

Rendering of Constraints with Underactuated Haptic Devices

Daniel Lobo and Miguel A. Otaduy

Abstract—Several previous works have studied the application of proxy-based rendering algorithms to underactuated haptic devices. However, all these works make oversimplifying assumptions about the configuration of the haptic device, and they ignore the user's intent. In this work, we lift those assumptions, and we carry out a theoretical study that unveils the existence of unnatural ghost forces under typical proxy-based rendering. We characterize and quantify those ghost forces. In addition, we design a novel rendering strategy, with anisotropic coupling between the device and the proxy. With this strategy, the forces rendered by an underactuated device are a best match of the forces rendered by a fully actuated device. We have demonstrated our findings on synthetic experiments and a simple real-world experiment.

Index Terms—Haptic rendering, underactuated haptics.

1 INTRODUCTION

UNDERACTUATED haptic devices enjoy multiple degrees of freedom (DoFs), but provide force feedback only on a portion of those DoFs. Underactuation allows for a high-dimensional configuration space without the need to equip all DoFs with actuation mechanisms, which translates into a small form factor at the price of limited feedback capabilities. In practice, underactuation has been a popular strategy for the design of haptic devices, including stylus devices [1] or exoskeletons [2], [3].

However, the design of appropriate rendering algorithms has received considerable less attention. In this paper, we carry out a theoretical analysis of the classic proxy-based strategy for rendering virtual walls [4], [5], when applied to underactuated devices. Proxy-based rendering decouples the design of stable interaction from the simulation of device interaction with the virtual environment [6], and can be applied to arbitrary devices and virtual environment interactions [7].

Previous works already identified passivity issues due to naïve application of proxy-based strategies to underactuated devices [8], [9]. But we have found that proxy-based rendering can suffer even more severe problems, which to our knowledge have not even been documented to date. Proxy-based rendering produces unwanted ghost forces on underactuated devices, which result in very unnatural motion when exploring virtual walls.

Take for example the 2D scenario shown in Fig. 1, with a haptic device with one active DoF and one passive DoF. The user, in yellow, tries to move the device from point A to point B, and collides on his/her way with a slanted virtual wall at W. With a fully actuated device, shown in blue, proxy-based rendering produces forces normal to the wall, and the device follows the user's intent subject to the wall constraint, reaching point F. With an underactuated device, one would expect the device to behave optimally as shown

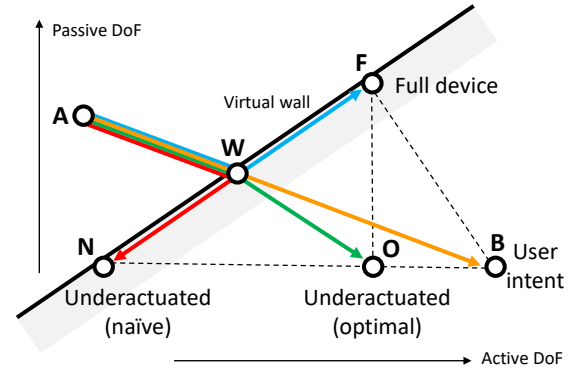


Fig. 1. When a user (in yellow) aims to penetrate a virtual wall, fully actuated rendering produces the response in blue, where the haptic device slides along the wall. We have found that, instead, naïve underactuated rendering produces the response in red, with unnatural ghost forces that make the device slide in the completely opposite direction. We have also designed an optimal underactuated rendering algorithm that produces the response in green, where the forces along actuated DoFs match exactly those of the fully actuated case.

in green, reaching point O, i.e., on the active DoF the motion of the device is the same as in the fully actuated case, and on the passive DoF the motion is dictated by the user. However, naïve proxy-based rendering makes the device slide along the virtual wall, hence it can never reach the optimal point O. Due to underactuation, the motion of the passive DoF is dictated by the user, while the motion of the active DoF respects the virtual wall, and moves the device as shown in red, reaching point N. In practice, underactuated proxy-based rendering produces forces that are not normal to the wall constraint. The user experiences this effect as ghost forces that may even drive the device away from his/her intent, as shown in the example in Fig. 1.

In our analysis, we have quantified the ghost forces of naïve proxy-based rendering strategies as a function of rendering parameters. Interestingly, we have found that ghost forces can be more severe under higher render-

• D. Lobo and M. Otaduy are both with Universidad Rey Juan Carlos, Madrid, Spain.

Preprint. Author version.

ing impedance, hence they compete with rendering transparency. Our analysis is based on a fundamental difference with all previous work. When studying the rendering algorithm, we do not consider the configuration of the haptic device as an independent variable, i.e., a known given input. Instead, we account for the user's intent as an independent variable, and study the device configuration as a variable that depends on the rendered forces and the user's intent together. One may wonder why is this additional complexity not necessary when analyzing proxy-based rendering for fully actuated devices, why can the device configuration be safely considered an independent variable. The reason is that, in fully actuated proxy-based rendering, rendered forces are workless, normal to the constraints, hence they only stop the device. In underactuated rendering, on the other hand, rendered forces may actually move the device.

Based on our analysis, we also provide a solution for optimal underactuated rendering. This solution relies on the use of anisotropic impedance in the coupling of the proxy and the device. In the paper, we provide a simple way to estimate optimal rendering parameters. We report synthetic experiments together with a simple real-world experiment where we have applied our solution for optimal underactuated rendering, and our results validate the theoretical findings.

Practical underactuated devices exhibit sources of complexity not covered in our initial analysis, such as nonlinear mappings to and from configuration space and operational space (e.g., devices with rotational DoFs), or multiple actuated DoFs with diverse geometric and mechanical characteristics. We believe that this work is just a first step toward the design of rendering algorithms for underactuated haptics, and our theoretical findings deserve further extension to address the complexity of practical settings.

2 RELATED WORK

Building underactuated haptic devices is a popular choice, due to the attractive form factor of the resulting devices, while retaining at least some force feedback capabilities. With impedance-type backdrivable designs, the non-actuated but useful workspace of the device can be quite rich despite a small number of actuated DoFs.

Stylus-type haptic devices have been one of the most popular forms of underactuated devices, with actuated 3-DoF translation and non-actuated 3-DoF rotation (i.e., translation and rotation sensing, with force—but no torque—actuation). There has been important effort on the analysis of such underactuated haptic devices, in particular from the point of view of rendering performance with and without torque feedback. Verner and Okamura [10] ran experimental analysis of the performance gain provided by torque feedback over force-only feedback, which serves to quantify the practical impact of full actuation vs. underactuation. Weller and Zachmann [11] further confirmed the performance gain of full actuation vs. underactuation, but they did not consider how to optimize the latter.

Some works have looked at ways to improve performance and overall perception to avoid the limitations of underactuation. Lecuyer et al. [12] designed a trick such that, when a virtual wall is explored, the virtual environment is

rotated to align the wall's normal with the actuated DoFs. Kadleček et al. [13], instead, used sensory substitution and pseudo-haptic feedback to simulate torque feedback.

