2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

An Admittance Control Scheme for Haptic Interfaces Based on Cable-Driven Parallel Mechanisms

Alexis Fortin-Côté, Philippe Cardou and Clément Gosselin

***Abstract*— This paper presents a cable-driven parallel mechanism as a haptic interface and its underlying control method. This human-sized, three-degree-of-freedom mechanism has a tetrahedral architecture, four cables and evolves in three-dimensional space. A brief review of the kinematics of the mechanism is presented. Also, an admittance control law coupled with a closed-loop velocity controller is proposed. The control method is then refined by introducing adaptations for smooth surfaces and sharp edges. This control method is then validated by experimental results. Furthermore, the geometry of the mechanism is identified by a method that does not require any other sensor than the motor encoders.**

I. INTRODUCTION

With virtual reality becoming more and more prominent with CAVE-style environments and wearable commercial devices like the occulus rift [1], arises a need to physically interact with these virtual worlds. Some devices providing force feedback and haptic rendering are already available to consumers, e.g., the PHANToM [2] and Falcon (by Novint inc). Both of these devices use rigid-link robots to transmit a force to the user. Another kind of device is the Maglev 200 (by Butterfly Haptics, LLC), which, as its name implies, uses magnetic fields to transmit haptic rendering to the user. All these solutions work for their respective domains but suffer from a common drawback: their limited workspace.

This is where a cable-driven parallel mechanism (CDPM) as that illustrated in Fig. 1 can provide an advantage relative to conventional mechanisms. CDPMs are known to possess a large workspace and a low inertia, two traits of a good haptic feedback mechanism. CDPMs are currently used in commercial applications like the Skycam [3] and as a control module for a radio telescope [4], for example.

The advantages mentioned above have also led researchers to use them as haptic interfaces. Several research initiatives addressed the development of haptic CDPMs, beginning as early as 1989 [5]. Most reports concentrate on the architecture and components of the CDPMs. Among the pioneering designs, we find the SPIDAR [6], a spatial haptic device that generates forces on the user's finger, but little is shown of the control laws that must have been used. Also, Gallina [9] presented the Feriba-3: a planar three-degree-of-freedom haptic interface sliding on compressed air to limit friction. More work, [7], [8], followed up on those initials research, but did not concentrate either on the control law. Nevertheless, we could infer that they relied on an impedance control scheme, meaning that the authors had to mitigate the effects of friction to obtain a good rendering. Williams [10] worked on a planar device and also used an impedance control law without a force sensor. Morizono et al. [11] used an impedance scheme with force feedback on each string to control the wrench applied by a sport simulating device. In this paper, we rely on an admittance control scheme, as it seems better suited to render contacts with hard surfaces. Such an approach to haptic rendering is explained by Carignan and Cleary [12], but not for a CDPM. Gosselin et al. [13] worked on an admittance control scheme for a planar CDPM, but used closed-loop position feedback as opposed to velocity feedback, which is used here.

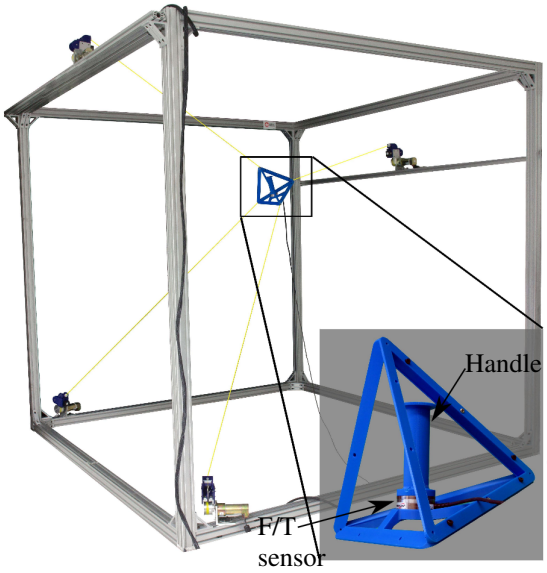


Fig. 1: Photograph of the CDPM used in this work.

Furthermore, it is desired to take advantage of the reconfiguration capabilities of CDPMs in order to adapt their workspace to different virtual environments. This requires that the mechanism be able to determine its geometry efficiently, i.e., that it calibrates itself. Miermeister and Pott [14] expose a method of for the calibration of these devices with force sensors. Borgstrom et al. [15] use a self-calibration method without force sensor to determine the initial lengths of the cables but not the position of the anchor points. This paper presents a method of calibration without force sensor

*This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Canada Research Chair program.

The authors are with the Laboratoire de robotique, Département de génie mécanique, Université Laval, Québec, Canada, G1V 0A6
alexis.fortin-cote.1@ulaval.ca and
philippe.cardou@gmc.ulaval.ca and
gosselin@gmc.ulaval.ca

and able to determine the initial lengths of the cables and position of the anchor points. This self-calibration method, which was briefly presented in [16], is further detailed in Section III. Experimental results are presented to demonstrate the effectiveness of this method.

