

Trajectory Deformations from Physical Human-Robot Interaction

Dylan P. Losey, *Student Member, IEEE*, and Marcia K. O'Malley, *Senior Member, IEEE*

Abstract—Robots are finding new applications where physical interaction with a human is necessary: manufacturing, health-care, and social tasks. Accordingly, the field of physical human-robot interaction (pHRI) has leveraged impedance control approaches, which support compliant interactions between human and robot. However, a limitation of traditional impedance control is that—despite provisions for the human to modify the robot’s current trajectory—the human cannot affect the robot’s future desired trajectory through pHRI. In this paper, we present an algorithm for physically interactive trajectory deformations which, when combined with impedance control, allows the human to modulate both the actual and desired trajectories of the robot. Unlike related works, our method explicitly deforms the future desired trajectory based on forces applied during pHRI, but does not require constant human guidance. We present our approach and verify that this method is compatible with traditional impedance control. Next, we use constrained optimization to derive the deformation shape. Finally, we describe an algorithm for real time implementation, and perform simulations to test the arbitration parameters. Experimental results demonstrate reduction in the human’s effort and improvement in the movement quality when compared to pHRI with impedance control alone.

Index Terms—physical human-robot interaction, shared control, haptics and haptic interfaces, learning from demonstration

I. INTRODUCTION

Physical human-robot interaction (pHRI) has become more pervasive as robots transition from structured factory floors to unpredictable human environments. Today we can find applications of pHRI not only within manufacturing, but also for rehabilitation, surgery, training, and comanipulation. In many of these situations, the human and robot are working collaboratively [1]; both agents share a common goal, mutually respond to each other’s actions, and provide assistance when needed [2]. Of course, while the human and robot may agree upon the goal they are trying to reach or the task they are attempting to perform, they might disagree on the trajectory that should be followed. Within this context, impedance control—as originally proposed by Hogan [3]—has traditionally been leveraged to relate interaction forces with deviations from the robot’s desired trajectory. Impedance control helps to provide compliant, safe, and natural robotic behavior [4], and is currently regarded as the most popular control paradigm for pHRI [5]. Unfortunately, while impedance control enables the human to modify the robot’s *actual* trajectory, it does not allow the human to interact with the robot’s *desired* trajectory.

This work was funded in part by the NSF GRFP-1450681.

The authors are with the Mechatronics and Haptic Interfaces Laboratory, Department of Mechanical Engineering, Rice University, Houston, TX 77005. (e-mail: dlosey@rice.edu)

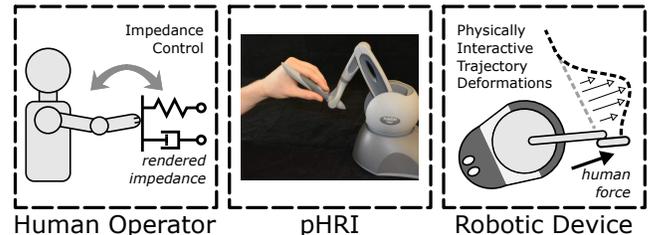


Fig. 1. Schematic of the proposed framework for pHRI. The human interacts with a robot, which renders a desired impedance (left). In response to the human’s applied force, the robot’s future desired trajectory is updated using physically interactive trajectory deformations (right).

Practically, this can cause humans to expend more effort when working to change the behavior of the robot, leading to higher effort, or “inefficient,” human-robot collaboration [6]–[8].

As a result, extensions of impedance control have been developed where the robot proactively moves along the human’s desired trajectory [7]–[9]. Under these techniques, the human dictates the desired trajectory and leads the interactions, while the robot estimates the human’s intent and acts as a transparent follower. Because the human guides the robot, however, the robot cannot meaningfully intervene towards reaching the goal or completing the task, and hence collaboration is restricted. Alternatively, shared control methods for comanipulation can be used, where the human and robot dynamically exchange leader and follower roles [10]–[13]. Both the human and robot are able to contribute to the robot’s motion, and the robot’s level of autonomy is adjusted by the shared control allocation. Although the robot can now meaningfully contribute within this shared control, we observe that the human is again unable to directly alter the robot’s desired trajectory through forces applied during pHRI. Hence, it may be beneficial to develop an approach which combines the advantages of both changing the desired trajectory and sharing control. Under such a scheme, haptic interactions could become a bidirectional information exchange; the human physically conveys task-relevant modifications to the robot, while the robot’s force feedback informs the human about the current desired trajectory.

In this paper, we propose physically interactive trajectory deformations, which—when implemented alongside impedance control—enables the human to simultaneously interact with the robot’s actual and desired trajectories. By applying input forces, the human operator not only experiences the rendered compliance of the impedance controller, but also continually modifies or deforms a *future* segment of the desired trajectory (see Fig. 1). Because the deformed desired trajectory returns to the original desired trajectory after some fixed time interval, the human does not need to constantly guide the

robot, and so the robot can also contribute towards completing the shared task. Our approach is intended for applications where the human wants to change *how* a robot behaves through physical interaction, and this robot is coupled to either a real or virtual environment. First, we derive constraints to ensure that the trajectory deformations are compatible with our impedance controller, and use constrained optimization to find a smooth, invariant deformation shape which resembles natural human motion. Next, an algorithm implementing our approach in real time is presented, along with simulations demonstrating how the algorithm’s parameters can be tuned to arbitrate between the human and robot. Finally, we evaluate impedance control with physically interactive trajectory deformations by conducting human-subject experiments on a haptic device. The results indicate that users exert less force and achieve better performance when interacting using the proposed method, as compared to a traditional impedance controller.

II. RELATED WORK

The problem we are considering and the solution we will pursue build on prior research from a variety of different fields. In particular, for applications that involve pHRI, we will discuss recent work on tracking the human’s desired trajectory and sharing control. Outside of pHRI applications, we also overview trajectory changes during human-robot collaboration, and examine algorithms which can be used to deform the robot’s desired trajectory in a smooth, human-like manner.

As previously mentioned, the robot can actively follow the human’s desired trajectory during pHRI; for example, [7] uses a Kalman filter to track the human’s desired timing of a point-to-point cooperative motion. Erden and Tomiyama [9] measure the controller force as a means to detect the human’s intent and update the robot’s desired position—when the human stops interacting with the robot, the robot maintains its most recent position. Similarly, Li and Ge [8] employ neural networks to learn the mapping from measured inputs to the human’s desired position, which the robot then tracks using an impedance controller. Although the human is able to change the robot’s desired behavior, these methods require the human to guide the robot along their intended trajectory.

In contrast with [7]–[9], shared control for comanipulation instead allows both the human and robot to dynamically exchange leader and follower roles [10]–[13]. In work by Li *et al.* [11], game theory is used to adaptively determine the robot’s role, such that the robot gradually becomes a leader when the human does not exert significant interaction forces. Kucukyilmaz *et al.* [12] have a similar criteria for role exchange, but find that performance decreases when visual and vibrotactile feedback informs the human about the robot’s current role. Medina *et al.* [13] utilize stochastic data showing how humans have previously completed the task; at times where the robot’s prediction does not match the human’s behavior, prediction uncertainty and risk-sensitive optimal control decide how much assistance the robot should provide. We note that shared control methods such as [1], [11], and [13] leverage optimal control theory in order to modulate the controller feedback gains, but—unlike our proposed approach—they track a fixed desired trajectory.

For shared control situations where the robot is continually in contact with an unpredictable environment—such as during a human-robot sawing task—Peternel *et al.* [14] propose multi-modal communication interfaces, including force, myoelectric, and visual sensors. By contrast, we consider tasks where the robot is attempting to avoid obstacles, and we focus on using pHRI forces without additional feedback. Besides comanipulation, shared control has also been applied to teleoperation, where the human interacts with a haptic device, and that device commands the motions of an external robot. In work by Masone *et al.* [15], [16], the authors leverage haptic devices to tune the desired trajectory parameters of a quadrotor in real time. These proposed adjustments are then autonomously corrected by the system to ensure path feasibility, regularity, and collision avoidance; afterwards, the haptic devices offer feedback about the resulting trajectory deformation.