Barbagli and Salisbury [8] were the first to formalize the analysis of underactuation. They formulated the controllability and observability of haptic devices in terms of the rank of the subspace of the device workspace that can be effectively controlled or observed. They also pointed out that the interaction with virtual environments through underactuated devices may not be energy-conservative (i.e., may not be passive). To this end, they analyzed the work done against a virtual wall rendered using a proxy strategy with an underactuated device. Verner and Okamura [9] extended the analysis of Barbagli and Salisbury, and they also reached similar conclusions about the lack of passivity of naïve proxy-based rendering strategies. However, in both these works, the authors made two oversimplifying assumptions: (i) that the force rendered by an underactuated device is simply the projection of the full force to the actuated DoFs, and (ii) that the motion of the device can be considered imposed, not a result of the rendering algorithm. In our work, we analyze the effects of underactuation using a more accurate model of the device and the rendering algorithm, which unveils the existence of unnatural ghost forces. The effect of these ghost forces can be more severe than the lack of passivity, and was completely missed by previous works.

Interestingly, Kadleček [14] documented the existence of ghost forces on stylus-type haptic devices with 3 actuated (translation) and 3 non-actuated (rotation) DoFs. He discussed how the user's intent to rotate against a virtual wall could produce a large and unnatural translational force. Nevertheless, his work does not make an attempt to characterize mathematically the ghost forces or solve them through a fundamental redesign of the rendering algorithm.

Given the limitations of underactuation, it is possible to design optimization strategies that favor different aspects when computing underactuated feedback subject to some target fully actuated feedback. Luecke [15] designed a rendering algorithm that considers the error between target forces or positions and those actually rendered by the device. Then, it applies a PD control strategy on these error, which defines the actual output of the device. Meli and Prattichizzo [16] studied methods to render contact forces through an underactuated device, while maximizing task performance. For each task, they defined an optimality criterion in the selection of underactuated feedback.

More recently, Lobo et al. [17] have considered the mapping to and from operational space and configuration space to define optimal underactuated rendering. They define proxies in both spaces to account for underactuation subject to the space mapping, and find conditions for passivity under certain constraints. Sarac et al. [18] have compared task performance when optimizing underactuated feedback according to target forces or target positions. In one of their comparisons, they separate the user's intent from the actual trajectory of the device. We find that this is crucial for understanding ghost forces, but they miss to analyze this aspect.

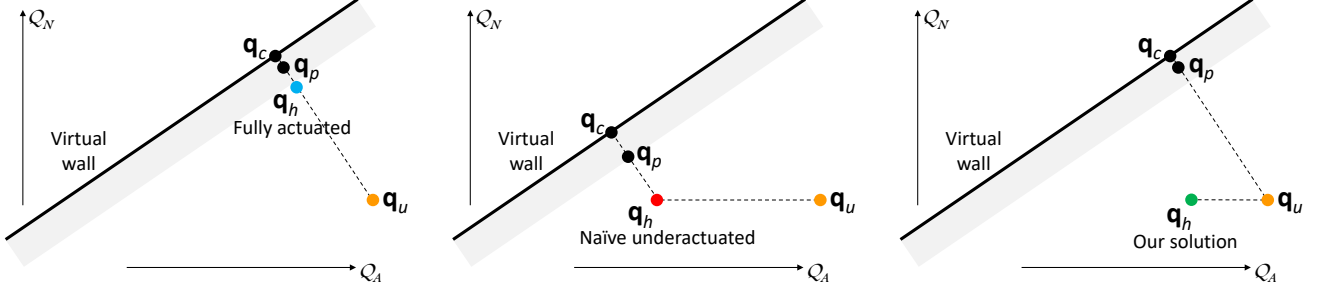


Fig. 2. Device configuration space, showing the elements that participate in our analysis of proxy-based constraint rendering: user goal \mathbf{q}_u , haptic device \mathbf{q}_h , proxy \mathbf{q}_p , and the constraint projection \mathbf{q}_c . The images show different equilibrium configurations for three different rendering settings, under the same user's intent. In the fully actuated case (left), the proxy and the device are aligned with the user's intent along the constraint normal. We wish the rendered forces in the underactuated case to match the forces of the fully actuated case on the actuated subspace \mathcal{Q}_A . These forces depend on the deviation between the device and the proxy. With underactuation, in the naïve case (middle), the proxy slides along the constraint while the underactuated DoFs of the device match the user's intent. In this situation, the rendered forces are larger than in the fully actuated case; they are perceived as ghost forces that drive the device away from the user's intent. With our solution (right), the rendered forces match the forces of the fully actuated case on the actuated subspace \mathcal{Q}_A . This is because the deviation between the device and the proxy along the actuated subspace is preserved. We achieve this by tuning the tracking impedance between the proxy and the device, such that the proxy is aligned with the user's intent along the constraint normal.

3 PRELIMINARIES

To understand the source of ghost forces in naïve underactuated rendering and lay out our solution, we first need to provide some definitions of the main elements present in our analysis. The reader may refer to Fig. 2 for a schematic visualization of these main elements. In the analysis, we make some assumptions that facilitate the derivations, but they do not limit the generality of our conclusions. This assumptions are: (i) The configuration and operational spaces of the haptic device are the same; (ii) all the actuated and non-actuated DoFs have the same geometric and mechanical characteristics; and (iii) the user and the device are in quasi-static equilibrium, with no dynamic oscillations.

3.1 Actuated and Non-Actuated DoFs

We denote as $\mathcal{Q} = \mathcal{Q}_A \times \mathcal{Q}_N$ the configuration space of the device, with \mathcal{Q}_A the actuated configuration space (resulting from actuated DoFs), and \mathcal{Q}_N the non-actuated configuration space (resulting from non-actuated DoFs). We characterize the actuated and non-actuated configuration spaces through orthonormal bases \mathbf{A} and \mathbf{N} , respectively, with $\mathbf{A}^T \mathbf{N} = 0$. $\mathbf{q}_h = \mathbf{A} \mathbf{q}_h^A + \mathbf{N} \mathbf{q}_h^N \in \mathcal{Q}$ represents a device state, with $\mathbf{q}_h^A \in \mathcal{Q}_A$ the state of actuated DoFs, and $\mathbf{q}_h^N \in \mathcal{Q}_N$ the state of non-actuated DoFs.

In Fig. 2, we show a two-dimensional representation of the possibly high-dimensional configuration space of a device, with actuated and non-actuated subspaces represented as orthogonal axes. In such high-dimensional configuration space, each configuration of the haptic device is depicted as a point.

3.2 User Action

A very important element of our analysis is to account for the user's intent. To understand whether a proposed haptic rendering algorithm is effective at opposing the user's motion against constraints, we first need to characterize the user's intent. In our analysis, we denote as ghost forces the forces produced by the rendering algorithm that pull the device in a direction opposite to the user's intent.