The mechanism studied, shown in Fig. 1, is a three-degree-of-freedom mechanism used for positioning a point in 3D space. It consists of a rigid platform, for the user to manipulate, held by a set of four cables whose lengths are controlled by reels that are affixed to a rigid frame. The platform shape is that of a regular tetrahedron, which allows each cable to be attached at a different vertex. A face of the tetrahedron is directed towards the user so as to increase access to the handle and to limit interferences between user and cables. The reels are also positioned to form a regular tetrahedron having a similar shape and orientation as the platform. This four-cable configuration allows point positioning in three-dimensional space. The orientation of the moving platform is not restrained but remains fairly constant; the end-effector being much smaller than the fixed frame. The platform itself consists of a handle mounted on a force sensor, which is attached to the platform frame. The handle is positioned at the centroid of the tetrahedron, which prevents the forces applied by the user on the platform from generating large moments. This centroid is the point where forces applied by the cables intersect when the platform is centred in its workspace. The forces applied by the user are measured using a force sensor and are used for control. A measure of these forces is crucial, as it is the input to the admittance control scheme. This method of control has been chosen over an impedance control because it is less sensitive to friction and other perturbations.

This paper is structured as follows: the kinematics of a generic CDPM are presented in Section II; Its calibration is explained in Section III; the haptics control scheme is detailed in Section IV; and finally, experimental results are presented in sec. V.

II. Kinematics

Consider a rigid platform, which is to be manipulated by the user, moving in three-dimensional space and constrained by m cables attached by reels to the fixed frame, as shown schematically in Fig. 2. The points of attachment to the reels are defined by vectors $\mathbf{a}_i$ with respect to reference point O, while the corresponding attachment points on the platform are defined by vectors $\mathbf{r}_i$ expressed relative to the position of the platform reference point P. The position of the latter point is given by vector $\mathbf{p}$. The orientation of the rigid platform is considered constant. Further experiments proved that this simplification is acceptable for the proposed application, as a change of 15° in orientation induces an error in the position $\mathbf{p}$ of less than 10 mm which is acceptable in a context of haptic rendering where user cannot easily perceive this error. It is important to note that cable elasticity and mass are considered negligible and are not accounted for in this model. Under the maximum tension, the cable axial deformation is less than 0.5%, and the cable mass is very low at 1.5 g/m. Thus, the assumption of rigid and massless cables is considered reasonable.

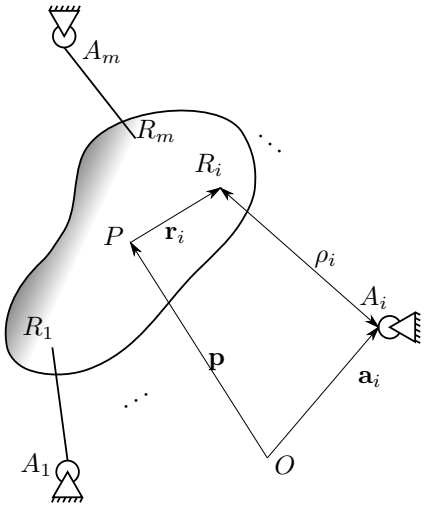


Fig. 2: Geometric model of a generic cable driven parallel mechanism.

Geometric constraints prescribe the length ρ_i of the i^{th} cables as

$$\rho_i^2 = (\mathbf{p} - \mathbf{a}_i + \mathbf{r}_i)^T(\mathbf{p} - \mathbf{a}_i + \mathbf{r}_i). \tag{1}$$

By differentiation and rearrangement of (1), the corresponding velocity equation is written as

$$\rho_i \dot{\rho}_i = (\mathbf{p} - \mathbf{a}_i + \mathbf{r}_i)^T \dot{\mathbf{p}}. \tag{2}$$

This leads, for m cables, to the following first-order kinematic relationship

$$\dot{\boldsymbol{\rho}} = \mathbf{J}\dot{\mathbf{p}} \tag{3}$$

where

$$\dot{\boldsymbol{\rho}} = \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_m \end{bmatrix} \text{ and } \mathbf{J} = \begin{bmatrix} \dfrac{(\mathbf{p} - \mathbf{a}_1 + \mathbf{r}_1)^T}{\rho_1} \\ \vdots \\ \dfrac{(\mathbf{p} - \mathbf{a}_m + \mathbf{r}_m)^T}{\rho_m} \end{bmatrix}_{[m\times 3]} \tag{4}$$

is the Jacobian matrix, which will be used in the control scheme of Section IV.

III. Calibration

The identification of the model's parameters is crucial to its accuracy and, in turn, to the quality of the haptic rendering. The parameters that need identification are the position of each fixed cable anchor point $\mathbf{a}_i$ and the initial length of each cable $\rho_{i,0}$ due to the use of incremental encoders. The position $\mathbf{r}_i$ of the moving attachment points are known accurately, since the end effector was built using fast prototyping. Hence, these parameters are treated as constant in the calibration procedure.