Interestingly, even in settings where pHRI does not occur, other works have used the human’s actions to cause changes in the robot’s desired trajectory. Mainprice and Berenson [17] present one such scheme, where the robot explicitly tries to avoid collisions with the human. Based on a prediction of the human’s workspace occupancy, the robot selects the desired trajectory which minimizes human-robot interference and task completion time. Indeed, as pointed out by Chao and Thomaz [18], if the human and robot are working together in close proximity—but wish to avoid physical contact—the workspace becomes a shared resource. To support these methods, human-subject studies have experimentally found that deforming the desired trajectory in response to human actions objectively and subjectively improves human-robot collaboration [19]. However, it is not necessarily clear which trajectory deformation is optimal; as a result, there is interest in understanding how humans modify their own trajectories during similar situations. Pham and Nakamura [20] develop a trajectory deformation algorithm which preserves the original trajectory’s affine invariant features, with applications in transferring recorded human motions to humanoid robots.

Finally, from a motion planning perspective, optimization methods can be used to find human-like and collision-free desired trajectories by iteratively deforming the initial desired trajectory. For example, in work on redundant manipulators by Brock and Khatib [21], an initial desired trajectory from start to goal is given, and then potential fields are used to deform this trajectory in response to moving obstacles. More recently, Zucker *et al.* developed CHOMP [22], an optimization approach which uses covariant gradient descent to find the minimum cost desired trajectory; each step down the gradient deforms the previous desired trajectory. STOMP, from [23], generates a set of noisy deformations around the current desired trajectory, and then combines the beneficial aspects of those deformations to update the desired trajectory. TrajOpt, from [24], uses sequentially convex optimization to deform the initial desired trajectory—the resulting deformation satisfies both equality and inequality constraints. We observe that the discussed trajectory optimization schemes, [21]–[25], are not intended for pHRI, but have been successfully utilized to share control during teleoperation [26].

III. NOMENCLATURE

We here introduce some of the variables that will be used in this paper. For ease of notation, these variables are defined for a single degree-of-freedom (1-DoF) linear robot. Our problem is first considered in an abstract manner—Section IV—before we pursue a practical solution which can be implemented in real time—Section V. Accordingly, in Section IV, we work with functions in order to be more accurate and complete. Conversely, within Section V, we approximate these functions using waypoint parameterizations. The nomenclature discusses both cases as applicable.

- x_d^* : $\mathbb{R}^+ \rightarrow \mathbb{R}$ Original desired trajectory, assumed to be a smooth (C^∞) function.
- x_d Desired trajectory, updated after each trajectory deformation. In Section IV, $x_d : \mathbb{R}^+ \rightarrow \mathbb{R}$ is a function. In Section V, x_d is a set of waypoints.
- $h, k \in \mathbb{N}$ Discrete time variables.
- $T \in \mathbb{R}^+$ Sample period for the computer interface.
- $\delta \in \mathbb{R}^+$ Sample period for the desired trajectory. Equivalently, the time between waypoints along x_d .
- $r \in \mathbb{Z}^+$ Ratio between δ and T .
- $\tau_i \in \mathbb{R}^+$ Time at which the current trajectory deformation starts. Also, in Section V, the time associated with the most recent waypoint along x_d .
- $\tau_f \in \mathbb{R}^+$ Time at which the current trajectory deformation ends. Also, in Section V, the time associated with some future waypoint along x_d .
- $\tau \in \mathbb{R}^+$ Length of time of the trajectory deformation. An integer multiple of δ .
- γ_d Segment of the desired trajectory between τ_i and τ_f . In Section IV, $\gamma_d : [\tau_i, \tau_f] \rightarrow \mathbb{R}$ is treated as a function. In Section V, $\gamma_d \in \mathbb{R}^N$ is a vector of waypoints.
- $\tilde{\gamma}_d$ Deformation of γ_d with the same dual representations.
- $\Gamma_d : \mathbb{R} \times [\tau_i, \tau_f] \rightarrow \mathbb{R}$ Smooth family of trajectories.
- $\Gamma_d^s : \mathbb{R} \times [\tau_i, \tau_f] \rightarrow \mathbb{R}$ Trajectory within the smooth family of trajectories, shorthand for $\Gamma_d(s, t)$ with $s = \text{constant}$. In Section IV, $\Gamma_d^0(t) = \gamma_d(t)$ and $\Gamma_d^1(t) = \tilde{\gamma}_d(t)$.
- V Variation of γ_d used to find $\tilde{\gamma}_d$. As before, in Section IV, $V : [\tau_i, \tau_f] \rightarrow \mathbb{R}$ is a vector field along γ_d , and in Section V, $V \in \mathbb{R}^N$ is a vector.
- $N \in \mathbb{Z}^+$ Number of waypoints along γ_d or $\tilde{\gamma}_d$.
- $R \in \mathbb{R}^{N \times N}$ Matrix determining an inner product on \mathbb{R}^N .
- $H \in \mathbb{R}^N$ Shape of the optimal variation.
- $\mu \in \mathbb{R}^+$ Admittance of the optimal variation.

IV. PROBLEM STATEMENT

In order to develop a framework for physically interactive trajectory deformations, we will specifically consider 1-DoF linear robots. Restricting ourselves to 1-DoF keeps the notation simple, and enables us to pose the problem in a straightforward, instructive manner. Later, in Section V-E, we will show that the algorithm we ultimately derive can be independently applied to each DoF; hence, we do not lose any generality by now focusing on a single DoF. The robot—which could be thought of as a haptic device—can accordingly be modeled as

$$m\ddot{x}(t) + b\dot{x}(t) = f_a(t) + f_h(t) \quad (1)$$

where m is a point mass and b is the viscous friction constant. The position of this robot is denoted by x , and the device is subject to two external forces: f_a , the force applied by the actuator, and f_h , the force applied by the human. We would like to control f_a so that the robot follows a desired trajectory, x_d , while rendering a virtual impedance, where this impedance consists of a desired stiffness, k_d , and a desired damping, b_d .

A computer interface connects the robotic device to the virtual impedance. Recalling that computers necessarily introduce analog-to-digital conversion [27], the virtual force at each sample time is given by

$$f_v(hT) = k_d \left(x_d(hT) - x(hT) \right) + b_d \left(\dot{x}_d(hT) - \dot{x}(hT) \right) \quad \forall h \in \mathbb{N} \quad (2)$$

Here h is a discrete time variable, which increases by one after each sample, and T is the sample period for the computer interface. Next, using a zero-order hold (ZOH) to return from digital-to-analog, the actuator force becomes

$$f_a(t) = f_v(hT) \quad \forall t \in [hT, (h+1)T) \quad (3)$$

Viewed together, (1), (2), and (3) describe a standard 1-DoF linear haptic device under impedance control, while also capturing the effects of discretization. For a more in-depth analysis of this haptic system, particularly in regards to passivity, we recommend [27]–[29].

Our objective is to enable the human to intuitively and consistently change the desired trajectory, x_d , by applying forces, f_h , to the robotic device. We argue that the human's input should not only affect the robot's current state through the impedance controller, but also interact with the robot's future behavior via trajectory deformations. In order to more formally explore this concept, let us discuss some notation. The desired trajectory is initialized as x_d^* , a smooth, C^∞ function provided by the operator, where, if pHRI never occurs, $x_d(t) = x_d^*(t)$. Each time the human physically interacts with the robot, however, the desired trajectory is modified or “deformed,” and x_d is updated to include this deformation. The overall timing of the desired trajectory x_d is assumed to be correct, and, while deformations can locally alter the speed of the robot, the total time required to complete the task will not be changed by these deformations. Although we will first focus on a single trajectory deformation, it should be understood that x_d is continually modified in response to f_h through an iterative process; we refer to this approach as physically interactive trajectory deformations.

