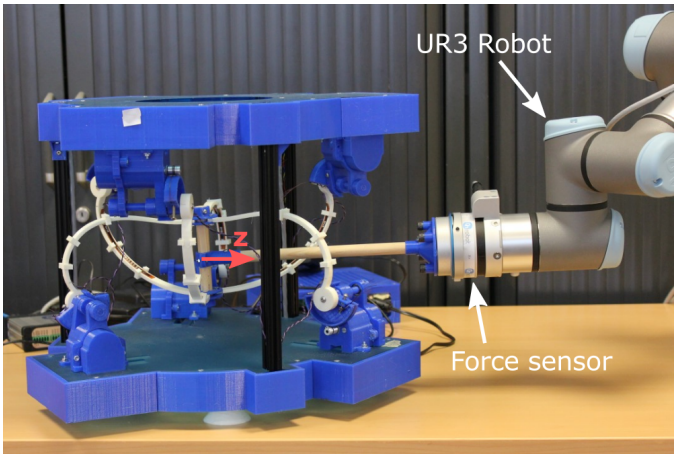


Fig. 10. A UR3 Robot was connected to the haptic device end-effector via a rod. An On Robot 6 axis force sensor was used to validate the force measurements obtained from SOFA. Only the motions and forces/torques in the z direction are considered so that lever effects of the connecting rod can be ignored.

shown in Figure 11.

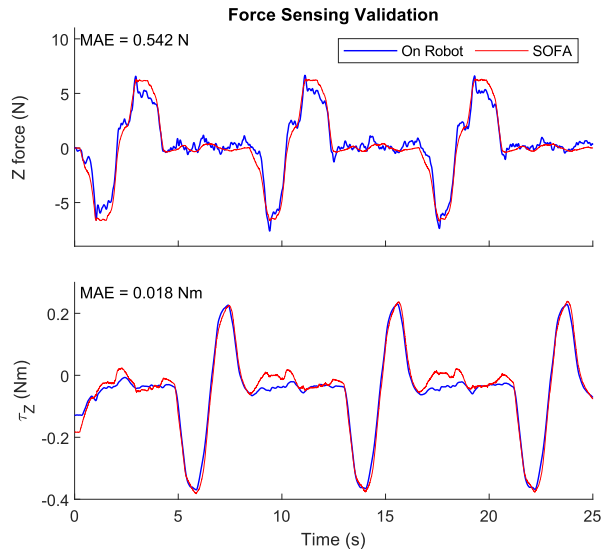


Fig. 11. Comparison of forces and torques measured via an external force sensor and from the high rate Sofa inverse simulation. A UR3 Robot moved the end-effector through trajectories that included z translation and z rotation. Measurements from the force sensor were filtered.

D. Discussion

Based on these results, we see that we are able to sense both position and applied forces at high rates with reasonable accuracy by updating the complete model at a slower rate and using the condensed linear system at high rates. We also see that the model-based calibration procedure for the flex sensors is sufficient to lead to accurate reconstruction of the end-effector force and position.

Accurately measuring the forces applied is an important result of this work because it gives an advantage to deformable parallel robots over rigid ones: thanks to the deformation of their structure, we can estimate the forces exerted on the end-effector without the need to use a load cell. This advantage could contribute to the development of smaller or less expensive haptic devices.

Another advantage is that this robot can now be seen as an *active* force and torque sensor (5D). Indeed, force sensors are usually based on a passive deformation measurement, often uncompensated. In our case, thanks to the actuation of the robot, we can measure the force exerted on the robot structure while maintaining (by control of the actuation) the robot end-effector at its reference position, which provides a pure force measurement (at constant position).

VIII. Haptic Rendering

For good haptic rendering, particularly to render higher stiffnesses, control rates of 500 Hz to 1 kHz are typically needed (see [42] and [43]). As in the case of sensing, the haptic rendering relies on the condensed QP provided from the low rate full SOFA model. In the high rate loop, the first QP optimization retrieves the end-effector position and force, and the second determines the motor position to achieve the desired haptic rendering.

In this section, we discuss the problem of virtual coupling with a deformable interface, then we describe the implementation we propose in this work, using high-rate QP optimization, and finally, we detail the obtained results.

A. Virtual coupling

To achieve a stable and transparent haptic interaction with a virtual environment, a virtual coupling scheme is defined. Reference works such as [44] give the different types of coupling, with a two-port representation and the impedance and admittance couplings widely used in the haptic community. In the case of impedance, the movements (of the interface and the environment) are measured to calculate the forces to be transmitted (by the interface and on the environment). In the case of admittance, it is the opposite, forces are measured and movements are imposed through the device.

In most coupling schemes with rigid robots, it is often pointed out that the impedance of the user's grasp plays an important role but the intrinsic impedance of the haptic interface should not interfere. Designs for these impedance-type haptic devices aim at having *transparent* interfaces with a very low mass and very little friction. For impedance schemes, the user position is often directly obtained by the geometric model of the robot, using the motor positions with the assumption of infinite stiffness. In a way, the rendering cannot be stiffer than the natural stiffness of the device.