As it will become evident, we find it convenient to characterize the user's intent in terms of a goal position $\mathbf{q}_u \in \mathcal{Q}$ and a force dependent on the deviation between the goal position and the current device position. With a user-intent impedance \mathbf{K}_u , the user's force on the device is

$$\mathbf{f}_u = \mathbf{K}_u (\mathbf{q}_u - \mathbf{q}_h). \quad (1)$$

In our analysis, we consider the user-intent impedance to be isotropic, i.e., $\mathbf{K}_u = k_u \mathbf{I}$.

3.3 Proxy-Based Rendering of Constraints

Let us define a virtual wall as a constraint surface $\mathcal{C} \subset \mathcal{Q}$. Without loss of generality, we consider a linearized constraint defined by a point \mathbf{o} and a unit normal vector \mathbf{n} . We wish to render haptic feedback that prevents (or minimizes) interpenetration of the device with the constraint. By providing forces that are orthogonal to the constraint, the user is able to slide passively along the constraint, i.e., with no work. Of course, tangential dissipative forces can be added to model friction, but we omit them in our analysis, for simplicity.

We assume a proxy-based haptic rendering algorithm. The use of a proxy allows stable rendering by decoupling the impedance rendered to the user through the device from the impedance of constraint satisfaction [6], and it also simplifies the generalization of the rendering algorithm to complex devices and virtual objects [7]. We denote the configuration of the proxy as $\mathbf{q}_p \in \mathcal{Q}$.

There are two main methods to compute the proxy configuration in proxy-based rendering. *Hard constraints* [4], [5], [19] compute the proxy position that is closest to the device position while satisfying the constraints, by solving an optimization problem. *Soft constraints* [20], [21] let the proxy penetrate the constraints to define penalty forces, and compute the proxy position by solving a static or dynamic mechanics problem. Our analysis considers a soft constraints method, as it simplifies the derivation, but it extends to hard constraints as the constraint stiffness tends to infinity. Under soft constraints, two forces act on the proxy: a tracking force and a constraint force.

The tracking force pulls the proxy toward the configuration of the device. In the absence of constraints, it ensures that the proxy follows the device exactly. Under constraints, it ensures that the proxy follows the device as close as possible, while sliding along the constraints. With a tracking impedance \mathbf{K}_t , the tracking force on the proxy is

$$\mathbf{f}_t = \mathbf{K}_t (\mathbf{q}_h - \mathbf{q}_p). \quad (2)$$

The constraint force pulls the proxy toward the constraint. We define the closest point to the proxy on the surface of the constraint as $\mathbf{q}_c \in \mathcal{Q}$. Based on the definition of the constraint as a point \mathbf{o} and a unit normal vector \mathbf{n} , as mentioned above, the projection of the proxy on the constraint is computed as

$$\mathbf{q}_c = \mathbf{o} + (\mathbf{I} - \mathbf{n}\mathbf{n}^T) (\mathbf{q}_p - \mathbf{o}). \quad (3)$$

With a constraint impedance \mathbf{K}_c , the constraint force on the proxy is

$$\mathbf{f}_c = \mathbf{K}_c (\mathbf{q}_c - \mathbf{q}_p). \quad (4)$$

In our analysis, we consider the constraint impedance to be isotropic, i.e., $\mathbf{K}_c = k_c \mathbf{I}$.

Once the proxy is well defined, we can also define how it produces haptic feedback through the device. We define a rendering force on the device, as a function of the deviation from the proxy. With a rendering impedance \mathbf{K}_r , the feedback force on the device is

$$\mathbf{f}_r = \mathbf{K}_r (\mathbf{q}_p - \mathbf{q}_h). \quad (5)$$

This force acts in the direction opposite to the tracking force. However, note that impedances \mathbf{K}_r and \mathbf{K}_t need not be the same. This is evident in the underactuated case, where the rendering impedance \mathbf{K}_r is obviously null for the non-actuated DoFs.

3.4 Equilibrium Equations

In all our analysis, we consider that the device and proxy are in quasi-static equilibrium, hence we omit dynamics effects. Then, all impedance values defined above boil down to stiffness terms.

Given the forces acting on the device and the proxy, as defined above, we can formulate their quasi-static equilibrium simply as:

$$\text{proxy: } \mathbf{f}_c + \mathbf{f}_t = 0, \quad (6)$$

$$\text{device: } \mathbf{f}_u + \mathbf{f}_r = 0. \quad (7)$$

In our subsequent analysis, we will solve these equilibrium equations for different impedance configurations, namely fully actuated rendering, naïve underactuated rendering, and our proposed solution. We will compute the positions of the proxy and the device, and thereby the rendered forces, to analyze and eliminate ghost force effects.

4 GHOST FORCES

In this section, we analyze the outcome of standard proxy-based rendering methods when applied to underactuated haptic devices. But before facing this analysis, we compute the outcome of fully actuated rendering, which will be used as a baseline for comparisons. We demonstrate that naïve

underactuated rendering suffers from ghost forces, and we quantify those forces as a function of the various algorithm impedance settings. Remarkably, we show that the ghost forces can be even more severe as the transparency of the rendering algorithm grows.

4.1 Baseline Fully Actuated Rendering

In our analysis of the standard proxy-based rendering algorithm, we consider the tracking stiffness to be isotropic, i.e., $\mathbf{K}_t = k_t \mathbf{I}$. Furthermore, in the fully actuated case, we also consider the rendering stiffness to be isotropic, i.e., $\mathbf{K}_r = k_r \mathbf{I}$. In this way, the various DoFs are decoupled and the stiffness is the same along all of them.

Under isotropic stiffness, the solution to the equilibrium equations (6) and (7) yields proxy and device positions that align with the user goal along the normal to the constraint, as shown in Fig. 2-left. The projection of the proxy on the constraint, \mathbf{q}_c , is also the projection of the user goal. It is convenient to compute the distance from the user goal to the constraint,

$$d = \mathbf{n}^T (\mathbf{o} - \mathbf{q}_u). \quad (8)$$

By solving the equilibrium equations, we obtain the positions of the proxy and the device, which depend on the ratios of the various stiffness values:

$$\mathbf{q}_p = \mathbf{q}_u + \frac{k_c k_u + k_c k_r}{k_c k_r + (k_t + k_c) k_u} d \mathbf{n}, \quad (9)$$

$$\mathbf{q}_h = \mathbf{q}_u + \frac{k_c k_r}{k_c k_r + (k_t + k_c) k_u} d \mathbf{n}. \quad (10)$$

In our analysis of ghost forces, we will study the component of the rendering force (5) along actuated DoFs. As a baseline, in the fully actuated case this force component is:

$$\mathbf{A}^T \mathbf{f}_r = \frac{k_u}{1 + \frac{k_u}{k_r} \left(1 + \frac{k_t}{k_c}\right)} d \mathbf{A}^T \mathbf{n}. \quad (11)$$

Note that, if the device were stopped fully at the constraint, the user's force on the device would be $-k_u d \mathbf{A}^T \mathbf{n}$. And we can see from the expression (11) that the rendered force opposes this value as $k_c \rightarrow \infty$ and $k_r \rightarrow \infty$; therefore, the rendered force succeeds at exactly counteracting the user's intent. In that situation, the rendering algorithm maximizes its transparency, and is able to render perfectly a virtual wall. In practice, it is possible to model fully or almost hard constraints on the proxy (i.e., $k_c \rightarrow \infty$); however, the rendering stiffness is limited by stability.