For the current trajectory deformation, define τ_i to be the time at which x_d starts to change, define τ_f to be a future time at which the current trajectory deformation ends, and define $\tau = \tau_f - \tau_i$ to be the duration of trajectory deformation. Thus, if the human applies a force f_h at an arbitrary time τ_i , that human input can alter x_d over the time interval $t \in [\tau_i, \tau_f]$. We will define $\gamma_d : [\tau_i, \tau_f] \rightarrow \mathbb{R}$ as the restriction of x_d to the interval $t \in [\tau_i, \tau_f]$, so that γ_d refers only to the segment of the desired trajectory with which the human can currently interact. Note that $\gamma_d(t) = x_d(t)$ when $t \in [\tau_i, \tau_f]$, but γ_d is not defined outside this time interval. Finally, let $\tilde{\gamma}_d$ denote

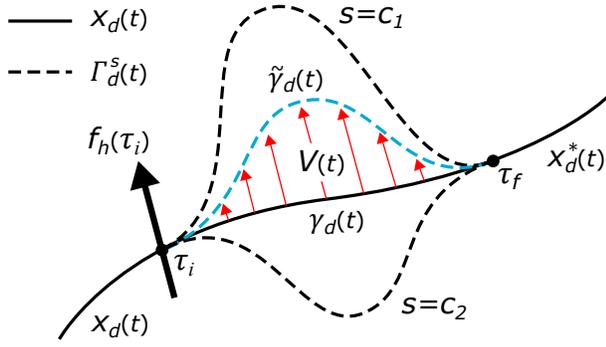


Fig. 2. Diagram of the current trajectory deformation. The segment of x_d between times τ_i and τ_f , labeled γ_d , is modified based on the human's force $f_h(\tau_i)$. The resultant deformation is $\tilde{\gamma}_d$ (in blue). The variation field V (in red) can be used to obtain $\tilde{\gamma}_d$ from γ_d . Other possible trajectory deformations, which are members of a smooth family of trajectories Γ_d , are also plotted (dashed lines). We use c_1 and c_2 to denote constants. Note that γ_d is Γ_d^0 , i.e., $s = 0$, and $\tilde{\gamma}_d$ is Γ_d^1 , i.e., $s = 1$. After $\tilde{\gamma}_d$ is determined, x_d would next be updated such that $x_d(t) = \tilde{\gamma}_d(t)$ over the time interval $t \in [\tau_i, \tau_f]$.

the deformation of γ_d , which we will derive in Section V by considering the human's force f_h at sample time τ_i . Once $\tilde{\gamma}_d$ has been determined, x_d is updated so that $x_d(t) = \tilde{\gamma}_d(t)$ over the interval $t \in [\tau_i, \tau_f]$. After time τ_f the robot again follows its original desired trajectory, x_d^* , since the human has not yet interacted with this future portion of the trajectory—although he or she can in subsequent trajectory deformations. To better visualize the setting, refer to Fig. 2.

Over the remainder of this section we identify the constraints on our choice of $\tilde{\gamma}_d$, and ensure that this trajectory deformation is compatible with our impedance controller. We use the notation and definitions presented by Lee [30]. Let $\Gamma_d(s, t)$ be a smooth family of trajectories—more commonly referred to as a smooth “family of curves”—where $\Gamma_d : \mathbb{R} \times [\tau_i, \tau_f] \rightarrow \mathbb{R}$. Intuitively, Γ_d contains two collections of curves; changing t when s is constant allows us to move along a single trajectory, while changing s when t is constant allows us to move between multiple trajectories, evaluating each of these trajectories at time t . For our purposes, the curves for which s is constant are especially important, because each value of $s = \text{constant}$ will yield a different smooth trajectory over the time interval $t \in [\tau_i, \tau_f]$. We write Γ_d^s to denote these smooth trajectories with constant s , and define

$$\Gamma_d^0(t) = \Gamma_d(0, t) = \gamma_d(t) \quad \forall t \in [\tau_i, \tau_f] \quad (4)$$

Hence, we have introduced Γ_d^s , where Γ_d^0 is γ_d , and all other values of s give deformations of γ_d . We observe that our use of Γ_d^s is similar to the sets of homotopic paths employed for trajectory deformations in [21]. Next, we utilize Γ_d^s with the relationship from (2) to obtain

$$F_v(\Gamma_d^s, hT) = k_d(\Gamma_d^s(hT) - x(hT)) + b_d(\partial_t \Gamma_d^s(hT) - \dot{x}(hT)) \quad \forall h \in [\tau_i/T, \tau_f/T] \quad (5)$$

Here F_v provides the virtual force at sample time hT when the desired trajectory is Γ_d^s , and ∂_t denotes a partial derivative with respect to time. As in (3), the actuator force F_a corresponding to Γ_d^s is simply equal to the virtual force from (5) evaluated

at the most recent sample time

$$F_a(\Gamma_d^s, t) = F_v(\Gamma_d^s, hT) \quad \forall t \in [hT, (h+1)T] \quad (6)$$

By construction, when $s = 0$, we know that $F_a(\Gamma_d^s, t) = f_a(t)$ over the time interval $t \in [\tau_i, \tau_f]$. As compared to (2) and (3), our equations (5) and (6) provide more generalized expressions for the virtual and actuator forces over the space of possible trajectory deformations.

We now constrain the actuator force to be continuous when the trajectory deformation starts, τ_i , and when the trajectory deformation ends, τ_f . This constraint guarantees that the human and robot will not experience any discontinuities in force when changing the desired trajectory, and accordingly prevents the trajectory modifications from interfering with our impedance controller. Indeed, continuous interaction forces have been observed in human-human dyads [31], and [32] created an accurate model for human-robot interaction forces by assuming that these forces were continuous. Leveraging the notation we have developed, we therefore assert that—regardless of our choice of s —the actuator force resulting from the deformed trajectory, F_a in (6), and the actuator force resulting from the original trajectory, f_a in (3), are equivalent at times τ_i and τ_f

$$F_a(\Gamma_d^s, \tau_i) = f_a(\tau_i) \quad \forall s \quad (7)$$

$$F_a(\Gamma_d^s, \tau_f) = f_a(\tau_f) \quad \forall s \quad (8)$$

Since (7) and (8) are satisfied when $s = 0$, and $F_a = F_v$ at sample times τ_i and τ_f , we could equivalently state

$$\partial_s F_v(\Gamma_d^0, \tau_i) = \partial_s F_v(\Gamma_d^0, \tau_f) = 0 \quad (9)$$

Again, ∂_s denotes a partial derivative with respect to s . For clarity, we point out that functions involving $\Gamma_d^s(t) = \Gamma_d(s, t)$ are really functions of t and s ; hence statements like $\partial_s \Gamma_d^0(t)$ imply that we first take the partial derivative of $\Gamma_d(s, t)$ with respect to s , and then evaluate this result for $s = 0$.

We have arrived at constraints on F_v which ensure that the human will experience continuous force feedback from the impedance controller when deforming the desired trajectory. Perhaps more usefully, by applying (9) to the right-side of (5), we can re-express these constraints in terms of Γ_d^s , and identify constraints on the shape of the trajectory deformation. Let Γ_d^s be linear in s

$$\Gamma_d^s(t) = \gamma_d(t) + sV(t) \quad \forall t \in [\tau_i, \tau_f] \quad (10)$$

Define $\Gamma_d^1(t) = \tilde{\gamma}_d(t)$, i.e., the curve $\Gamma_d(s, t)$ for which $s = 1$ is our trajectory deformation. We now use (10) to write

$$\tilde{\gamma}_d(t) = \gamma_d(t) + V(t) \quad \forall t \in [\tau_i, \tau_f] \quad (11)$$

where V , the variation, is a vector field along γ_d such that

$$V(t) = \partial_s \Gamma_d^0(t) \quad \forall t \in [\tau_i, \tau_f] \quad (12)$$

Furthermore, from [30] and (12), the following expressions are equivalent to the derivative of V with respect to t

$$\partial_s(\partial_t \Gamma_d^0(t)) = \partial_t(\partial_s \Gamma_d^0(t)) = \dot{V}(t) \quad (13)$$

Now, if we let V and \dot{V} equal zero at the start and end of the trajectory deformation,

$$V(\tau_i) = V(\tau_f) = 0 \quad (14)$$

$$\dot{V}(\tau_i) = \dot{V}(\tau_f) = 0 \quad (15)$$

we have that (9) is satisfied, and hence the actuator force of the robot, f_a , is continuous when transitioning between original and deformed trajectories.