With the parallel continuum haptic device, the situation is different: our device is designed to be compliant. The control must therefore be able to provide a stiffer rendering than the natural low stiffness of the device by compensating the natural compliance so that the device does not interfere in the transparency of the rendering.

For this reason, we opted for a hybrid scheme which performs the following steps at high rates:

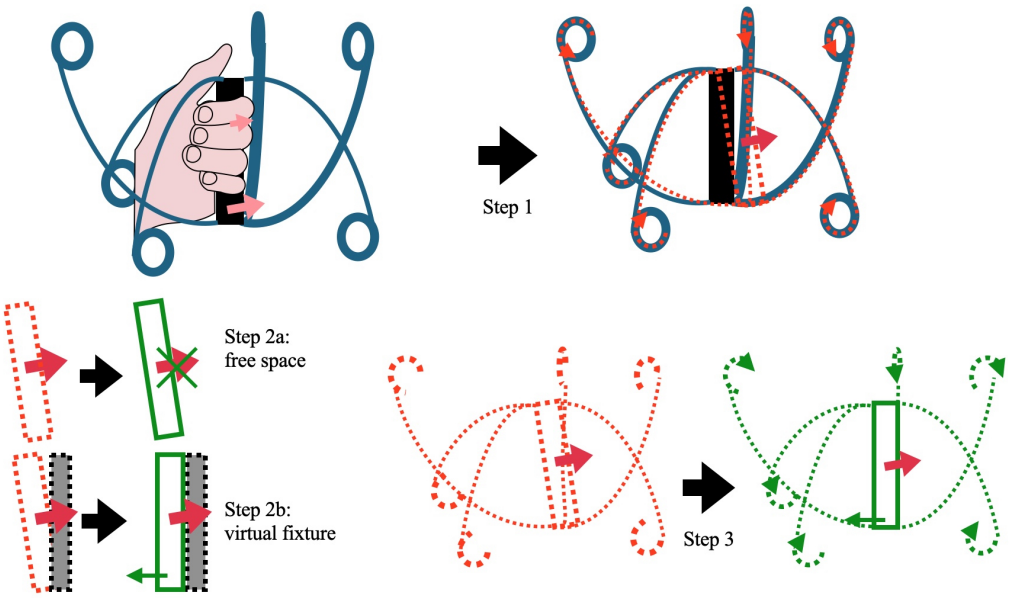


Fig. 12. During step 1, the movement of the interface and the force of the user are estimated. Then, a new position is found to be coherent in every direction with the virtual environment (step 2a: free space or step 2b: virtual fixture) and during step 3, the inverse model is applied to reach this coherent position of the interface by actuating the device

- Step 1: From the estimation of the end-effector position described in section VII we obtain the position and orientation of the end effector that can be coupled to a proxy in the virtual environment (like for impedance coupling). We also obtain the force exerted by the user from this estimation.
- Step 2a: If there is no virtual fixture in the virtual environment (i.e. free space), the desired position is set to match the current position of the user and the desired user force is set to zero.
- Step 2b: If there is a virtual fixture, a desired position for the proxy is computed. We have tested the algorithm with fixed and rigid obstacles, so the proxy stays at the surface (note that if the environment had had a given compliance, the force could have been integrated to obtain a desired position, but it has not been tested in this work as our motivation was to evaluate stiff rendering). The desired user force is set to match that currently measured, so that, if the user maintains the same force, they will move to the desired proxy position. Note that if the user maintains a given position, the force will continue to increase under this scheme.
- Step 3: The desired position of the proxy and the desired user force is the input of a new inverse problem (see the following section VIII-B). The goal is to compute the change of actuation which minimizes the position error of the device (like admittance coupling).

Figure 12 illustrates these three steps that are computed at high rates (500 Hz) in the haptic loop. Figure 13 illustrates the control loops running at different rates: the low level motor control loop (1 kHz), the haptic rendering loop (500 Hz), and the low rate loop that includes the full FEM model (20 Hz).

B. High Rate Rendering

The QP used for haptic rendering is defined below. To distinguish between the values used for sensing and those used for rendering, rendering variables will be depicted with a tilde if different from those used in sensing:

$$\begin{bmatrix}\tilde{\boldsymbol{\delta}}_e\\ \tilde{\boldsymbol{\delta}}_a\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{ee} & \mathbf{W}_{ea}\\ \mathbf{W}_{ae} & \mathbf{W}_{aa}\end{bmatrix}\begin{bmatrix}d\tilde{\boldsymbol{\lambda}}_e\\ d\tilde{\boldsymbol{\lambda}}_a\end{bmatrix} + \begin{bmatrix}\tilde{\boldsymbol{\delta}}_e^0\\ \tilde{\boldsymbol{\delta}}_a^0\end{bmatrix}. \tag{23}$$

We use a constraint-based approach to render different haptic environments. For each virtual environment, we define directions that should be fixed (virtual fixture, step 2b) and directions that should be free (step 2a). Then, we define goal positions and forces for the end-effector directions and solve a QP to determine the actuator positions to achieve those positions given the assumed forces (step 3).