Since a cable-driven parallel mechanism possesses a large workspace, standard measurement methods are cumbersome

or give inaccurate results. The ideal calibration method does not require any other sensor than those used for the haptic control of the CDPM, namely, the motor encoders and the force sensor. The calibration technique used for this mechanism uses only the encoder measurements and is performed in two steps. The first step is an approximation of the positions of $A_i, i = 1, \ldots, m$, by reading the value of the encoders when the platform is located at each of the anchoring points A_i. This method gives two estimates of the distances between pairs of anchoring points, which can then be averaged to yield an initial estimate of their positions. These initial estimates are refined in step two.

The second step of the calibration method consist in exploiting the measurement redundancy of the mechanism. An over-determined system of equations can be generated if the assumption of rigid and massless cables is valid. To this end, the platform is moved across the workspace by the user with cables under constant tension, thus generating n cable-length measurements for each of the m cables. For the j^{th} calibration position, it is possible to rearrange (1) and replace $\rho_{i,j}$ by

$$\rho_{i,j} = \Delta\rho_{i,j} + \rho_{i,0}, \tag{5}$$

where $\Delta\rho_{i,j}$ is the measured length and $\rho_{i,0}$ is the initial length of the cable at initialization of the encoder. It is then possible to generate a system of nm equations, each of which can be written as

$$0 = (\mathbf{p}_j - \mathbf{a}_i + \mathbf{r}_i)^T(\mathbf{p}_j - \mathbf{a}_i + \mathbf{r}_i) - (\Delta\rho_{i,j} + \rho_{i,0})^2, \tag{6}$$

with $i = 1, \ldots, m$, and $j = 1, \ldots, n$, where m is the number of cables and n is the number of positions used for the calibration. Note that vectors $\mathbf{r}_i$ are considered constant, since they are small compared to the length of the cables and since the orientation of the platform remains approximately constant. In order to avoid having to deal with an under determined system of equations, the number of equations should be greater than or equal to the number of unknowns. Here, m equations are obtained for each position of the moving platform, for a total of mn. The unknowns appearing in the equations are $\mathbf{a}_i$, $\rho_{i,0}$, $i = 1, \ldots, m$, and $\mathbf{p}_j$, $j = 1, \ldots, n$, for a total of $4m + 3n$ unknowns. Moreover, it is important to note, that six components of the $\mathbf{a}_i$ vectors need to be prescribed, so as to constrain the mechanism in space. This yields a revised total of $4m+3n-6$ unknowns and the condition to avoid an under determined system becomes

$$n \geqslant \frac{4m-6}{m-3}. \tag{7}$$

Since, in our case, $m = 4$, we must measure a minimum of 10 distinct moving platform positions.

A solution can be obtained by formulating the nonlinear least-squares problem and using as an initial guess the outcome of the first calibration step. Mathematically, this problem takes the following form

minimize

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(u_{i,j})^2 \tag{8}$$

over $\mathbf{a}_i, \rho_{i,0}, i = 1, \ldots, m$,

where

$$u_{i,j} = \| \mathbf{p}_j - \mathbf{a}_i + \mathbf{r}_i \|_2^2 - (\Delta\rho_{i,j} + \rho_{i,0})^2. \tag{9}$$

The solution to this nonlinear least-square problem is obtained via Matlab's implementation of the interior-reflective Newton method. This method is invoked by the *lsqnonlin* function of the Matlab optimization toolbox.

An algebraic Jacobian $\mathbf{H}$, not to be confused with the general Jacobian matrix of (3), can be calculated by differentiation of

$$\mathbf{u} = [\mathbf{u}_1^T \ \cdots \ \mathbf{u}_n^T]^T, \tag{10}$$

with

$$\mathbf{u}_j = [u_{1,j} \ \cdots \ u_{m,j}]^T, \tag{11}$$

and

$$\mathbf{v} = \begin{bmatrix}\mathbf{a}_1^T \ \cdots \ \mathbf{a}_m^T & \rho_{0,1} \ \cdots \ \rho_{0,m} & \mathbf{p}_1^T \ \cdots \ \mathbf{p}_n^T\end{bmatrix}^T, \tag{12}$$

where $\mathbf{u} \in \mathbb{R}^{m \cdot n}$, $\mathbf{u}_j \in \mathbb{R}^m$, and $\mathbf{v} \in \mathbb{R}^{(4m+3n)}$. The expression of this Jacobian matrix is then

$$\frac{\partial \mathbf{u}}{\partial \mathbf{v}} = \mathbf{H} = \begin{bmatrix} \frac{\partial u_{1,1}}{\partial \mathbf{a}_1}^T & \cdots & \frac{\partial u_{1,1}}{\partial \rho_{0,1}} & \cdots & \frac{\partial u_{1,1}}{\partial \mathbf{p}_n}^T \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial u_{n,m}}{\partial \mathbf{a}_1}^T & \cdots & \frac{\partial u_{n,m}}{\partial \rho_{0,1}} & \cdots & \frac{\partial u_{n,m}}{\partial \mathbf{p}_n}^T \end{bmatrix}, \tag{13}$$

where the components can be written as

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial \mathbf{a}_k} &= \begin{cases} -2\mathbf{p}_j + 2\mathbf{a}_i - 2\mathbf{r}_i & \text{if } i = k, \\ \mathbf{0}_3 & \text{otherwise.} \end{cases} \\ \frac{\partial u_{i,j}}{\partial \rho_{0,k}} &= \begin{cases} -2(\Delta\rho_{i,j} + \rho_{i,0}) & \text{if } i = k, \\ \mathbf{0}_3 & \text{otherwise.} \end{cases} \\ \frac{\partial u_{i,j}}{\partial \mathbf{p}_k} &= \begin{cases} 2\mathbf{p}_j - 2\mathbf{a}_i + 2\mathbf{r}_i & \text{if } j = k, \\ \mathbf{0}_3 & \text{otherwise.} \end{cases} \end{aligned} \tag{14}$$