More intuitively, recalling (11), our derived constraints state that $\tilde{\gamma}_d(t) = \gamma_d(t)$ and $\dot{\tilde{\gamma}}_d(t) = \dot{\gamma}_d(t)$ both at the start, τ_i , and the end, τ_f , of the trajectory deformation. These constraints aren't entirely unexpected. On the one hand, having fixed endpoint positions—our constraint (14)—has been a typical requirement for motion planning with trajectory optimization [22], [23], [25]. On the other hand, having fixed endpoint velocities—our constraint (15)—can be a requirement for other trajectory deformation algorithms [20], and even for modeling human-robot interaction forces [32]. As a result of our formulation, we have revealed that finding $\tilde{\gamma}_d$ boils down to determining V , the variation field. Of particular interest, selecting a variation field fulfilling (14) and (15) ensures that the resultant trajectory deformation is compatible with our impedance controller. Clearly, there are many choices of V that satisfy these constraints; for demonstration, one such choice is shown in Fig. 2. Because we would like to find the “best” variation field in accordance with (14) and (15), however, we turn our attention to optimization theory.

V. OPTIMAL TRAJECTORY DEFORMATIONS

Using the notation introduced in Section III and the trajectory deformations described by Section IV, we now present our method for physically interactive trajectory deformations. At the conclusion of Section IV, we argued that the optimal variation satisfying our endpoint constraints should be selected. Hence, we will first describe the energy of the trajectory deformation, which is conveniently expressed using a waypoint trajectory parameterization. Next, constrained optimization is leveraged to derive the variation V , and we explore how the shape of V can be tuned to arbitrate between the human and robot. We then present an algorithm that implements our results in real time, and combines impedance control with physically interactive trajectory deformations. Finally, we extend the approach to n -DoF robotic manipulators, and consider potential implementation issues. A block diagram of the scheme we are working towards is shown in Fig. 3. The left panel of Fig. 3 depicts equations (1), (2), and (3) from Section IV, while the steps and expressions in the right panel will be derived and explained below.

A. Energy of the Trajectory Deformation

In order to choose the optimal trajectory deformation, we first need to develop an energy function that maps trajectory deformations to their relative costs [33]. Following the example of [22]–[25], we will formulate our energy function using a waypoint trajectory parameterization. Let x_d now be discretized into an arbitrary number of waypoints equally

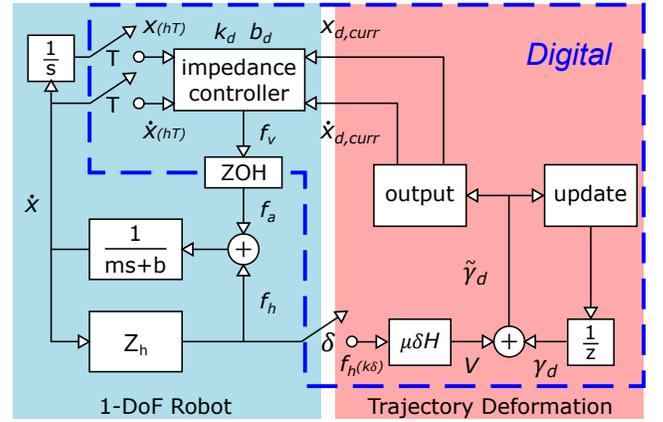


Fig. 3. Block diagram of a 1-DoF linear robot (on left, blue) with physically interactive trajectory deformations (on right, red). The digital portion (inside dashed line) is executed on a computer. Z_h is the human’s impedance, ZOH stands for zero-order hold, and the device is modeled as a point mass (m) and viscous damper (b). Recall that $1/z$ is a unit delay, and notice that sampling is occurring at two rates, with periods T and δ .

spaced in time, such that δ is the time interval between consecutive waypoints. In particular, because the time duration of the segment of x_d with which the human can interact is τ , the number of waypoints along γ_d and $\tilde{\gamma}_d$ is given by

$$N = \frac{\tau}{\delta} + 1 \quad (16)$$

We can therefore represent γ_d and $\tilde{\gamma}_d$ as vectors of length N , where each element of these vectors is a waypoint. For instance, referring back to the block diagram in Fig. 3, at $\tau_i = 0$ we initialize γ_d to be

$$\gamma_d = [x_d^*(0), x_d^*(\delta), x_d^*(2\delta), \dots, x_d^*(\tau)]^T \quad (17)$$

Throughout the rest of Section V, when we refer to γ_d , $\tilde{\gamma}_d$, or V , we will mean a vector in \mathbb{R}^N , and we will use the subscript $\gamma_{d,j}$ to refer to the j -th element of vector γ_d . Applying this waypoint parameterization, the energy of the trajectory deformation is defined as

$$E(\tilde{\gamma}_d) = E(\gamma_d) + (\tilde{\gamma}_d - \gamma_d)^T (-\hat{F}_h) + \frac{1}{2\alpha} (\tilde{\gamma}_d - \gamma_d)^T R (\tilde{\gamma}_d - \gamma_d) \quad (18)$$

Examining (18), the trajectory deformation’s energy is a summation of (a) the undeformed trajectory’s energy, (b) the work done by the trajectory deformation to the human, and (c) the squared norm of V with respect to the matrix R . We will separately discuss (b) and (c).

The desired trajectory should continually change in response to the human’s applied force, f_h . For this reason, we have included the second term in (18), which will enable f_h to affect the energy of the trajectory deformation. Let $F_h \in \mathbb{R}^N$ be defined as the human’s force applied along the variation vector V , where we recall from (11) that $V = \tilde{\gamma}_d - \gamma_d$. Thus, $F_{h,1}$ is the force applied at V_1 , $F_{h,2}$ is the force applied at V_2 , and so on. While we know that $F_{h,1} = f_h(\tau_i)$, i.e., the current human force, we of course cannot know what $F_{h,2}$ through $F_{h,N}$ will be, since these forces occur at future times. So as to develop a prediction of F_h —denoted as \hat{F}_h —we refer to work by Chipalkatty *et al.* [34]. These authors considered a situation

where, given the current human input, the robot attempts to both accomplish a high-level task and minimize the difference between its future actions and the human's future inputs. After human experiments, the authors found that simpler prediction methods outperform more complex approaches, and are also preferred by users. Like in [34] and [32], we therefore assume that future applied forces will be the same as the current input

$$\hat{F}_h = \vec{1}f_h(\tau_i), \quad \hat{F}_h \in \mathbb{R}^N \quad (19)$$

Note that $\vec{1} \in \mathbb{R}^N$ is just a vector of all ones. Substituting (19) into (18), the second term is equivalent to the negative of the total waypoint displacement multiplied by the current force, i.e., the work done to the human. Interestingly, if we compare (18) to the energy function used for deriving the update rule within the CHOMP algorithm [22], we observe that $\nabla E(\gamma) = -\hat{F}_h$. This implies that \hat{F}_h is the direction of steepest descent, and, by deforming the desired trajectory in the direction of the human's applied force, we will move down the gradient of their energy function. Thus, rather than attempting to explicitly learn E in (18), we are instead allowing the human to define the direction of steepest descent through pHRI.