For rendering free space (step 2a), we set

$$\begin{aligned}\mathbf{x}_{goal} &= \mathbf{x}_{sense}\\ \tilde{\boldsymbol{\lambda}}_e &= 0.\end{aligned} \tag{24}$$

Intuitively, we want the device to move to the user's current position with no forces applied at the effector. If the user moves relative to this position, they will feel the inherent stiffness of the device until the next update (though, given a fast update rate, this should be minimal).

For a fixed position control (step 2b), we set

$$\begin{aligned}\mathbf{x}_{goal} &= \mathbf{x}_{constraint}\\ \tilde{\boldsymbol{\lambda}}_e &= \boldsymbol{\lambda}_{e,sense}.\end{aligned} \tag{25}$$

This means that the device will move to the constraint position if the user applies a constant force.

This fixed position control and free space can be combined to form different environments, by fixing some directions while allowing others to be free. For example, for a stiff

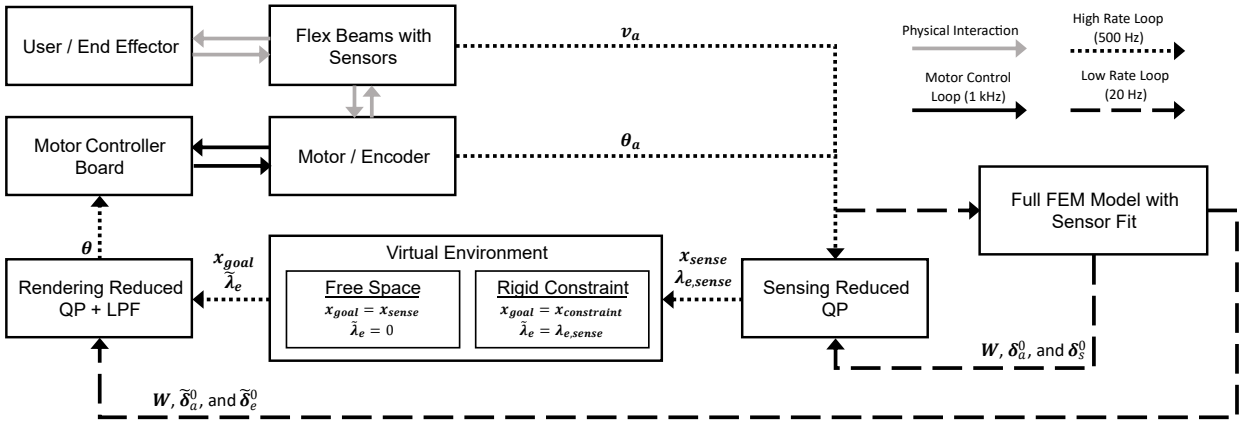


Fig. 13. The block diagram shows the control loops running at different rates, as well as the signals passed between the various blocks. LPF is the low-pass filter (Eq. 30).

wall in the yz-plane at x_{wall}, we render free space in all directions when $x_{sense} > x_{wall}$. When $x_{sense} < x_{wall}$, we set $x_{goal} = x_{wall}$ and $\tilde{\lambda}_x = \lambda_{x,sense}$. The other directions continue to use the free space rendering, with goal position equal to sensed position and desired force equal to 0.

Using the desired end-effector positions and forces defined by the virtual environment, we update the rendering QP from the low-rate loop as follows and solve it to determine the actuator positions to command.

First we update $\tilde{\boldsymbol{\delta}}_e^0$ by taking into account both the new goal position and the desired force:

$$\tilde{\boldsymbol{\delta}}_e^0 = \mathbf{x}^0 - \mathbf{x}_{goal} + \mathbf{W}_{ee} d\tilde{\boldsymbol{\lambda}}_e \tag{26}$$

Then we solve the following QP:

$$\min_{d\tilde{\boldsymbol{\lambda}}_a} \quad \frac{1}{2} \tilde{\boldsymbol{\delta}}_e^T \tilde{\boldsymbol{\delta}}_e. \tag{27}$$

which is equivalent to

$$\min_{d\tilde{\boldsymbol{\lambda}}_a} \quad \frac{1}{2} d\tilde{\boldsymbol{\lambda}}_a^T \mathbf{W}_{ea}^T \mathbf{W}_{ea} d\tilde{\boldsymbol{\lambda}}_a + \tilde{\boldsymbol{\delta}}_e^{0,T} \mathbf{W}_{ea} d\tilde{\boldsymbol{\lambda}}_a. \tag{28}$$

Then we solve for new actuator positions (29), taking into account both the prescribed end-effector force, $\tilde{\boldsymbol{\lambda}}_e$, and the actuator force, $\tilde{\boldsymbol{\lambda}}_a$, that was solved for in the QP. In practice, the matrix $\mathbf{W}_{ea}^T \mathbf{W}_{ea}$ is 1 rank deficient (rank 5). Depending on the QP solver, this could be an issue. If needed, the matrix $\alpha \mathbf{W}_{aa}$ can be added to the QP objective function, with α a small scalar value, to find a unique solution which minimizes the work of the actuator forces as described in [31].