4.2 Naïve Underactuated Rendering

Now we are ready to analyze an underactuated setting. This setting can be modeled mathematically by making the rendering stiffness null along non-actuated DoFs, i.e.,

$$\mathbf{K}_r = \begin{pmatrix} \mathbf{A} & \mathbf{N} \end{pmatrix} \begin{pmatrix} k_r \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{A}^T \\ \mathbf{N}^T \end{pmatrix} = k_r \mathbf{A} \mathbf{A}^T. \quad (12)$$

A naïve rendering strategy is to apply the same proxy-based rendering algorithm as in the fully actuated setting, with all other impedances unchanged. Then, the positions

of the proxy and the device are defined by the same equilibrium conditions (6) and (7) as for the fully actuated setting. And they can be computed as:

$$\mathbf{q}_p = \mathbf{q}_u + \frac{k_c k_u \mathbf{I} + k_c k_r \mathbf{A} \mathbf{A}^T}{k_c k_r \mathbf{n}^T \mathbf{A} \mathbf{A}^T \mathbf{n} + (k_t + k_c) k_u} d \mathbf{n}, \quad (13)$$

$$\mathbf{q}_h = \mathbf{q}_u + \frac{k_c k_r}{k_c k_r \mathbf{n}^T \mathbf{A} \mathbf{A}^T \mathbf{n} + (k_t + k_c) k_u} d \mathbf{A} \mathbf{A}^T \mathbf{n}. \quad (14)$$

As an evident consequence of underactuation, the position of the device along non-actuated DoFs matches exactly the position of the user goal, i.e., $\mathbf{N}^T (\mathbf{q}_h - \mathbf{q}_u) = \mathbf{0}$. With isotropic tracking and constraint impedance, the equilibrium condition on the proxy aligns the device and the proxy along the constraint normal. The actual positions of the device and the proxy are determined by the ratios of the various stiffness values, and produce a configuration such as the one shown in Fig. 2-middle. Unlike the fully actuated case, the device slides along the constraints until its non-actuated DoFs match those of the user goal.

We can also compare the results vs. fully actuated rendering in terms of the rendering force along actuated DoFs (11):

$$\mathbf{A}^T \mathbf{f}_r = \frac{k_u}{\mathbf{n}^T \mathbf{A} \mathbf{A}^T \mathbf{n} + \frac{k_u}{k_r} \left(1 + \frac{k_t}{k_c}\right)} d \mathbf{A}^T \mathbf{n}. \quad (15)$$

We can see that the expression differs only in the term $\mathbf{n}^T \mathbf{A} \mathbf{A}^T \mathbf{n}$ in the denominator. Recall also that, for highly transparent rendering, we aim at high constraint and rendering stiffness values, i.e., $k_c \rightarrow \infty$ and $k_r \rightarrow \infty$. Then, the difference with (11) can be devastating. If the constraint is close to parallel to the actuated DoFs, i.e., $\mathbf{A}^T \mathbf{n} \rightarrow 0$, the rendered force can grow unbounded.

The user expects a reaction of the device aligned with the normal to the constraint, but experiences instead a ghost force. Instead of simply counteracting the user's intent, the rendered force slides the device along the constraint away from the user goal, as discussed earlier and also shown schematically in Fig. 1. All previous work missed to identify this ghost-force effect produced by underactuated rendering, and considered that the configuration of the device could be dictated by the user.

5 ANISOTROPIC TRACKING IMPEDANCE

In this section we elaborate our solution for underactuated rendering, which consists in making the tracking impedance anisotropic. The degree of impedance anisotropy, i.e., the ratio between actuated and non-actuated tracking impedances, will affect the quality of underactuated rendering, and we seek the ratio that produces optimal rendering. We start by defining optimality conditions for underactuated rendering, by analyzing the baseline fully actuated rendering case. Next, we apply the optimality conditions to compute optimal anisotropic tracking impedance.

5.1 Conditions for Optimal Underactuated Rendering

Let us recall the baseline fully actuated rendering described in Section 4.1. Under isotropic stiffness for all forces, the user goal, the device, and the proxy are all aligned along

the direction \mathbf{n} normal to the constraint, as shown in Fig. 2-left.

If some DoFs are underactuated, on the other hand, the device cannot provide rendering forces along those underactuated DoFs, and its motion is completely dictated by the user goal. Moreover, as shown in Section 4.2, with naïve underactuated rendering, the rendering force along actuated DoFs may differ strongly from the fully actuated case. This is experienced as a ghost force that pulls the device along the constraint, as shown in Fig. 2-middle.

With our optimal underactuated rendering, we wish to remove the ghost force. Then, underactuated rendering is optimal if the device matches perfectly the forces and motion of the fully actuated case along the actuated DoFs. This result is indicated schematically in Fig. 1.

But we can also characterize optimal underactuated rendering through the position of the proxy. In the fully actuated case, the proxy and the user goal are aligned along the normal to the constraint. If the constraint stiffness is high, this means that the proxy is located practically at the projection of the user goal onto the constraint. We seek an underactuated rendering solution where the proxy and the user goal are also aligned along the normal to the constraint. Then, both the proxy and the user goal are practically at the same locations in the fully actuated and underactuated cases. And it follows that, on the actuated DoFs, the position of the device and the rendered force are also practically the same. To conclude, our optimality condition for underactuated rendering is to align the proxy and the user goal along the normal to the constraint.

5.2 Optimal Anisotropic Impedance

In the underactuated setting, if the proxy and the user goal are aligned along \mathbf{n} , the device must be misaligned with them. We observe that this is possible only if the tracking impedance is anisotropic. Intuitively, if the proxy cannot impose isotropic rendering impedance on the device due to underactuation, this condition should be reflected back to the proxy, and the device should not impose isotropic tracking impedance on the proxy. On non-actuated DoFs, where the proxy itself cannot impose a rendering force on the device, the tracking impedance should be low. On actuated DoFs, where the proxy can indeed impose a rendering force, the tracking impedance should remain high.

Following our observation, let us consider an anisotropic tracking stiffness of the form

$$\begin{aligned} \mathbf{K}_t &= \begin{pmatrix} \mathbf{A} & \mathbf{N} \end{pmatrix} \begin{pmatrix} k_{ta} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_{tn} \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}^T \\ \mathbf{N}^T \end{pmatrix} \\ &= k_{ta} \mathbf{A} \mathbf{A}^T + k_{tn} \mathbf{N} \mathbf{N}^T. \end{aligned} \quad (16)$$

with k_{ta} and k_{tn} , respectively, the tracking stiffness on actuated and non-actuated DoFs.