This Jacobian $\mathbf{H}$ is given as an argument to the Matlab function to increase the calculation speed of the algorithm. A conditioning of approximately 10^3 is obtained for $n > 15$, which is deemed acceptable. Notice that the end-effector positions used for the calibration were distributed approximately evenly across the workspace. Using positions that are not sufficiently far apart would lead to a higher condition number of the Jacobian.

By implementing this method, the mechanism is self-calibrated and resulting identified parameters are the position of each fixed cable anchor point $\mathbf{a}_i$ shown on Table I and initial length of each cable $\rho_{i,0}$ shown on Table II.

TABLE I: Position of each cable anchor point A_i

	x (m)	y (m)	z (m)
A_1	0	1.827	0
A_2	0	0	0
A_3	1.886	0.936	1.713
A_4	0	0.866	1.994

TABLE II: Initial length of each cable $\rho_{i,0}$

	x (m)
$\rho_{1,0}$	1.241
$\rho_{2,0}$	0.977
$\rho_{3,0}$	1.528
$\rho_{4,0}$	1.242

As an indication of the accuracy of the calibration, a plot of the value of a model parameter as a function of the number of points used in the calibration is shown in Fig. 3.

A minimum of $n = 10$ points is necessary to perform the calibration as per (7), for $m = 4$. It can be seen that the function plateaus for a number of points n of 30 . For $n \geq 30$, the observed standard deviation is approximately 0.01 m for this parameter, and can reach up to 0.03 m for others. Fig. 4 shows the quadratic mean of the residuals of (8), which also reflects the impact of the number of points n on the solution accuracy and also plateaus around $n = 30$.

It is also pointed out that the residual error for $n = 10$ is not zero, which may seem erroneous at first glance. This is correct, however, and indicates that there is no exact solution even when the system contains as many unknowns as equations. Given a mechanism roughly the size of a 2 m cube, we obtain approximately one to two percent error on the positioning of the anchor points.

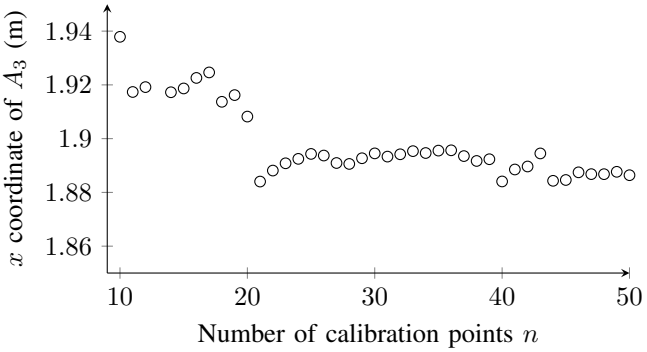


Fig. 3: Coordinate y of an anchor point as a function of the number of points used in the calibration.

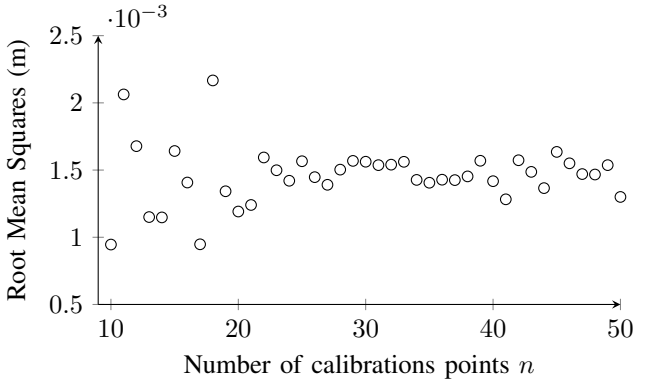


Fig. 4: Quadratic mean of the residual error to the least squares problem for the mechanism as a function of the number of points used in the calibration.

IV. CONTROL

Two classes of control schemes can be used in the control of haptic devices as described in [17]. Impedance control is the most widely used of these two classes, because it does not require a force sensor. An impedance control law takes a displacement as input and outputs a force. The second class of control schemes are called admittance controllers. They use the opposite principle, taking forces as inputs, and supplying displacements as outputs. Admittance control was chosen here since the additional force feedback alleviates the inconveniences caused by friction, which has a great impact on the quality of an impedance control.