The third term in (18) ensures that the resulting trajectory deformation will seem natural to the human operator. De Santis *et al.* [4] explain that robotic trajectories for pHRI should resemble human movements. In particular, the trajectory of human reaching movements can be accurately described by a minimum-jerk model [35], and, more recently, the minimum-jerk model has been extended to account for human trajectory modifications [36]. Thus, we wish to assign lower energies to natural, minimum-jerk choices of $V = \tilde{\gamma}_d - \gamma_d$. To achieve this end, let A be a finite differencing matrix such that—when ignoring boundary conditions [22]—we have $\ddot{V} = \delta^{-3} \cdot AV$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ 3 & -3 & 1 & \dots & 0 \\ -1 & 3 & -3 & 0 \\ 0 & -1 & 3 & 0 \\ 0 & 0 & -1 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & \dots & 3 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad A \in \mathbb{R}^{(N+3) \times N} \quad (20)$$

Next, we define R , a positive definite and symmetric matrix formed using (20)

$$R = A^T A, \quad R \in \mathbb{R}^{N \times N} \quad (21)$$

Both A and R are unitless quantities. By construction,

$$\|\ddot{V}\|^2 = \ddot{V}^T \ddot{V} = \delta^{-6} \cdot V^T R V = \delta^{-6} \cdot \|V\|_R^2 \quad (22)$$

and so R in (21) determines an inner product on \mathbb{R}^N . Unwinding these definitions, we additionally find that $\|V\|_R^2$ is proportional to the sum of squared jerks along the variation. As a result, the third term in (18) associates lower energies with natural shapes of V , i.e., those which minimize jerk.

B. Constrained Optimization

Our definition of $E(\tilde{\gamma}_d)$ trades-off between the work done to the human and the variation's total jerk. In order to find V , we would like to optimize $E(\tilde{\gamma}_d)$ subject to the constraints derived in Section IV; however, (14) and (15) must first be re-written using our waypoint parameterization. Let us introduce a matrix B , where

$$B = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}, \quad B \in \mathbb{R}^{4 \times N} \quad (23)$$

Using (23), we concisely express (14) and (15) as

$$B(\tilde{\gamma}_d - \gamma_d) = 0 \quad (24)$$

Intuitively, the four constraints in (24) guarantee that the first two and last two waypoints of $\tilde{\gamma}_d$ are the same as γ_d , which, in our prior terminology, implies that $V(t) = \dot{V}(t) = 0$ at the endpoints of the trajectory deformation, τ_i and τ_f . Combining (18) and (24), we now propose our optimization problem

$$\begin{aligned} & \text{minimize } E(\tilde{\gamma}_d) \\ & \text{subject to } B(\tilde{\gamma}_d - \gamma_d) = 0 \end{aligned} \quad (25)$$

Solving (25) both provides a “best” choice of V , and ensures that the resulting trajectory deformation does not interfere with the impedance controller. Furthermore, we can augment this optimization problem by adding terms to the energy equation [22], [23], [25], and/or providing additional constraints [22], [24]. Although we focus only on the simplest case, (25), we do wish to point out that aspects of other optimal motion planners could also be incorporated within our approach.

As demonstrated by [37], the optimization problem (25) can be solved through a straightforward application of the method of Lagrange multipliers. Let the Lagrangian be defined as

$$\mathcal{L}(\tilde{\gamma}_d, \lambda) = E(\tilde{\gamma}_d) + \lambda^T B(\tilde{\gamma}_d - \gamma_d) \quad (26)$$

where $\lambda \in \mathbb{R}^4$ is a vector of Lagrange multipliers. We use ∇_X to represent a gradient operator with respect to some vector X . Following the procedure in [33], the first-order necessity conditions for an extremum of \mathcal{L} are given by

$$\nabla_{\tilde{\gamma}_d} \mathcal{L}(\tilde{\gamma}_d, \lambda) = -\hat{F}_h + \frac{1}{\alpha} R(\tilde{\gamma}_d - \gamma_d) + B^T \lambda = 0 \quad (27)$$

$$\nabla_{\lambda} \mathcal{L}(\tilde{\gamma}_d, \lambda) = B(\tilde{\gamma}_d - \gamma_d) = 0 \quad (28)$$

After some algebraic manipulation of (27) and (28), and then substituting (19) in for \hat{F}_h , we obtain

$$\tilde{\gamma}_d = \gamma_d + \alpha G f_h(\tau_i) \quad (29)$$

We observe that the vector $G \in \mathbb{R}^N$ is formed using the identity matrix $I \in \mathbb{R}^{N \times N}$, R from (21), and B from (23). Accordingly, G is both unitless and constant, and can be completely defined once N from (16) is known

$$G = \left(I - R^{-1} B^T (B R^{-1} B^T)^{-1} B \right) R^{-1} \vec{1} \quad (30)$$

Although we have found an extremum, it is not yet clear whether $\tilde{\gamma}_d$ in (29) actually minimizes the energy function (18). To resolve this question, consider the Hessian of (18)

$$\nabla_{\tilde{\gamma}_d}^2 E(\tilde{\gamma}_d) = \frac{1}{\alpha} R \quad (31)$$

Because we have already established that R is positive definite, and we define the constant $\alpha > 0$, we conclude that the Hessian (31) is also positive definite. This implies that $\tilde{\gamma}_d$ in (29) minimizes $E(\tilde{\gamma}_d)$ subject to the equality constraints—solving the optimization problem (25), as desired.

C. Invariance and Arbitration

Comparing the result of our constrained optimization, (29), to the original relationship between $\tilde{\gamma}_d$ and γ_d , (11), we have revealed that $V = \alpha G f_h(\tau_i)$. We next discuss our selection of the constant α , where we have already constrained $\alpha > 0$. Because the variation V is a geometric object, the manner in which that object is represented should not alter its underlying shape [22]; hence, the shape of V should be invariant with respect to N , the number of waypoints. Indeed, referring back to Fig. 3, our method for physically interactive trajectory deformations should additionally be invariant with respect to $1/\delta$, the rate at which trajectory deformations occur. To achieve both types of invariance, we recall that G in (30) is a constant vector, and so we define

$$\alpha = \mu\delta \cdot \frac{\sqrt{N}}{\|G\|}, \quad H = \frac{\sqrt{N}}{\|G\|} G \quad (32)$$

where μ is a positive scalar. Then, substituting this choice of α into (29), we reach our final equation for $\tilde{\gamma}_d$

$$\tilde{\gamma}_d = \gamma_d + \mu\delta H f_h(\tau_i) \quad (33)$$

Due to the constraints found in Section IV, the energy function from Section V-A, the optimization problem within Section V-B, and the invariance just discussed, we have ultimately arrived

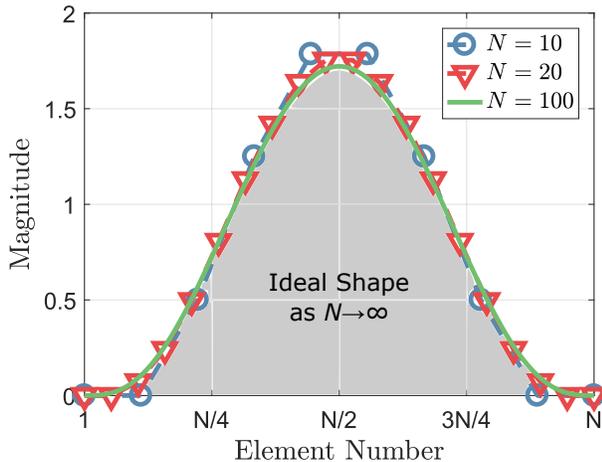


Fig. 4. Shape of the optimal variation (H) vs. the number of waypoints (N). The elements 1 through N of each vector $H \in \mathbb{R}^N$ are plotted, where we evaluated (32) for H and (30) for G . As N increases, H converges to the border of the shaded region. Increasing the number of waypoints leads to a better approximation of the ideal shape, but does not alter the ideal shape itself. Thus, the shape of V is invariant with respect to N . The first and last two elements of H have zero magnitude due to our endpoint constraints (24).

at $V = \mu\delta H f_h(\tau_i)$. Notice that the shape of V is determined by the constant, unitless vector $H \in \mathbb{R}^N$. Plots illustrating both the shape of our optimal variation and the invariance of H with respect to N can be seen in Fig. 4.