$$\tilde{\boldsymbol{\theta}}_a = \boldsymbol{\theta}^0 + \begin{bmatrix} \mathbf{W}_{ae} & \mathbf{W}_{aa} \end{bmatrix} \begin{bmatrix} d\tilde{\boldsymbol{\lambda}}_e \\ d\tilde{\boldsymbol{\lambda}}_a \end{bmatrix} \tag{29}$$

We filter the commanded actuator position using a low-pass filter with time constant of 6 ms (α = 0.25 with Δ_t = 2 ms), so the position sent to the motor controller board is

$$\boldsymbol{\theta} = \alpha \tilde{\boldsymbol{\theta}}_a + (1 - \alpha) \boldsymbol{\theta}_{last}. \tag{30}$$

This filter was added in order to achieve stability of the controller. Since we have not done a thorough stability analysis of the system, we do not know the precise limits of stability. This filter will increase the apparent damping of the system.

C. Results

First we evaluated the stiffness rendering of the device, comparing three test cases in translation and rotation. Figures 14 and 15 show the results for translation and rotation, respectively. The forces were measured from the force sensor while the UR3 robot moved the end-effector through a trajectory. The end-effector position and orientation were recorded from the high-rate sensing. In the first case, the motors of the robot were controlled to a constant position corresponding to the robot being at the origin. This represents the natural stiffness of the device without any active sensing or control. The next condition tested was free space, in which the haptic device was controlled to minimize the force on the end-effector. Finally, the haptic device was controlled to maintain the robot at the origin, representing the maximum output stiffness of the device.

These stiffness comparisons show that the haptic device is able to render both higher and lower stiffness than the inherent stiffness of the device. This is seen in the differences in slope of the force-position curves. This is clearer in the rotational case than the translation case. Particularly in the translation case, aligning the UR3 to the origin of the haptic device proved challenging, leading to an offset in the initial position. For the free space measurement, the motion from the UR3 does not start at the origin, but over the same distance, does not reach the same maximum force as the fixed motor position. For the position control case, the small initial offset in position in the negative direction leads to a large accumulation of force over time. If the end-effector position is held with a constant offset from a desired position, the restorative force will increase. This leads to the initial force being large and negative. Other contributing factors to the differences seen in translation and rotation could include the relative magnitude of the motions relative to the workspace of the robot and the inherent stiffness of the device in these different directions.

A rigid virtual wall in orientation was also rendered. Moving in free space toward the wall, a relatively constant, damping-like force is seen. When the wall is reached and the UR3 dwells in the wall, the force increases to push the user out of the wall. Some chatter is seen as the end-effector is moved

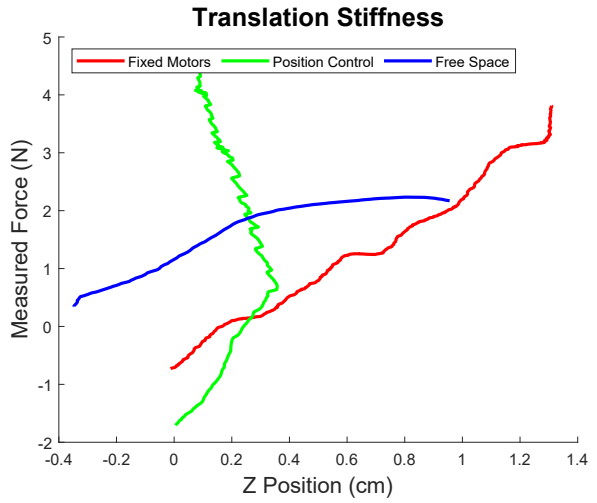


Fig. 14. Force-position curves of three test conditions show the control algorithm rendering maximum-stiffness position control and minimum-stiffness free space, relative to the inherent stiffness of the device when the motors are fixed. Offsets in initial position between the UR3 robot and the haptic device origin lead to the curves being offset from each other.

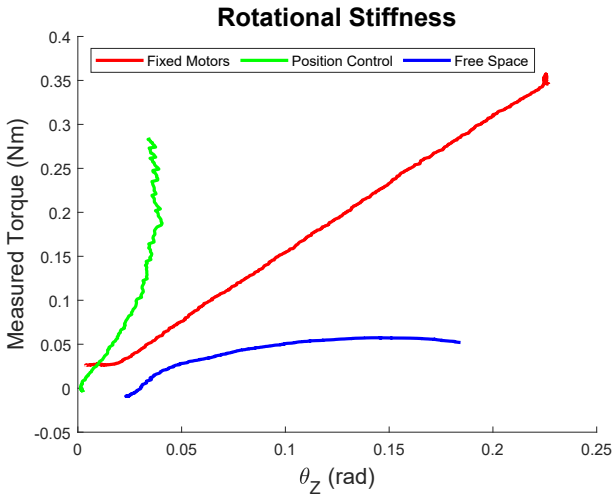


Fig. 15. Torque-orientation curves show the relative rotational stiffness for three test conditions: fixed motors (inherent device stiffness), position control (maximum active stiffness), and free space (minimum active stiffness). The control is capable of rendering both higher and lower stiffness than the inherent device stiffness.

away from the wall, and a damping force is again seen in the opposite direction.