A. Control Scheme

Furthermore, a velocity controller is used instead of a position controller because the resulting haptic rendering has proven superior in previous experiments. While a closed-loop control over the velocity of the actuator does not replicate exactly the intended model, it is less sensitive to external parasitic forces than a position controller. The opposite can be said of a closed-loop position control. It replicates the exact intended behaviour of the model but any physical perturbation generates erratic movements due to the controller trying to catch up with the model. An unexpected collision into a physical object, exceeding the actuator capabilities or leaving the workspace are examples of physical perturbations that greatly degrade haptic rendering if closed-loop position control is used and such situations may be very difficult to avoid. All these perturbations still have an effect when velocity control is used, but are less problematic because they do not result in unstable movements of the user platform. Humans seem far more sensitive to instability errors than to velocity errors.

A force control feedback is another approach to haptic control that could have been used. We discarded it because its mass and damper virtual model need acceleration estimates, which are too noisy when derived from encoder measurements.

The general control scheme used for this mechanism is shown in Fig. 5. Forces measured by the sensor and applied by the virtual environment are driving the physical model. This mechanical model of the virtual environment prescribes a Cartesian velocity for the platform to follow the virtual object. The Jacobian matrix $\mathbf{J}$ transforms this Cartesian velocity into joint velocity which is given as a command to the low-level controller G_c, which is a closed-loop controller on articular velocity as displayed by the feedback of $\dot{\boldsymbol{\rho}}$ in Fig. 5. The forward kinematics are used to feedback the Cartesian position to the Jacobian and the virtual environment. It is noted that, unlike in many other control schemes devised for cable driven mechanisms, the controller proposed here does not require the management of internal cable tensions.

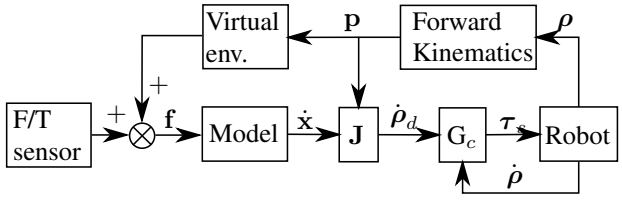


Fig. 5: Block diagram of the control scheme.

The chosen admittance model is that of a mass and damper, i.e.,

$$\mathbf{f} = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} \tag{15}$$

where $\mathbf{f}$ is the resultant force applied on the object by the environment and is further described in Section IV-B. The terms of mass and damping take the form

$$\mathbf{M} = m_p \mathbf{1}_{[3x3]} \text{ and } \mathbf{C} = c\mathbf{1}_{[3x3]}, \tag{16}$$

where $\mathbf{1}_{[3x3]}$ is the 3×3 identity matrix. This gives a realistic model where the mass and damping are the same in all directions.

The admittance model inputs a force $\mathbf{f}$ and outputs a velocity $\dot{\mathbf{x}}$, which is computed as follows. The acceleration is determined from (15) to be

$$\ddot{\mathbf{x}}(t) = \mathbf{M}^{-1}\left(\mathbf{f} - \mathbf{C}\dot{\mathbf{x}}(t)\right). \tag{17}$$

The velocity at the next time step is then computed as

$$\dot{\mathbf{x}}(t+\Delta t) = \dot{\mathbf{x}}(t) + \int_t^{t+\Delta t} \ddot{\mathbf{x}}(u)du, \tag{18}$$

yielding

$$\dot{\mathbf{x}}(t+\Delta t) = \dot{\mathbf{x}}(t) + \int_t^{t+\Delta t} \mathbf{M}^{-1}(\mathbf{f} - \mathbf{C}\dot{\mathbf{x}}(u))du. \tag{19}$$

The Cartesian velocity computed by the admittance model is then used to control the mechanism. The Jacobian matrix $\mathbf{J}$ calculated from the Cartesian position $\mathbf{p}$ of the platform is used to compute the desired joint velocities $\dot{\boldsymbol{\rho}}_d$, as per (3) to obtain :

$$\dot{\boldsymbol{\rho}}_d = \mathbf{J}\left(\dot{\mathbf{x}}(t) + \int_t^{t+\Delta t} \mathbf{M}^{-1}(\mathbf{f} - \mathbf{C}\dot{\mathbf{x}}(u))du\right). \tag{20}$$

The low-level controller G_c use to command the actuators is, in this case, a closed-loop; high-gain; proportional velocity controller, namely

$$\boldsymbol{\tau} = K_p(\dot{\boldsymbol{\rho}}_d - \dot{\boldsymbol{\rho}}), \tag{21}$$

where $\dot{\boldsymbol{\rho}}_d$ is the vector of desired joint velocities and $\dot{\boldsymbol{\rho}}$ is the vector of actual joint velocities.

The control signals to the actuators are then computed from (20) and (21) as

$$\boldsymbol{\tau} = K_p\left(\mathbf{J}\dot{\mathbf{x}}(t) + \mathbf{J}\int_t^{t+\Delta t} \mathbf{M}^{-1}(\mathbf{f} - \mathbf{C}\dot{\mathbf{x}}(u))du - \dot{\boldsymbol{\rho}}\right). \tag{22}$$

This control law can specify negative tension (compression) to the actuator, which is not feasible by a cable-driven parallel mechanism. To circumvent this limitation, a saturation is placed on $\boldsymbol{\tau}$ in order to keep a minimum tension in the cable, namely

$$\boldsymbol{\tau}_s = \begin{cases} \boldsymbol{\tau} & \text{if } \boldsymbol{\tau} > \boldsymbol{\tau}_{min} \\ \boldsymbol{\tau}_{min} & \text{if } \boldsymbol{\tau} \leq \boldsymbol{\tau}_{min} \end{cases}, \tag{23}$$

where $\boldsymbol{\tau}_{min}$ is the minimum admissible tension in the cables and is adjusted experimentally.