Our choice of μ in (33) arbitrates how V is impacted by f_h . The user-specified parameter $\mu \in \mathbb{R}^+$ has SI units $\text{m}/(\text{N}\cdot\text{s})$, and is therefore analogous to an admittance. Although there is some similarity between this parameter and the virtual springs in [21], or the “level of assistance” in [7], our use of μ is most similar to L in [9]. Erden and Tomiyama leverage L , also a positive scalar, to relate the integral of the controller force to a change in the desired trajectory [9]. Like this L , μ can be tuned to arbitrate between human and robot. When μ is large, our method for physically interactive trajectory deformations arbitrates towards the human—smaller input forces cause larger deformations. Conversely, as μ approaches zero, our method for physically interactive trajectory deformations arbitrates towards the robot—it becomes increasingly difficult for users to deform x_d .

D. Algorithm for Implementation

We have thus far focused on deriving an optimal variation for the current trajectory deformation. Of course, after $\tilde{\gamma}_d$ is found using (33), x_d is updated to include $\tilde{\gamma}_d$, and the process iterates at the next trajectory deformation (see Fig. 3). Here we present our algorithm to implement this iterative process in real time. The algorithm has two loop rates; the computer interface—and impedance controller—have a sample period T . Physically interactive trajectory deformations, on the other hand, has a sample period of δ , since δ is the time interval between waypoints along x_d . Let us define the ratio r as $r = \delta/T$. We assume that $r \in \mathbb{Z}^+$, since it is impractical to update the desired trajectory faster than the impedance controller can track x_d . Our algorithm simplifies if $r = 1$, but, with $r = 1$, the algorithm could become too computationally expensive as T decreases; this trade-off will be explored in Section VI. Finally, we observe that x_d is implicitly treated as a series of waypoints, where the desired position and velocity are held constant between those waypoints. Explicitly, we will maintain only the most recent waypoint, $x_{d,curr}$, and the subsequent waypoint, $x_{d,next}$, which correspond to sample times $k\delta$ and $(k+1)\delta$. From these two waypoints, discrete differentiation can be used to extract $\dot{x}_{d,curr}$. Considering the ZOH between waypoints, x_d as a function of time becomes

$$x_d(t) = x_{d,curr}(k\delta) \quad \forall t \in [k\delta, (k+1)\delta) \quad (34)$$

$$\dot{x}_d(t) = \dot{x}_{d,curr}(k\delta) \quad \forall t \in [k\delta, (k+1)\delta) \quad (35)$$

With these details in mind, we refer the reader to Algorithm 1. This algorithm can be seen as complementary to Fig. 3, where the algorithm implements the “digital” portion of that block diagram. To make Algorithm 1 more concise, we have initialized h and k as -1 , noting that, within the while loop, they are non-negative integers ($h, k \in \mathbb{N}$).

E. Extending to Multiple Degrees-of-Freedom

Now that physically interactive trajectory deformations has been established for a 1-DoF linear robot, we will show how

Algorithm 1 method for combining impedance control with physically interactive trajectory deformations

Given:

- Sampling periods: T, δ
- User-selected parameters: τ, μ
- Original desired trajectory: $x_d^*(t)$
- Desired stiffness and damping: k_d, b_d

Precompute:

- Vector H from (32), where $N = \tau/\delta + 1$ from (16)

Initialize:

- $\gamma_d \leftarrow [x_d^*(0), x_d^*(\delta), x_d^*(2\delta), \dots, x_d^*(\tau)]^T \triangleright (17)$
- $h \leftarrow -1, k \leftarrow -1$
- $x_{d,curr} \leftarrow \emptyset, \dot{x}_{d,curr} \leftarrow \emptyset$

while h is less than t_{stop}/T **do** $\triangleright t_{stop}$ is the stop time

$h \leftarrow h + 1$

if h is equal to $r(k + 1)$ **then** $\triangleright r = \frac{\delta}{T}$

$k \leftarrow k + 1$

$\tau_i \leftarrow k\delta \quad \triangleright \tau_f = \tau_i + \tau$

$f_h \leftarrow \text{sampleForce}(\tau_i)$

$\tilde{\gamma}_d \leftarrow \gamma_d + \mu\delta H f_h \quad \triangleright (33)$

output:

$x_{d,curr} \leftarrow \tilde{\gamma}_{d,1} \quad \triangleright \tilde{\gamma}_{d,1} = x_{d,curr}$

$\dot{x}_{d,curr} \leftarrow \frac{1}{\delta}(\tilde{\gamma}_{d,2} - \tilde{\gamma}_{d,1}) \quad \triangleright \tilde{\gamma}_{d,2} = x_{d,next}$

update:

$\gamma_d \leftarrow [\tilde{\gamma}_{d,2}, \tilde{\gamma}_{d,3}, \dots, \tilde{\gamma}_{d,N}, x_d^*(\tau_f + \delta)]^T$

end if

$x \leftarrow \text{samplePosition}(hT)$

$\dot{x} \leftarrow \text{sampleVelocity}(hT)$

$f_v \leftarrow k_d(x_{d,curr} - x) + b_d(\dot{x}_{d,curr} - \dot{x}) \quad \triangleright (2)$

end while

this method can be extended to n -DoF. In particular, we will consider a serial manipulator with forward kinematics $\mathbf{x} = \Phi(q)$, such that $q \in \mathbb{R}^n$ is the configuration in joint space, and $\mathbf{x} \in \mathbb{R}^m$ is the end-effector pose in task space ($m \leq 6$). Here **bold** denotes the generalization of previously defined scalars into m -length vectors, so, for instance, $\mathbf{f}_h \in \mathbb{R}^m$ becomes the wrench applied by the human at the robot's end-effector. From [38], the joint space dynamics of the robotic manipulator are

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u_r + J(q)^T \mathbf{f}_h \quad (36)$$

where $M \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C \in \mathbb{R}^{n \times n}$ contains the Coriolis and centrifugal terms, and $g \in \mathbb{R}^n$ is the gravity vector. The robot actuators apply joint torques $u_r \in \mathbb{R}^n$, and $J(q) \in \mathbb{R}^{m \times n}$ is the robot's Jacobian matrix, which defines the mapping $\dot{\mathbf{x}} = J(q)\dot{q}$, and is assumed to be full rank.

As before, we would like this robotic manipulator to track a desired end-effector trajectory, \mathbf{x}_d , while rendering a virtual stiffness and damping. Hence, the virtual wrench—the wrench that the robot ideally applies in task space—can be written

$$\mathbf{f}_v(hT) = K_d(\mathbf{x}_d(hT) - \mathbf{x}(hT)) + B_d(\dot{\mathbf{x}}_d(hT) - \dot{\mathbf{x}}(hT)) \quad \forall h \in \mathbb{N} \quad (37)$$

Within (37), $K_d \in \mathbb{R}^{m \times m}$ and $B_d \in \mathbb{R}^{m \times m}$ are the desired stiffness and damping matrices, respectively. Let us define

$K_d = \text{diag}(k_d^1, k_d^2, \dots, k_d^m)$ and $B_d = \text{diag}(b_d^1, b_d^2, \dots, b_d^m)$, where k_d^j and b_d^j are the desired stiffness and damping of the j -th task space coordinate. Re-examining (37) using these matrices, and introducing the notation $\mathbf{f}_v = [f_v^1, f_v^2, \dots, f_v^m]^T$, we see that the virtual force or torque f_v^j is found by implementing Algorithm 1 on the j -th coordinate in task space. In other words, because there are no cross-terms in K_d or B_d , \mathbf{f}_v can be obtained by performing Algorithm 1 separately with each of the m task space coordinates, and then combining the results into an m -length vector. We observe that H from (32), however, still only needs to be precomputed once.