Finally, we rendered virtual fixtures, including a line constraint (Figure 17), a box (Figure 18), and a circle constraint (Figure 19). The box constraint was rendered as virtual walls set at symmetric positions about the origin. The user was free to translate and rotate the end-effector inside the box (with some residual damping forces), but would be constrained from moving out of the walls using the same approach as described for the virtual wall. For the line constraint, the user was constrained to move along the line. Motion in the direction along the line was rendered as free motion, while motion normal to the line was constrained. The rotational degrees of freedom were constrained as well.

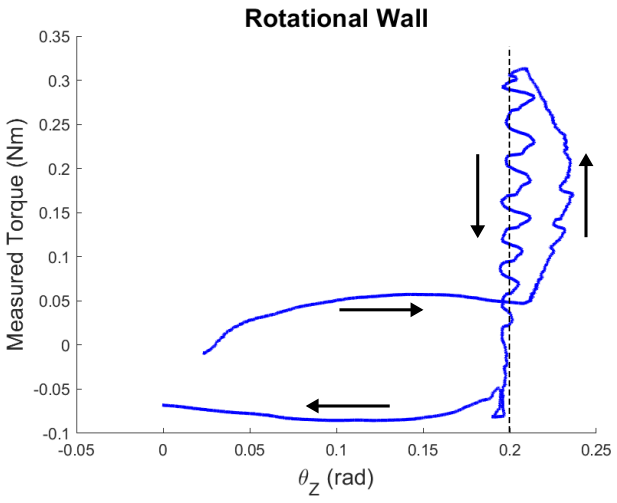


Fig. 16. A virtual wall was rendered in rotation at 0.2 rad. As the end-effector is rotated toward the wall, a small, constant resistive torque is experienced, similar to damping. The force increases when the end-effector penetrates the wall and as the UR3 robot dwells at the maximum position. Some chatter is seen as the robot moves away from the wall, and a damping torque is again seen on the return to the origin through free space.

The circular virtual fixture constrained the user to move along a circle in 3D space. The figure shows position data as sensed in the SOFA simulation, which showed that the user could be constrained along the circle with a high degree of accuracy. This constraint was rendered with two different approaches: a hybrid approach and an admittance approach. In the hybrid approach, the user force in the direction tangent to the constraint was set to zero, the end-effector force in the direction normal to the constraint was set to the sensed force in that direction, and the target position was set to the sensed position projected onto the circular constraint. Then the actuator position was solved for. In the admittance approach, the end-effector force was set to the sensed force and the target position was set to the end-effector position projected onto the constraint with an additional displacement in the tangent direction proportional to the force in the tangent direction (representing the response of a mass). Figure 19 shows the constrained motion using the admittance controller.

The rendering algorithm that we use is based primarily on position. Both the flex sensors and the encoders give a measure of position, and this is mapped to a force and position of the end-effector through the model. Then, the control forces are mapped into control positions via the model. This makes the control less sensitive to absolute errors in the model parameters than if forces were used directly. However, discrepancies between the relative values of the modeled stiffness and mass can lead to errors, since the interaction between these will impact the rendered position and force.

IX. Conclusion and Future Work

In this paper, we presented a novel 5-DOF haptic device featuring a parallel arrangement of continuously deformable beams. The device was controlled via a model of the beam mechanics using a constraint-based approach to render both

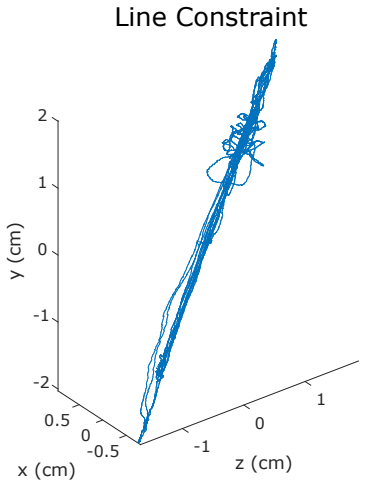


Fig. 17. A line virtual fixture was rendered in 3D space using an admittance approach. The sensed position is the position computed by the high rate linearization from the SOFA simulation.