Based on the discussion in Section 5.1 above, optimality of underactuated rendering is characterized by the alignment of the user goal and the proxy along the normal to the constraint. Then, same as in the fully actuated case, the projection of the proxy on the constraint, \mathbf{q}_c , is also the projection of the user goal. And, similarly, the distance from the user goal to the constraint, d , can be computed as in (8).

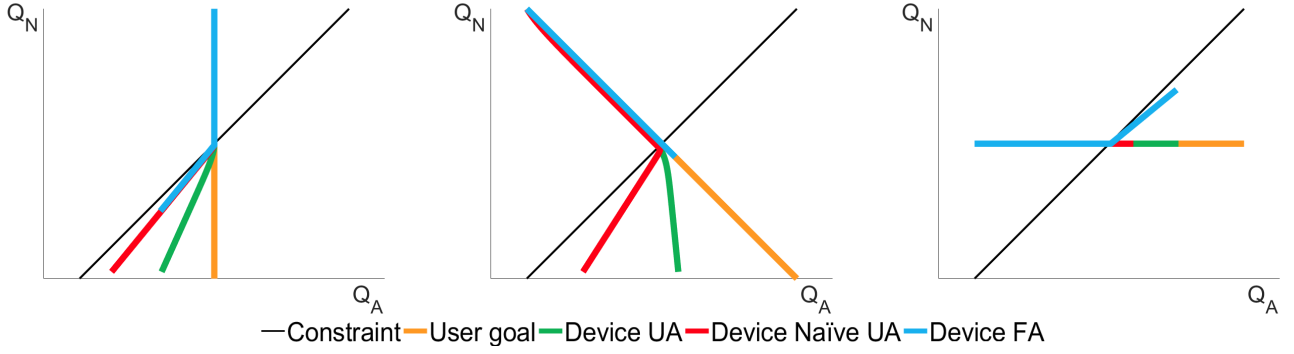


Fig. 3. Comparison of rendering strategies under various directions of user intent. On a synthetic experiment with a 2-DoF device with one actuated DoF and one non-actuated DoF, the user follows different goal trajectories against a slanted virtual wall. We compare our underactuated rendering solution (green, UA), fully actuated rendering (blue, FA), and naïve underactuated rendering (red, naïve UA). Our solution is, in all cases, optimal given the limitations of underactuation, i.e., the position of the device matches the fully actuated solution along the actuated DoF. As a result, forces on the actuated DoF also match those of the fully actuated solution. In contrast, in naïve underactuated rendering, the device moves away from the user's intent, due to ghost forces. In the example, the impedance ratio k_r/k_u is 10.

We conclude that it is possible to achieve optimal underactuated rendering if the tracking stiffness satisfies the following anisotropy ratio:

$$\frac{k_{ta}}{k_{tn}} = 1 + \frac{k_r}{k_u}. \quad (17)$$

We defer the derivation of the stiffness ratio to the Appendix.

The anisotropy ratio can be interpreted in the following way. The tracking stiffness along actuated DoFs should be higher than the tracking stiffness along non-actuated DoFs, by a factor based on the ratio between the rendering stiffness and the user action stiffness. In contrast to the fully actuated case, optimal underactuated rendering can be achieved by *softening* the tracking stiffness along non-actuated DoFs.

Given optimal tracking stiffness, and by solving the equilibrium conditions (6) and (7), we obtain the positions of the proxy and the device:

$$\mathbf{q}_p = \mathbf{q}_u + \frac{k_c}{k_{tn} + k_c} d \mathbf{n}, \quad (18)$$

$$\mathbf{q}_h = \mathbf{q}_u + \frac{k_c k_r}{(k_{tn} + k_c)(k_u + k_r)} d \mathbf{A} \mathbf{A}^T \mathbf{n}. \quad (19)$$

It is easy to validate that the proxy and the user goal are aligned along the constraint normal, while the device is misaligned, as shown in Fig. 2-right.

Based on these positions, the rendering force (5) along actuated DoFs is:

$$\mathbf{A}^T \mathbf{f}_r = \frac{k_u}{\left(1 + \frac{k_u}{k_r}\right) \left(1 + \frac{k_t}{k_c}\right)} d \mathbf{A}^T \mathbf{n}. \quad (20)$$

We see that this force is a very similar, lower estimate of the fully actuated force (11). It acts on the same direction, with a slightly lower stiffness, hence it does not produce a ghost force effect and it always counteracts the user's action. Moreover, for hard constraints $k_c \rightarrow \infty$, it is the exact same force as in the fully actuated case.

6 EXPERIMENTS

We have carried out two types of experiments. First, synthetic experiments that allow us to test our theoretical

findings on fully controlled settings. Second, a real-world experiment that confirms our findings on a real scenario. We validate that our findings persist when we lift simplifying assumptions made in the theoretical derivations, and we demonstrate that our optimal underactuated rendering does not suffer ghost forces, and hence outperforms naïve underactuated rendering in practice. In the accompanying video, we show dynamic captures of both the synthetic and real-world experiments.

6.1 Device Model for Synthetic Experiments

In our theoretical analysis of underactuated rendering, we make several simplifying assumptions about the haptic rendering problem. Notably, we consider that the device and the proxy are in quasi-static equilibrium, and the effect of haptic rendering is instantaneous. In our synthetic experiments we lift these two assumptions. We simulate rendering problems where we consider dynamics effects of the haptic device, as well as the sampled-data nature of the rendering problem.

We consider a 2-DoF haptic device, with one DoF actuated and the other one not. We model together the mass m_h and damping b_h of the device and the user, and we consider them to be constant throughout the experiment. Then, we replace the device equilibrium equation (7) with the following dynamics equation:

$$m_h \ddot{\mathbf{q}}_h(t) = \mathbf{f}_u(t) + \mathbf{f}_r(t - dt) - b_h \dot{\mathbf{q}}_h(t). \quad (21)$$

As noted in the equation, the rendering force (5) is sampled at the previous time step, to model the update rate of the rendering algorithm. The user goal position acts as input to the model, and defines the user force (1). We solve the dynamics of the device using implicit Euler integration.

To compute the proxy, we use a quasi-static model, which takes as input the device position from the previous time step. Then, we replace the proxy equilibrium equation (6) with the following quasi-static problem:

$$\mathbf{f}_c(\mathbf{q}_c(t), \mathbf{q}_p(t)) + \mathbf{f}_t(\mathbf{q}_h(t - dt), \mathbf{q}_p(t)) = 0. \quad (22)$$

By substituting the constraint projection (3), the position of the proxy can be solved analytically.