B. Virtual Environment

The virtual environment is the part of the control which generates the forces acting on the virtual object represented by the user's moving platform in the real world. According to our model of the virtual environment, the object evolves in either of three modes. The first mode occurs when the object is moved freely by the user in the workspace. The second and third modes occur respectively when the object is in contact with a smooth surface and a smooth edge in the virtual environment. Each of these situations has its own component of the resultant force $\mathbf{f}$, i.e.,

$$\mathbf{f} = \mathbf{f}_s + \mathbf{f}_g + \mathbf{f}_c \tag{24}$$

where $\mathbf{f}_s$, $\mathbf{f}_g$ and $\mathbf{f}_c$ are respectively the force reported by the sensor, the non-contact external forces and the contact forces.

1) Free Space Motion: Forces acting on the object while it is freely moving are forces induced by the user and non-contact external forces like gravity. Those are represented in (24) by the terms $\mathbf{f}_s$ and $\mathbf{f}_g$, respectively. The force $\mathbf{f}_s$ is that measured by a force sensor. For this approach to work, low inertias of the sensor and handle are essential, so that the sensor does not pick up inertial forces. If this is not the case, a dynamic model of the handle would be necessary to account for their effects. External forces are forces that act in a constant direction in the virtual environment and that can be defined by a static vector $\mathbf{f}_g$. A low damping coefficient c is normally given to the model to insure system stability, unless a specific scenario requires it otherwise. The mass m_p of the model has to correspond to that of the moving object in the scenario of the virtual environment.

2) Contact with a Smooth Surface: To generate a contact force $\mathbf{f}_c$, a contact situation must be defined. For point motion in 3D space, a point of contact and a normal direction are necessary to define a contact without friction.

Consider a smooth surface defined by $g(\mathbf{p}) = 0$, where $g(\mathbf{p}) < 0$ represents the interior of the object and $\mathbf{n}$ a unit vector normal to its surface. The contact force is defined by two aspects : the normal $\mathbf{n}$ to the plane and the distance from P to the surface of the virtual object. The contact occurs when $g(\mathbf{p}) < 0$. The resulting force $\mathbf{f}_c$ is calculated as a proportion of the penetration distance in the direction normal to the plane, that is.

$$\mathbf{f}_c = \begin{cases} 0 & \text{if } g(\mathbf{p}) \geq 0, \\ -K_c\mathbf{n}\, g(\mathbf{p}) & \text{if } g(\mathbf{p}) < 0. \end{cases} \tag{25}$$

where K_c is a positive gain defining the stiffness of the wall and which usually takes a high value for rigid contacts.

As discussed in [18], this method could easily create instability and energy leaks due to quantization effects for a high gain K_c. A damping coefficient needs to be added to limit this effect. The damping is added only in the direction normal to the contacted surface and only while $g(\mathbf{p}) < 0$, so as to keep a consistent rendering in the other directions. This damping is added to the damping coefficient of (16),

which becomes

$$\mathbf{C} = c\mathbf{1}_{[3\times3]} + \mathbf{n}\mathbf{n}^T c_w \tag{26}$$

where c_w is a damping gain. This approach has the second benefit of allowing to adjust the degree of kinetic restitution of collisions, since it is equal to $c + c_w$.

3) Contact with Sharp Edges: When an object is described by a patchwork of smooth surfaces connected by sharp edges, contacts at the edges are not properly defined by the model of Section IV-B.2 and its direct application can result in poor haptic immersion. This occurs because the sole normal $\mathbf{n}$ of the closest surface is used. This effect is apparent when contacts are made with a virtual cube. The normal to each face is defined as pointing outward of the cube. In this manner, a contact with the cube occurs when all the distance terms are smaller than or equal to zero. The contact force is defined in the same manner as in (25). A degradation of the haptic feedback may arise for this kind of situation, as depicted by the trajectory shown in Fig. 6(a). As the distance to the vertical face becomes smaller than the distance to the horizontal face, the contact force $\mathbf{f}_c$ suddenly changes direction. This instantaneous rotation of 90° of $\mathbf{f}_c$ is perceived by the user as an artificial push away from the vertical face. This unrealistic behaviour should thus be avoided.