The control law in (36) should be selected such that the human operator experiences \mathbf{f}_v , the virtual wrench, when interacting with our robotic manipulator. For simplicity of exposition, we might consider the following controller

$$u_r = J(q)^T \mathbf{f}_a + u_{ff} \quad (38)$$

Again applying a ZOH, \mathbf{f}_a at time t is equivalent to \mathbf{f}_v evaluated at the most recent sample time, hT . The feedforward torque vector, $u_{ff} \in \mathbb{R}^n$, could be used to either compensate for gravity or apply inverse dynamics [38]. Of course, many other choices for u_r are equally valid—e.g., see [8], [11], [13]—but our principal finding here is not the design of (38). Instead, like the CHOMP [22], STOMP [23], and ITOMP [25] algorithms, we have demonstrated that physically interactive trajectory deformations can straightforwardly scale to n -DoF manipulators. More specifically, we must maintain a total of $m \cdot N$ waypoints, where $1 \leq m \leq 6$, and so our approach scales linearly with the dimensionality of the task space.

F. Unintended Interactions and Trajectory Constraints

When implementing physically interactive trajectory deformations on single-DoF or multi-DoF robots, we must consider (a) unintended interactions and (b) trajectory constraints. Up until this point, it has been assumed that \mathbf{f}_h measures the external forces applied only by the human, and that all pHRI is intentional. In practice, however, the robot could collide with some unknown obstacle, or the human might accidentally interact with the robot. These unintended interactions alter \mathbf{f}_h , and, from Algorithm 1, change \mathbf{x}_d . Therefore, to mitigate the effects of unintended interactions, we recommend filtering the external force \mathbf{f}_h . For instance, [9] and [12] address similar issues by requiring the magnitude of \mathbf{f}_h to exceed a predefined threshold for a given length of time, and then adjust the desired trajectory or shared control allocation based on that filtered \mathbf{f}_h . During our human-subject experiments in Section VII, we will apply both a low-pass filter and minimum-force threshold before using \mathbf{f}_h for physically interactive trajectory deformations. As an aside, recall that these deformations only affect the future desired trajectory, and—since the stability of an impedance controller does not depend on the future desired trajectory [38]—we can conclude that our approach *will not influence* impedance controller stability.

After using Algorithm 1 to deform \mathbf{x}_d , the updated desired trajectory may not satisfy some trajectory constraints; for instance, \mathbf{x}_d could surpass the robot's joint limits, or collide with known obstacles. Like we previously indicated, this

TABLE I
BASELINE PARAMETERS USED IN SIMULATIONS

Parameter	T [s]	δ [s]	τ [s]	μ [m/N·s]
Value	10^{-3}	10^{-2}	1	1

problem can be addressed by incorporating additional costs or constraints within (25), as shown by [22]–[25]. A more straightforward solution, however, is simply to reject deformations which violate the trajectory constraints, and maintain the acceptable desired trajectory from the previous iteration. After computing (33) in Algorithm 1, we first check whether $\tilde{\gamma}_d$ satisfies some margin of safety with respect to the robot’s joint limits, free space, or other constraints. If $\tilde{\gamma}_d$ passes this check, proceed as usual; otherwise, reset $\tilde{\gamma}_d = \gamma_d$ before continuing. Readers should be aware that—because of the computational time associated with collision checking—this additional procedure may decrease the rate at which physically interactive trajectory deformations are performed.

VI. SIMULATIONS

To validate our method for physically interactive trajectory deformations and understand the effects of our user-specified parameters, we performed 1-DoF simulations in MATLAB (MathWorks). These simulations demonstrate how the desired trajectory, x_d , deforms in response to forces applied by the human, f_h . Unlike the hardware experiments in Section VII, here we will omit any robotic device. Simulations were conducted using the “Trajectory Deformation” portion of the block diagram in Fig. 3, or, equivalently, Algorithm 1, excluding the final lines involving x , \dot{x} , or f_v . In each simulation, we defined the human’s force input to be a pulse function

$$f_h(t) = \begin{cases} 1 & \text{if } 1 \leq t < 2 \\ 0 & \text{otherwise} \end{cases} \quad (39)$$

where f_h is measured in Newtons. Likewise, the original desired trajectory, x_d^* , was set as a sin wave

$$x_d^*(t) = -0.75 \cdot \sin(t) \quad (40)$$

Both x_d^* and x_d were measured in meters. So that readers can more easily reproduce our results, we have listed our simulation parameters in Table I. Note that k_d and b_d are not necessary, since the virtual force f_v is never computed. Given these parameters, f_h in (39), and x_d^* in (40), we ran Algorithm 1 with (34) to find x_d as a function of time. Parameters δ , τ , and μ were then separately varied in different figures, where the parameters not currently being tested maintained their values from Table I. Our results are shown in Figs. 5–8. Fig. 5 shows how x_d smoothly and progressively deforms over time as the user continues to apply force f_h . Fig. 6 illustrates that changing δ , the sampling period for physically interactive trajectory deformations, does not alter x_d . Figs. 7 and 8, on the other hand, display the different ways in which user-specified parameters τ and μ can affect x_d .

Viewing these figures together, it is clear that our method for physically interactive trajectory deformations enables the

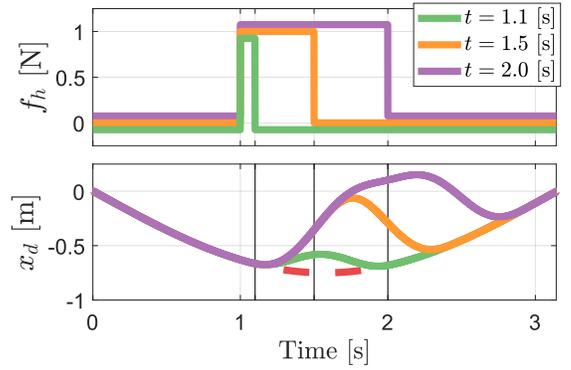


Fig. 5. Desired trajectory x_d corresponding to different durations of applied force f_h . The magnitude of f_h is the same in every plot, but we have added a visual offset for clarity. x_d^* from (40) is the dashed red line. The plot $t = 1.1$ s shows x_d after force is applied for 0.1 s, the plot $t = 1.5$ s shows x_d after force is applied for 0.5 s, and the plot $t = 2.0$ s shows x_d after force is applied for 1.0 s. As the user applies force, x_d iteratively deforms; these deformations aggregate smoothly over time.

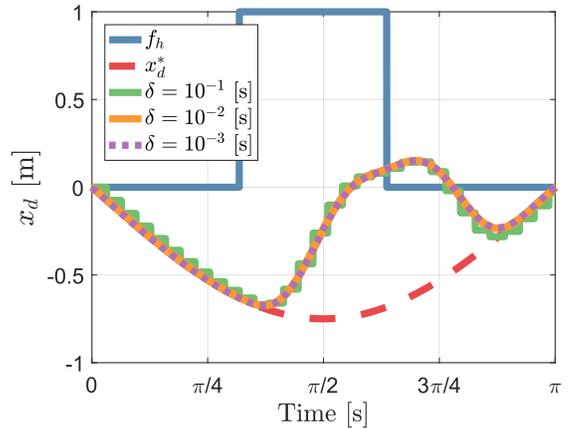


Fig. 6. Effect of the sample period δ on the desired trajectory x_d . Both f_h from (39), with units N, and x_d^* from (40), with units m, are plotted for reference. Regardless of the value chosen for δ , we follow the same underlying trajectory. Hence, our method is invariant with respect to δ . Note that decreasing δ leads to a better approximation of the underlying trajectory, but it also increases N , the number of waypoints. For example, when $\delta = 10^{-1}$ s we have $N = 11$, but when $\delta = 10^{-3}$ s, $N = 1001$.

human to actively vary the desired trajectory. When applying force $f_h = 1$ N, the simulated user induces x_d to iteratively deform, and, when no force is applied, x_d again follows x_d^* after τ seconds. Importantly, we observe that the transitions between x_d and x_d^* are smooth; moreover, the deformed portions of the desired trajectory are also smooth. Note that x_d deforms in the direction of f_h , maximizing the work done by the human’s applied force. Besides validating Algorithm 1, these simulations have additionally resolved the previously discussed trade-off between $r = 1$ and computational efficiency. In order to have $r = 1$, here we must choose $\delta = 10^{-3}$ s, which means we will have to constantly maintain $1000\tau + 1$ waypoints. As τ increases, this could eventually prevent real time implementation—however, even when $r = 1$ and $\tau = 5$ s, i.e., $N = 5001$, the simulations were still ten times faster than real time on a laptop computer. Finally, Figs. 7 and 8 provide some guidelines behind the selection of the user-defined parameters τ and μ . Increasing τ both causes the trajectory deformation to occur over a longer time interval, and