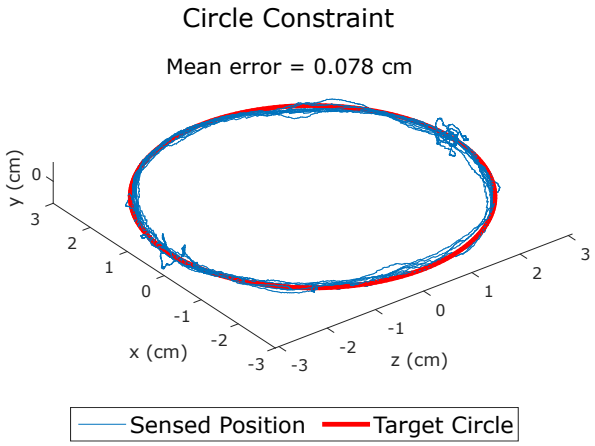


Fig. 19. A circular virtual fixture was rendered in 3D space using an admittance approach. The circle radius is 3 cm and the mean error was less than 1 mm. The sensed position is the position computed by the high rate linearization from the SOFA simulation.

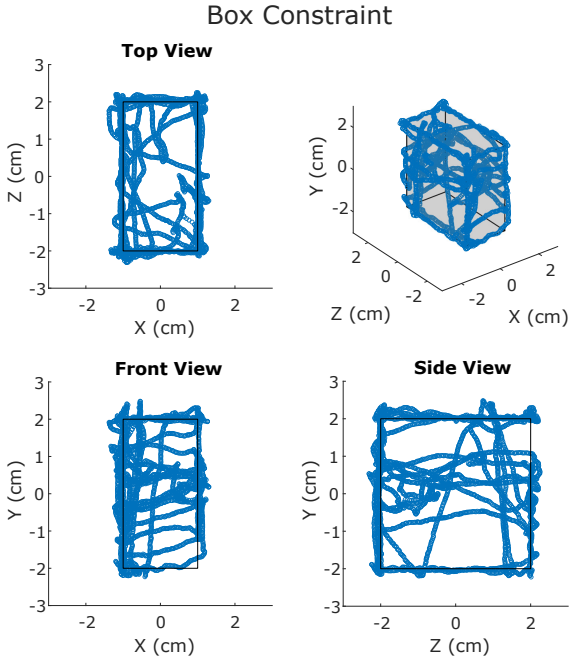


Fig. 18. A box virtual fixture was rendered as wall constraints around the origin. Rotational axes were free throughout the space, and the translational axes were free within the box.

free space and rigid fixtures, representing the two extremes of impedance. The following main contributions were presented: 1) A workspace analysis was performed which showed that, free from actuator torque constraints, a stiffer device allows for a higher force output, but that actuator torque constraints limit the reachable workspace of a stiffer device more than that of a less stiff device. 2) We used flex sensors, calibrated via a direct model of the device, to map sensor signals to the relative angle of nodes along the flexible beam. These sensors were then integrated into a condensed model that could be used to sense the end-effector force and position. 3) Finally, a multi-rate haptic loop used a linearization of the dynamic model updated at low rates to control the haptic device at high rates and to render stiff and free space environments. Combining deformation sensing and the model, we actively compensate for the deformation of the device, allowing us to render higher stiffnesses than a relatively soft structure.

There are certainly some improvements that could be made to the device and control. A more complex sensor calibration which allow for more than a linear function of the sensor nodes could improve the sensor fit, which should improve the position sensing. The robot has a relatively small usable workspace, in practice. In future work, we could use the proposed approach to study the workspace of the robot on other continuous parallel robot architectures, to see which designs are more favorable to haptic interaction. After about two years of irregular use, we have observed fatigue on the material properties on some of the 3D printed legs over time. In future work, we could use industrial materials, such as spring steel for example, to improve the durability and the repetitive behavior of the device.

For rigid virtual fixtures, the interface's response to a user force is sometimes a bit delayed resulting in some motion, but as the force is maintained, the interface corrects the position. In order to detect this user force, the interface must be deformed so that the sensors measure bending of the legs. Then the model accounts for this force, and the interface can apply its correction. In this series of steps, delays accumulate: the viscosity of the material, the responsiveness of the sensors, the processing time of the model, the controllers, electronic devices, and motors. All these elements could be optimized in terms of speed to improve overall quality.

This work considered two extreme cases for control, using an admittance-type controller for rigid constraints and a force-based impedance type controller for free space. A true impedance-type control could be implemented by controlling the motors in force instead of position. Additionally, either of

these controllers could be applied to a different haptic environment. In this case we also had constant virtual environments (the walls or constraints were static). More complex virtual environments would be more challenging to render with the multi-rate control loop, since the dynamics of the scene would need to be included in the high rate loop. This would lead to greater time delay in controlling the system, which would need to be accounted for. A thorough analysis and experimental validation of the dynamic response and stability of the closed-loop system, taking into account the effects of sensor response time, time delay, and discretization, would also provide more insight into the potential benefits and limitations of the device and the presented methodology. Finally, we observe sometimes some uncompensated vibrations of the device. A full dynamic control loop, including a dynamic model analysis of the device could be investigated to remove these vibrations.