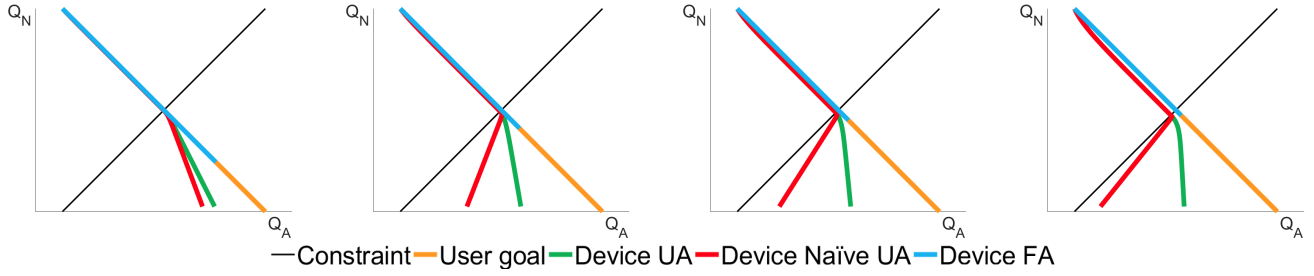


Fig. 4. Comparison of rendering strategies for different values of the impedance ratio k_r/k_u , on a synthetic experiment with a 2-DoF device with one actuated DoF and one non-actuated DoF. We compare our underactuated rendering solution (green, UA), fully actuated rendering (blue, FA), and naïve underactuated rendering (red, naïve UA). From left to right, the ratio is 1, 5, 10 and 20. With a high impedance ratio, naïve rendering suffers devastating ghost forces. With a low impedance ratio, it seems to produce a result similar to our solution just in this particular case, but it fails to render stiff constraints. Our solution, on the other hand, eliminates ghost forces and enables a high impedance ratio and therefore high transparency.

Putting it all together, the synthetic models of the device (21) and the proxy (22) are updated on every simulation step as follows:

$$(m_h \mathbf{I} + dt b_h \mathbf{I} + dt^2 \mathbf{K}_u) \dot{\mathbf{q}}_h(t) = m_h \dot{\mathbf{q}}_h(t - dt) + \quad (23)$$

$$dt (\mathbf{K}_u \mathbf{q}_u(t) + \mathbf{K}_r \mathbf{q}_p(t - dt) - (\mathbf{K}_u + \mathbf{K}_r) \mathbf{q}_h(t - dt)), \quad (24)$$

$$\mathbf{q}_h(t) = \mathbf{q}_h(t - dt) + dt \dot{\mathbf{q}}_h(t), \quad (25)$$

We carry out various synthetic experiments, by varying the trajectory of the user goal $\mathbf{q}_u(t)$ and the rendering settings. In all the experiments, we consider a slanted virtual wall (at 45 degrees). This wall orientation is representative of all possible cases, as it affects both the actuated and non-actuated DoFs. The next two subsections discuss the results of the synthetic experiments. The accompanying video also shows animations of these results.

6.1.1 Evaluation under Varying User Trajectories

Fig. 3 compares our underactuated rendering solution, fully actuated rendering, and naïve underactuated rendering for different directions of the user's intent: aligned with the non-actuated DoF, normal to the wall, and aligned with the actuated DoF. Notice how, in all three cases, our solution achieves optimal underactuated rendering, as the configuration of the device along actuated DoFs is the same as with fully actuated rendering. With naïve rendering, on the other hand, the trajectory of the device moves away from the user's intent, due to ghost forces.

Despite considering device dynamics and the sampled-data nature of the rendering problem, we demonstrate that our solution based on anisotropic tracking impedance still provides an optimal result. Actually, the assumption of static equilibrium becomes valid once the user intent becomes fixed, hence the theoretical result holds exactly. Possible, but minor, differences exist during dynamic motion.

6.1.2 Evaluation under Varying Rendering Impedance

According to expression (15), the effect of ghost forces in naïve underactuated rendering becomes more severe with higher rendering impedance k_r . Fig. 4 compares the various rendering strategies with different impedance values. Specifically, we vary the impedance ratio k_r/k_u between 1 (left) and 20 (right). At low impedance ratios, the naïve strategy and our solution produce similar results, but the virtual wall appears excessively compliant. At high impedance ratios, the virtual wall appears stiff (on the actuated DoF) with our solution, while naïve underactuated rendering suffers devastating ghost forces.

6.2 Real-World Experiment

We have designed a 2D interaction example to confirm our theoretical findings about ghost forces with proxy-based rendering, and to validate our proposed solution. The example, shown in Fig. 5, uses a 3-DoF Novint Falcon haptic device, limited to 2D motion. Furthermore, we disable forces on the vertical axis to mimic underactuation. We have placed a slanted virtual wall in the 2D scene, and we ask a user to move from the top-left corner toward the bottom-right corner, hence colliding normal to the virtual wall.

We invite the reader to watch the recorded interactions in the accompanying video, where the ghost forces of naïve rendering are evident, as well as the success of our proposed anisotropic tracking impedance. Fig. 5 shows snapshots of these interactions. The top rows shows the result of naïve rendering, with ghost forces pulling the device away from the user's intent. The bottom row shows the result with our solution, which suffers no ghost forces and matches the result of fully actuated rendering along the actuated horizontal axis. The device penetrates slightly the constraint in the vertical direction, but this is inevitable due to underactuation. The video also shows how, with our solution, the user can smoothly slide along the constraint.

In Fig. 6, we compare rendering performance on the same example for some representative interactions, under various rendering settings (fully actuated, naïve underactuated rendering, and our method). The plots of trajectories confirm the motions visible in the video, with ghost forces of naïve rendering moving the user away from their intent. The user is unable to explore the constraint naturally. With our method, the penetration depth is larger than in a fully actuated case, and this can be interpreted as a reduction in the perceived stiffness. But the interaction is natural.

7 CONCLUSIONS AND FUTURE WORK

We have characterized ghost forces in standard proxy-based rendering strategies when applied to underactuated haptic devices, and we have designed an optimal rendering strategy based on anisotropic tracking impedance, which eliminates the ghost forces of naïve methods. Even though our analysis and solution are based on a quasi-static assumption, our synthetic and real-world experiments show that our findings hold in practice in the presence of device dynamics and the sampled-data nature of the rendering algorithm.