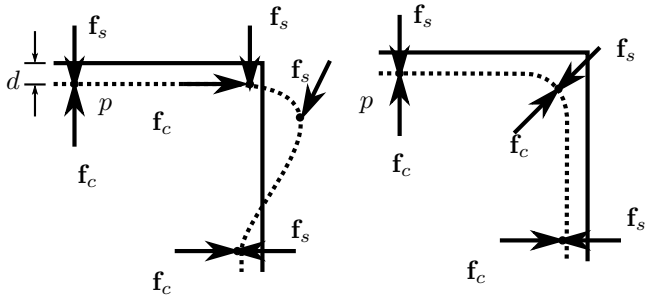
To correct this undesirable effect, we propose to adjust the direction of the normal $\mathbf{n}$. Instead of taking the normal to the plane closest to point P, each normal to the cube is taken into account and weighted according to the inverse of the distance to its corresponding plane, that is

$$\boldsymbol{\eta} = \sum_{k=1}^{6} \frac{\mathbf{n}_k}{\mid d_k \mid^p}, \tag{27}$$

where the exponent p defines the smoothness of the edges. The adjusted normal is then normalised,

$$\mathbf{n} = \frac{\boldsymbol{\eta}}{\parallel \boldsymbol{\eta} \parallel_2}, \tag{28}$$

to be used in (25) to generate $\mathbf{f}_c$. A trajectory similar to that of Fig. 6(b) can then be obtained, which matches more closely what a user would expect when touching a sharp edge.



(a) Without adjusted normal (b) With adjusted normal

Fig. 6: Trajectories around a sharp corner from left to right.

V. EXPERIMENTAL RESULTS

Experiments were conducted on the mechanism of Fig. 1 with the admittance control scheme described above. A series of scenarios are presented to show the capabilities of this haptic feedback device.

First, as a characterization of the device bandwidth, it was subjected to a sinusoidal velocity command of increasing frequency, without any human intervention. The input command was well followed until 10 Hz, at which point oscillations caused by unconstrained end-effector rotations become noticeable. These effects are minimised when a person is holding the platform, but remain the limiting factor in the admissible bandwidth.

Also, a trajectory in a virtual environment containing different objects representing challenges for haptic rendering is accomplished. It contains a portion of free motion, a box for contacts with planes and sharp edges, and a sphere for contacts with smooth surfaces. All surfaces are rigid, which mean that their virtual stiffness is as high as possible without instabilities. The trajectory as reconstructed from the encoder measurements is presented in Fig. 7.

The quality of a haptic rendering is hard to quantify. Qualitatively, it can be mentioned that the rendering provided by the mechanism is excellent. A low mass m_p and damping coefficient c allow the platform to move with minimal forces. The contacts are sharp and convincing, without vibration on the surfaces. A degradation of the rendering occurs only when pressing so hard on a surface that the limits of the actuators are reached. In such a case, the platform starts to penetrate the virtual surface. A video attached to this article presents the mechanism being manipulated along a trajectory similar to that presented in Fig. 7.

The behaviour of the haptic-feedback device was also compared against that of the real word in a simple experiment. This experiment consists in reproducing a ball bouncing on a hard surface. In a virtual environment artificially subjected to gravity, the object was dropped on a virtual horizontal plane. The contact between the object and the horizontal plane is modeled as a spring and damper. The object

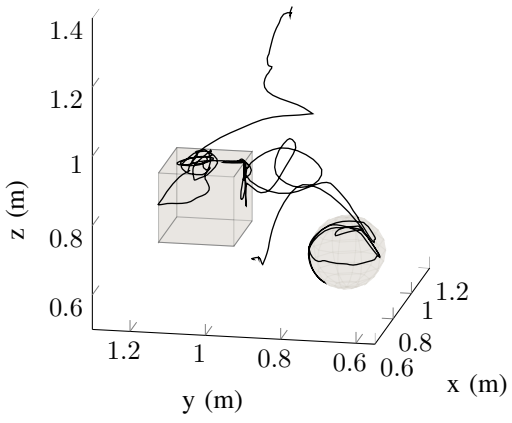


Fig. 7: Trajectory of user's platform in a virtual environment (lengths in metres).

mass, the contact stiffness and the contact damping were adjusted to resemble those of a bouncing ball. The object trajectory was recorded through the encoder measurements. The height of the object is traced against time in Fig. 8. In a separate experiment, the height of each bounce of a real freely-bouncing ball were recorded and are also traced in Fig. 8.

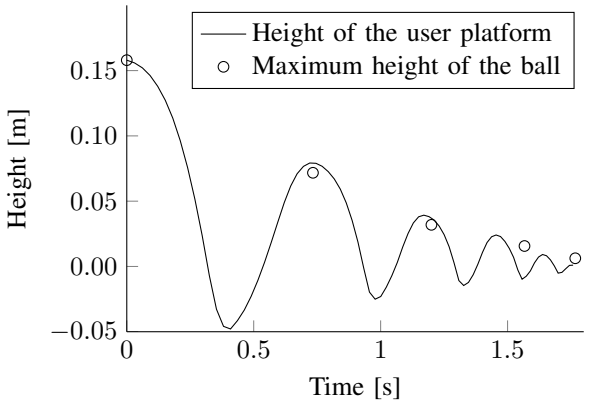


Fig. 8: Height of the user platform while bouncing in a virtual environment in comparison to the bouncing height of a ball in the real world.