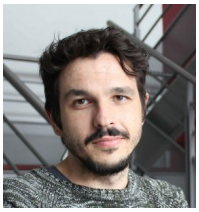
APPENDIX

Extension	Media type	Description
1	Video	Device kinematics and haptic rendering

FUNDING

This work was supported in part by a National Science Foundation Graduate Research Fellowship to MK, the National Science Foundation under Grant 1830163 to AO, a Chateaubriand grant to MK, a Fulbright grant to CD and Inria associated team ACDC grant to CD and MK.

Margaret Koehler received the B.S., M.S., and Ph.D. degrees in 2014, 2016, and 2020, respectively, from Stanford Univeristy, Stanford, CA, USA, all in mechanical engineering. Her research interests include soft robotics, controls, and simulation, particularly for medical applications.



Thor Bieze received a PhD degree in robotics from University of Lille in France in 2017. He is a research and development engineer at Inria Lille - Nord Europe. His research topics include continuum robot design and control, morphological computation and intelligent system design.

Alexandre Kruszewski received his PhD Degree in automatic control form the university of Valenciennes in France in 2006. He arrived at Ecole Centrale de Lille in 2007 as an assistant professor and worked on digital control for networked systems. In 2015 in joined the DEFROST Team in which he focused his research on control design for soft robots. In 2020 was promoted full professor.

Allison M. Okamura (Fellow, IEEE) received the B.S. degree from the University of California, Berkeley, Berkeley, CA, USA, in 1994, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, USA, in 1996 and 2000, respectively, all in mechanical engineering. She is currently the Richard W. Weiland Professor in the School of Engineering and Professor of Mechanical Engineering with Stanford University. Her research interests include haptics, teleoperation, medical robotics, virtual environments and simulation, neuromechanics and rehabilitation, prosthetics, and engineering education. Prof. Okamura's awards include the 2020 IEEE Engineering in Medicine and Biology Society Technical Achievement Award and the 2019 IEEE Robotics and Automation Society Distinguished Service Award. She was the Editor-in-Chief of the *IEEE Robotics and Automation Letters* from 2018-2021.

Christian Duriez is Research Director at Inria Lille - Nord Europe. He received a PhD degree in robotics from University of Evry and CEA in France in 2004 followed by a postdoctoral position at the CIMIT SimGroup in Boston. He arrived at INRIA in 2006 and worked on interactive simulation of deformable objects and haptic rendering with focus on surgical simulation. He was promoted Directeur de Recherche in 2014 and is now the head of DEFROST team, created in January 2015. In 2018 he was invited researcher at Stanford University which was the starting point for this study. His research topics are Soft Robot models and control, Fast Finite Element Methods, simulation of contact response, new algorithms for haptics... He participated to the creation of the open-source SOFA framework. He was also one of the founders of the company InSimo.

REFERENCES

[1] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, ser. Studies in Mechanobiology, Tissue Engineering and Biomaterials, Y. Payan, Ed. Springer, 2012, vol. 11, pp. 283–321.

[2] B. Hannaford and A. M. Okamura, "Haptics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer International Publishing, 2016, pp. 1063–1084.

[3] J. Arata, H. Kondo, N. Ikedo, and H. Fujimoto, "Haptic Device Using a Newly Developed Redundant Parallel Mechanism," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 201–214, 2011.

[4] G. Pratt and M. Williamson, "Series elastic actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 1995, pp. 399–406.

[5] R. B. Gillespie, Dongwon Kim, J. M. Suchoski, Bo Yu, and J. D. Brown, "Series elasticity for free free-space motion for free," in *IEEE Haptics Symposium (HAPTICS)*, 2014, pp. 609–615.

[6] R. B. Gillespie, T. Shin, F. Huang, and B. Trease, "Automated Characterization and Compensation for a Compliant Mechanism Haptic Device," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 1, pp. 136–146, 2008.

[7] M. Koehler, A. M. Okamura, and C. Duriez, "Stiffness Control of Deformable Robots Using Finite Element Modeling," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 469–476, 2019.

[8] V. Hayward and K. Maclean, "Do it yourself haptics: part I," *IEEE Robotics & Automation Magazine*, vol. 14, no. 4, pp. 88–104, 2007.

[9] N. Hogan, "Impedance Control: An Approach to Manipulation: Part II—Implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.

[10] J. Colgate and J. Brown, "Factors affecting the Z-Width of a haptic display," in *IEEE International Conference on Robotics and Automation*, 1994, pp. 3205–3210.

[11] C. R. Carignan and K. R. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *Haptics-e*, vol. 1, no. 2, 2000.

[12] N. Colonnese and A. Okamura, "Stability and quantization-error analysis of haptic rendering of virtual stiffness and damping," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1103–1120, 2016.

[13] R. J. Adams, M. R. Moreyra, and B. Hannaford, "Stability and Performance of Haptic Displays: Theory and experiments," in *ASME International Mechanical Engineering Congress and Exhibition*, 1998, pp. 227–234.

[14] B. Hannaford and Jee-Hwan Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[15] H. Vallery, J. Veneman, E. van Asseldonk, R. Ekkelenkamp, M. Buss, and H. van Der Kooij, "Compliant actuation of rehabilitation robots," *IEEE Robotics & Automation Magazine*, vol. 15, no. 3, pp. 60–69, 2008.