One important contribution of our work is to consider the user's intent in the analysis. Currently, we model the user's intent through a goal position and a pulling impedance, but the model could be extended to consider richer characteristics of the user's intent. Moreover, in our examples, we consider the

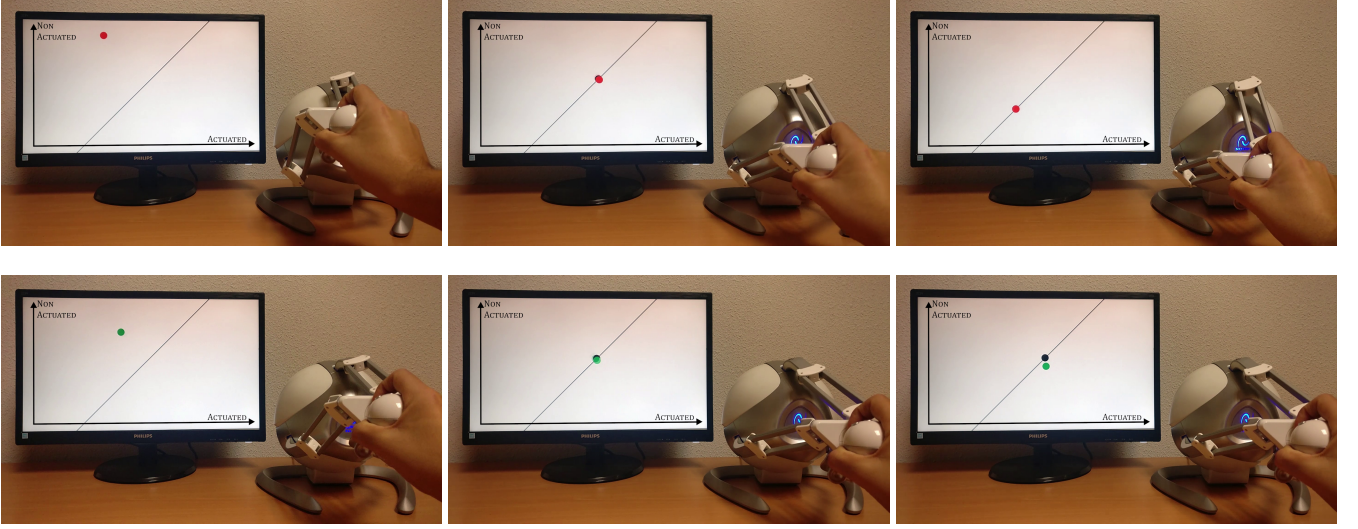


Fig. 5. The images show a 2D interaction example with a Novint Falcon haptic device, where the forces in the vertical axis are disabled to mimic underactuation. The top row shows interaction with naïve proxy-based rendering, and the bottom row interaction with our solution. In both cases, the user tries to move from the top-left corner of the scene toward the bottom-right corner, hence collides normal to the slanted virtual wall. With naïve rendering, ghost forces pull the device (red) to the left, against the user intent, as shown in the top-right image. With our solution, there are no ghost forces, and the motion of the device (green) along the actuated horizontal axis matches the motion in the fully actuated case. Please watch full recorded interactions in the accompanying video.

user's impedance as given, which is not a valid assumption in practical settings. To apply our rendering strategy in practice, it is necessary to estimate the user's impedance dynamically based on interaction data.

Our analysis and solution are based on several additional assumptions, and those assumptions should be relaxed to extend our findings to more general settings. We have considered a device configuration space where all DoFs are linear and have common geometric and mechanical characteristics, but this is rarely the case for actual devices. If each DoF has different properties, the associated impedance may also be different; i.e., the maximum stable rendering impedance will be different for each DoF. Anisotropy of the device DoFs should be accounted for in the theoretical analysis. Furthermore, practical settings decouple the operational and configuration spaces. Constraints are defined in operational space, while underactuation is expressed in configuration space. To apply our findings in practical settings, it is also necessary to map the various rendering elements to and from the operational and configuration spaces, and the mapping is typically nonlinear and not invariant. Addressing these limitations would be necessary, for example, to extend our findings to typical stylus devices with 6-DoF input (translation and rotation) and 3-DoF output (force but no torque). Other devices, e.g., the SPIDAR-S [22], introduce other challenges such as a time-varying direction of the actuated DoFs.

As a general conclusion, our work opens the door to novel considerations in the design of rendering algorithms for underactuated devices, and our theoretical findings should be revisited in the context of more complex settings. Moreover, it would be interesting to revisit previous works that have analyzed underactuated rendering from the point of view of passivity, task performance, or user experience [16], [17], [18].

APPENDIX A DERIVATION OF THE TRACKING STIFFNESS RATIO

Given underactuation with rendering impedance defined as in (12) and anisotropic tracking stiffness defined as in (16), the

equilibrium equations (6) and (7) for the proxy and device can be written as:

$$k_r \mathbf{A} \mathbf{A}^T (\mathbf{q}_p - \mathbf{q}_h) + k_u (\mathbf{q}_u - \mathbf{q}_h) = 0, \quad (26)$$

$$k_c (\mathbf{q}_c - \mathbf{q}_p) + \left(k_{ta} \mathbf{A} \mathbf{A}^T + k_{tn} \mathbf{N} \mathbf{N}^T \right) (\mathbf{q}_h - \mathbf{q}_p) = 0. \quad (27)$$

We project the equations to the actuated and non-actuated subspaces,

$$k_r \mathbf{A}^T (\mathbf{q}_p - \mathbf{q}_h) + k_u \mathbf{A}^T (\mathbf{q}_u - \mathbf{q}_h) = 0, \quad (28)$$

$$k_u \mathbf{N}^T (\mathbf{q}_u - \mathbf{q}_h) = 0, \quad (29)$$

$$k_c \mathbf{A}^T (\mathbf{q}_c - \mathbf{q}_p) + k_{ta} \mathbf{A}^T (\mathbf{q}_h - \mathbf{q}_p) = 0, \quad (30)$$

$$k_c \mathbf{N}^T (\mathbf{q}_c - \mathbf{q}_p) + k_{tn} \mathbf{N}^T (\mathbf{q}_h - \mathbf{q}_p) = 0, \quad (31)$$

and we eliminate the unknown device position from the equations:

$$k_c \mathbf{A}^T (\mathbf{q}_c - \mathbf{q}_p) + \frac{k_{ta} k_u}{k_r + k_u} \mathbf{A}^T (\mathbf{q}_u - \mathbf{q}_p) = 0, \quad (32)$$

$$k_c \mathbf{N}^T (\mathbf{q}_c - \mathbf{q}_p) + k_{tn} \mathbf{N}^T (\mathbf{q}_u - \mathbf{q}_p) = 0. \quad (33)$$

We can apply the optimality condition for underactuated rendering, i.e., the user goal and the proxy are aligned along the constraint normal. Mathematically, we express this as $\mathbf{q}_u = \mathbf{q}_c - d \mathbf{n}$ and $\mathbf{q}_p = \mathbf{q}_c - \alpha \mathbf{n}$ for some unknown value α . Substituting these expressions in the equations above, we obtain:

$$\alpha k_c \mathbf{A}^T \mathbf{n} + (\alpha - d) \frac{k_{ta} k_u}{k_r + k_u} \mathbf{A}^T \mathbf{n} = 0, \quad (34)$$

$$\alpha k_c \mathbf{N}^T \mathbf{n} + (\alpha - d) k_{tn} \mathbf{N}^T \mathbf{n} = 0. \quad (35)$$

These equations must hold for arbitrary constraints \mathbf{n} , hence they reduce to:

$$\alpha k_c + (\alpha - d) \frac{k_{ta} k_u}{k_r + k_u} = 0, \quad (36)$$

$$\alpha k_c + (\alpha - d) k_{tn} = 0. \quad (37)$$

We subtract both equations to obtain:

$$(\alpha - d) \left(\frac{k_{ta} k_u}{k_r + k_u} - k_{tn} \right) = 0, \quad (38)$$

which must hold for arbitrary values of d and α , and as a result we obtain the optimal anisotropy ratio (17).