As shown on this graph, the mechanism closely follows the real-world ball for the first bounce, but diverges slowly from the real trajectory in subsequent bounces. This result is expected because the control scheme does not guarantee a perfect replication of the physics of the real world, but focuses on smooth and stable rendering. Nevertheless, the mechanism follows the trajectory of a real object closely enough to avoid disrupting the immersion of the user. A video of this experiment shows both objects at the same time and demonstrates this life-like behaviour.

VI. CONCLUSIONS

A control law based on an admittance scheme and closed-loop velocity control was presented and proven to provide a realistic haptic rendering on the three-degree-of-freedom cable-driven mechanism presented herein. Also, a simple self-calibration method based on measurement redundancy was proven effective in an experiment providing a relative error from one to two percent on the mechanism geometry. Future work will include adjusting the low-level controller G_c to allow a wider dynamic bandwidth by, for example, including a feed-forward term taking into account inertial forces. Other improvements include the possibility of interacting with moving and movable objects in the virtual environment, as well as adding four cables to increase the number of degrees of freedom to six. Having six degrees of freedom will allow the platform to generate torque, extending the system to more complex virtual objects and collisions. The last step will consist in placing the mechanism inside a CAVE environment, to immerse the user into a full 3D both visual and haptic feedbacks.

ACKNOWLEDGMENT

The authors would like to thank Boris Mayer-St-Onge, Thierry Laliberté and Simon Foucault for their help in building the mechanism presented in this paper.

REFERENCES

[1] (2013, Sep.) Occulus VR^{TM}. [Online]. Available: http://http://www.oculusvr.com/

[2] T. H. Massie and K. J. Salisbury, "PHANToM haptic interface: a device for probing virtual objects," in *Proceedings of the 1994 International Mechanical Engineering Congress and Exposition*, ser. Dynamic Systems and Control Division, vol. 55-1. Massachusetts Inst of Technology, Cambridge, United States: ASME, 1994, pp. 295–299.

[3] L. L. Cone, "Skycam: an aerial robotic camera system," *BYTE*, vol. 10, pp. 122–132, 1985.

[4] G. Meunier, B. Boulet, and M. Nahon, "Control of an overactuated cable-driven parallel mechanism for a radio telescope application," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1043–1054, 2009.

[5] R. Lindemann and D. Tesar, "Construction and Demonstration of a 9-String 6-DOF Force Reflecting Joystick for Telerobotics," in *NASA International Conference on Space Telerobotics*, 1989, pp. 55–64.

[6] M. Ishii and M. Sato, "A 3D interface device with force feedback: a virtual work space for pick-and-place tasks," in *Proceedings of IEEE Virtual Reality Annual International Symposium*. Seattle, WA, USA: IEEE, 1993, pp. 331–5.

[7] R. Xiao, J. Huang, and P. Wu, "Large space force feedback system based on string," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. vol.4. Harbin, China: IEEE, Dec. 2011, pp. 2705–2708.

[8] Z. Chen, Y. Zhang, D. Wang, C. Li, and Y. Zhang, "iFeel6-BH1500: a large-scale 6-DOF haptic device," in *2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS)*. Tianjin, China: IEEE, 2012, pp. 121–125.

[9] P. Gallina, G. Rosati, and A. Rossi, "3-d.o.f. Wire Driven Planar Haptic Interface," *Journal of Intelligent and Robotic Systems*, vol. 32, no. 1, pp. 23–36, 2001.

[10] R. L. Williams II, "Cable-Suspended Haptic Interface," *International Journal of Virtual Reality*, vol. 3, pp. 13–21, 1998.

[11] T. Morizono, K. Kurahashi, and S. Kawamura, "Realization of a virtual sports training system with parallel wire mechanism," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, no. April. Albuquerque, NM, USA: IEEE, 1997, pp. 3025–3030.

[12] C. Carignan and K. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *Haptics-e*, vol. 1, no. 2, pp. 1–14, 2000.

[13] C. Gosselin, R. Poulin, and D. Laurendeau, "A Planar Parallel 3-DOF Cable-Driven Haptic Interface," in *12th World Multi-Conference on Systemics, Cybernetics and Informatics*, vol. 3, Orlando, FL, USA, 2008, pp. 266–271.

[14] P. Miermeister and A. Pott, "Auto calibration method for cable-driven parallel robots using force sensors," *Latest Advances in Robot Kinematics*, pp. 269–276, 2012.

[15] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "NIMS-PL: A Cable-Driven Robot With Self-Calibration Capabilities," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1005–1015, 2009.

[16] J. Alexandre Dit Sandretto, D. Daney, M. Gouttefarde, and C. Baradat, "Calibration of a fully-constrained parallel cable-driven robot," INRIA, Tech. Rep. RR-7879, 2012.

[17] M. Otis, V. Duchaine, G. Billette, S. Perreault, C. Gosselin, and D. Laurendeau, "Cartesian Control of a Cable-Driven Haptic Mechanism," in *Advances in Haptics*, Mehrdad Hosseini Zadeh, Ed. Intech, 2010, ch. 4, pp. 75–103.

[18] R. B. Gillespie and M. R. Cutkosky, "Stable user-specific haptic rendering of the virtual wall," in *Proceedings of the ASME International Mechanical Engineering Congress and Exhibition*, vol. 58, 1996, pp. 397–406.