[16] F. Sergi and M. K. O'Malley, "On the stability and accuracy of high stiffness rendering in non-backdrivable actuators through series elasticity," *Mechatronics*, vol. 26, pp. 64–75, 2015.

[17] D. P. Losey and M. K. O'Malley, "Effects of discretization on the K-width of series elastic actuators," in *IEEE International Conference on Robotics and Automation*, May 2017, pp. 421–426.

[18] V. Aloi, C. Black, and C. Rucker, "Stiffness Control of Parallel Continuum Robots," in *ASME Dynamic Systems and Control Conference*, 2018, pp. 1–7.

[19] C. E. Bryson and D. C. Rucker, "Toward parallel Continuum manipulators," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 778–785.

[20] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2017.

[21] B. Mauzé, R. Dahmouche, G. J. Laurent, A. N. André, P. Rougeot, P. Sandoz, and C. Clévy, "Nanometer precision with a planar parallel continuum robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3806–3813, 2020.

[22] A. W. Mahoney, P. L. Anderson, P. J. Swaney, F. Maldonado, and R. J. Webster, "Reconfigurable parallel continuum robots for incisionless surgery," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4330–4336.

[23] J. Till and D. C. Rucker, "Elastic stability of cosserat rods and parallel continuum robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 718–733, 2017.

[24] L. L. Howell, *Compliant mechanisms*. John Wiley & Sons, 2001.

[25] M. Salerno, A. Firouzeh, and J. Paik, "A Low Profile Electromagnetic Actuator Design and Model for an Origami Parallel Platform," *Journal of Mechanisms and Robotics*, vol. 9, no. 4, p. 041005, 2017.

[26] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, and S. Kim, "Soft robot review," *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 3–15, 2017.

[27] S. S. Robinson, K. W. O'Brien, H. Zhao, B. N. Peele, C. M. Larson, B. C. Mac Murray, I. M. Van Meerbeek, S. N. Dunham, and R. F. Shepherd, "Integrated soft sensors and elastomeric actuators for tactile machines with kinesthetic sense," *Extreme Mechanics Letters*, vol. 5, pp. 47–53, 2015.

[28] R. K. Kramer, C. Majidi, R. Sahai, and R. J. Wood, "Soft curvature sensors for joint angle proprioception," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1919–1926.

[29] Y. L. Park, B. R. Chen, and R. J. Wood, "Design and Fabrication of Soft Artificial Skin Using Embedded Microchannels and Liquid Conductors," *IEEE Sensors Journal*, vol. 12, no. 8, pp. 2711–2718, 2012.

[30] G. Gerboni, A. Diodato, G. Ciuti, M. Cianchetti, and A. Menciassi, "Feedback Control of Soft Robot Actuators via Commercial Flex Bend Sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1881–1888, 2017.

[31] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[32] C. Duriez, S. Cotin, J. Lenoir, and P. Neumann, "New approaches to catheter navigation for interventional radiology simulation," *Computer Aided Surgery*, vol. 11, no. 6, pp. 300–308, 2006.

[33] J. S. Przemieniecki, *Theory of matrix structural analysis*. Dover, 1985.

[34] T. Morales Bieze, A. Kruszewski, B. Carrez, and C. Duriez, "Design, implementation, and control of a deformable manipulator robot based on a compliant spine," *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1604–1619, 2020.

[35] Z. Zhang, J. Dequidt, A. Kruszewski, F. Largilliere, and C. Duriez, "Kinematic modeling and observer based control of soft robot using real-time Finite Element Method," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5509–5514.

[36] R. Finotello, T. Grasso, G. Rossi, and A. Terribile, "Computation of kinetostatic performances of robot manipulators with polytopes," in *IEEE International Conference on Robotics and Automation*, 1998, pp. 3241–3246.

[37] T. Yoshikawa, "Translational and rotational manipulability of robotic manipulators," in *American Control Conference*, 1990, pp. 228–233.

[38] K. Fukuda, "cddlib - C Double Description Library," 2018, Accessed: 15 Sept 2019. [Online]. Available: https://inf.ethz.ch/personal/fukudak/cdd_home

[39] M. C. Troffaes, "pycddlib - python wrapper for cddlib," 2015, Accessed: 15 Sept 2019. [Online]. Available: https://github.com/mcmtroffaes/pycddlib

[40] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[41] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[42] D. Kubus, I. Weidauer, and F. M. Wahl, "1khz is not enough—how to achieve higher update rates with a bilateral teleoperation system based on commercial hardware," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5107–5114.

[43] S. Booth, F. Angelis, and T. Schmidt-Tjarksen, "The influence of changing haptic refresh-rate on subjective user experiences-lessons for effective touch-based applications," in *Proceedings of eurohaptics*. Citeseer, 2003, pp. 374–383.

[44] R. Adams and B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 465–474, 1999.