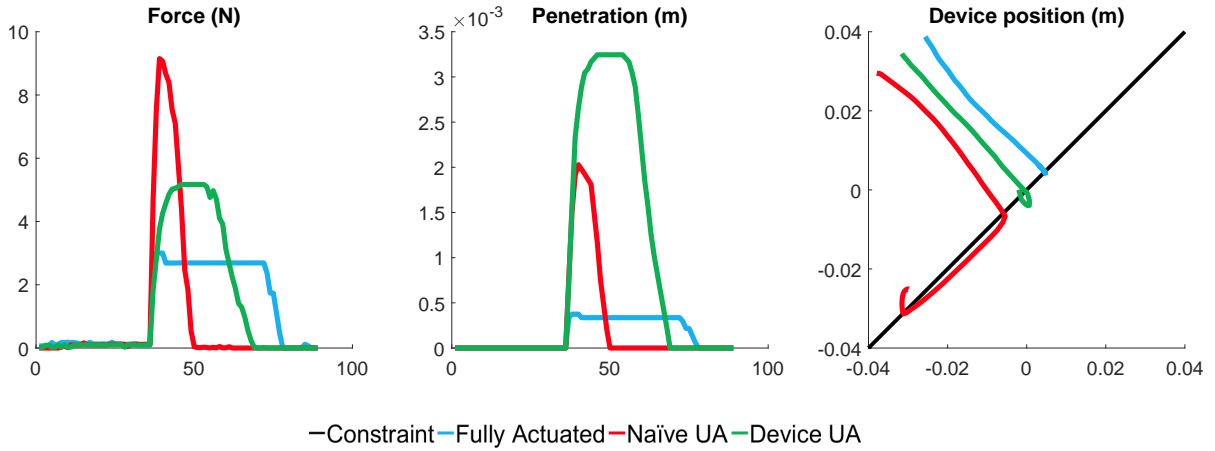


Fig. 6. Comparison of rendering performance on the real-world example shown in Fig. 5, for various rendering settings (fully actuated in blue, naïve underactuated rendering in red, and our method in green). The user moves from the top-left corner of the environment toward the bottom-right corner, and collides with a slanted constraint along the way. The plots on the right show the resulting user trajectories for a sample representative motion. With naïve rendering, ghost forces move the user away from their intent, toward the bottom-left corner. The user is unable to explore the constraint naturally. Please see also the accompanying video. With our method, the penetration depth is larger than in a fully actuated case, and this can be interpreted as a reduction in the perceived stiffness. But the interaction is natural.

ACKNOWLEDGMENTS

This work was funded in part by the European Research Council (ERC Consolidator Grant no. 772738 TouchDesign).

REFERENCES

- [1] T. H. Massie and J. K. Salisbury, "The phantom haptic interface: A device for probing virtual objects," in *Proceedings of the ASME Dynamic Systems and Control Division*, 1994, pp. 295–301.
- [2] T. Laliberte, L. Birglen, and C. Gosselin, "Underactuation in robotic grasping hands," *Machine Intelligence & Robotic Control*, vol. 4, no. 3, pp. 1–11, 2002.
- [3] P. Heo, G. M. Gu, S.-j. Lee, K. Rhee, and J. Kim, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 5, pp. 807–824, 2012.
- [4] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 3, 1995, pp. 146–151 vol.3.
- [5] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '97, 1997, pp. 345–352.
- [6] J. E. Colgate, M. C. Stanley, and J. M. Brown, "Issues in the haptic display of tool use," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 140–145, 1995.
- [7] M. Otaduy, C. Garre, and M. Lin, "Representations and algorithms for force-feedback display," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2068–2080, Sept 2013.
- [8] F. Barbagli and K. Salisbury, "The effect of sensor/actuator asymmetries in haptic interfaces," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003. *HAPTICS 2003. Proceedings.*, 2003, pp. 140–147.
- [9] L. N. Verner and A. M. Okamura, "Sensor/actuator asymmetries in telemanipulators: Implications of partial force feedback," in *2006 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2006, pp. 309–314.
- [10] —, "Force & torque feedback vs force only feedback," in *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2009, pp. 406–410.
- [11] R. Weller and G. Zachmann, "User performance in complex bi-manual haptic manipulation with 3 dofs vs. 6 dofs," in *2012 IEEE Haptics Symposium (HAPTICS)*, 2012, pp. 315–322.
- [12] A. Lecuyer, J. Burkhardt, J. Le Biller, and M. Congedo, "'A4': a technique to improve perception of contacts with under-actuated haptic devices in virtual reality," in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, 2005, pp. 316–322.
- [13] P. Kadleček, P. Kmoch, and J. Krivánek, "Haptic rendering for under-actuated 6/3-dof haptic devices," in *Haptics: Neuroscience, Devices, Modeling, and Applications: 9th International Conference, EuroHaptics 2014, Versailles, France, June 24–26, 2014, Proceedings, Part II*. Springer Berlin Heidelberg, 2014, pp. 63–71.
- [14] P. Kadleček, "Haptic Rendering for 6/3-DOF Haptic Devices," Master's thesis, Charles University in Prague, Czech Republic, 2013.
- [15] G. R. Luecke, "Haptic interactions using virtual manipulator coupling with applications to underactuated systems," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 730–740, 2011.
- [16] L. Meli and D. Prattichizzo, *Task-Oriented Approach to Simulate a Grasping Action Through Underactuated Haptic Devices*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 249–257.
- [17] D. Lobo, M. Sarac, M. Verschoor, M. Solazzi, A. Frisoli, and M. A. Otaduy, "Proxy-based haptic rendering for underactuated haptic devices," in *2017 IEEE World Haptics Conference (WHC)*, 2017, pp. 48–53.
- [18] M. Sarac, M. Solazzi, M. A. Otaduy, and A. Frisoli, "Rendering strategies for underactuated hand exoskeletons," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2087–2092, 2018.
- [19] M. Ortega, S. Redon, and S. Coquillart, "A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 458–469, 2007.
- [20] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degrees-of-freedom haptic rendering using voxel sampling," in *Proc. of SIGGRAPH 99*, ser. Computer Graphics Proc., Aug. 1999, pp. 401–408.
- [21] M. A. Otaduy and M. C. Lin, "A modular haptic rendering algorithm for stable and transparent 6-dof manipulation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 751–762, 2006.
- [22] S. Ma, M. Toshima, K. Honda, K. Akahane, and M. Sato, "Spidar-s: A haptic interface for mobile devices," in *Smart Graphics*, Y. Chen, M. Christie, and W. Tan, Eds. Springer International Publishing, 2017, pp. 203–206.