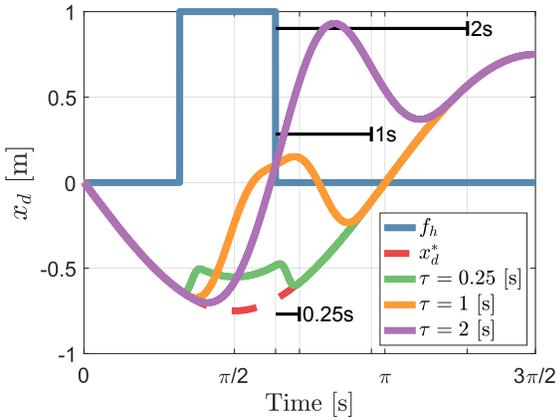


Fig. 7. Effect of the parameter τ on the desired trajectory x_d . Both f_h from (39), with units N, and x_d^* from (40), with units m, are plotted for reference. We observe that x_d smoothly returns to the original desired trajectory τ seconds after the human stops applying an input force (marked time intervals). Increasing τ not only allows the human deform x_d over a longer timescale, but it also increases the magnitude of the total deformation.

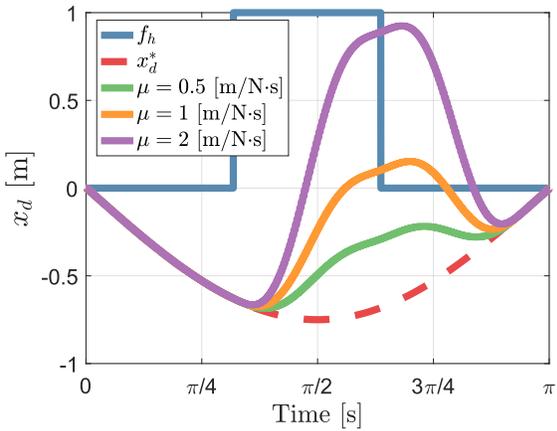


Fig. 8. Effect of the parameter μ on the desired trajectory x_d . Both f_h from (39), with units N, and x_d^* from (40), with units m, are plotted for reference. Increasing μ arbitrates towards the human, so that the given force input causes a larger change in x_d . The relationship between μ and the total deformation is not linear, although μ does linearly scale a single trajectory deformation.

increases the magnitude of the total deformation; therefore, to keep the total deformation magnitude constant, μ should be decreased as τ is increased. In general, the choice of τ and μ may depend on the given robot, environment, and human-robot arbitration. During implementation, we recommend setting τ and μ at values close to zero, and then gradually increasing these values in accordance with our guidelines until the desired performance is achieved.

VII. EXPERIMENTS

To compare impedance control (IC) and impedance control with physically interactive trajectory deformations (IC-PITD), we conducted human-subject experiments on a multi-DoF haptic device, as shown in Fig. 9. Here the human and robot collaborated to follow a desired trajectory, where the human must intervene to help the robot avoid “unknown” obstacles. The physical robot was coupled to a virtual environment—which visualized these obstacles—and the virtual robot represented the position of the actual haptic device. We performed repetitions of the same tracking and obstacle avoidance task

with IC and IC-PITD. We hypothesized that (a) humans would better complete the task with IC-PITD, both in terms of less torque applied and increased movement smoothness, and (b) differences in tracking error and obstacle collisions between IC and IC-PITD would be negligible.

A. Experimental Setup

Ten subjects (three females) participated in this experiment after signing a consent form approved by the Rice University Institutional Review Board. Subjects physically interacted with the flexion/extension (FE) and radial/ulnar (RU) joints of the OpenWrist, a robotic exoskeleton recently developed for wrist rehabilitation [39]. Thus, the human and robot shared control over 2-DoF. In order to measure the human’s input torque about both axes of rotation—FE and RU—we employed the nonlinear disturbance observer originally proposed in [40], and specifically applied to haptic devices by [41]. This disturbance observer, along with the IC and IC-PITD algorithms, were implemented upon a desktop PC through Matlab/Simulink (MathWorks) together with the QUARC blockset (Quansar). The computer communicated with the haptic device using a Q8-USB data acquisition device (Quansar), and the sample period T was set as $T = 10^{-3}$ s.

Subjects observed a computer monitor (see Fig. 9), where the virtual task was displayed. An image and explanation of this virtual task is provided by Fig. 10. The robot’s original desired trajectory, x_d^* , was defined as a circle in joint space

$$x_d^*(t) = \pi/9 \cdot [\cos(t), \sin(t)]^T \quad (41)$$

The robot’s position at time t , i.e., $x(t)$, was denoted by an orange torus. Subjects were instructed to move this torus to follow (41), while also interacting with the robot to avoid two obstacles. A single trial consisted of a complete revolution around x_d^* . It should be noted that—because of the IC and IC-PITD algorithms—this was a shared control task, where the robot could autonomously track (41) if no human inputs were present. The virtual stiffness, $K_d \in \mathbb{R}^2$, and virtual damping, $B_d \in \mathbb{R}^2$, were the same for both IC and IC-PITD. Based on our prior experience and preliminary experiments, we heuristically selected $K_d = \text{diag}(k_d, k_d)$, with $k_d = 35$ Nm/rad, and $B_d = \text{diag}(b_d, b_d)$, with $b_d = 0.5$ Nm·s/rad.

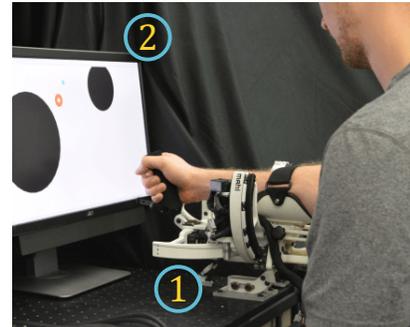


Fig. 9. Experimental setup. (1) participants grasped the handle of the Rice OpenWrist, a multi-DoF haptic device. (2) by interacting with this robot, participants could move their cursor within the virtual task.

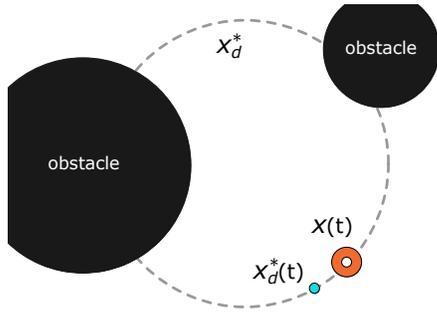


Fig. 10. Virtual task. By rotating the flexion/extension and radial/ulnar joints of the haptic device, subjects changed $x(t)$, their current position (orange torus). The original desired position at the current time, $x_d^*(t)$, was denoted by the blue sphere. Without any human intervention, $x(t)$ tracks $x_d^*(t)$ around a circular trajectory (dashed line), and intersects both obstacles. Participants were asked to interact with the robot such that the torus avoids each obstacle.

Recalling our findings from Section VI concerning computational efficiency, we additionally chose $\delta = 10^{-3}$ s, noting that, as a result of T and δ , $r = 1$ for this experiment. The IC-PITD parameters τ and μ were identified offline during preliminary experiments, where we iteratively increased τ and μ until we obtained interaction behavior that was noticeably different from impedance control. Using this heuristic process, we selected $\tau = 1.25$ s and $\mu = 0.35$ rad/Nm·s.

We utilized a within-subjects design, where all ten participants completed the same experimental protocol twice: once with IC, and once using IC-PITD. To better eliminate ordering effects, we counterbalanced which controller was tested first. Given a controller—either IC or IC-PITD—the experimenter initially demonstrated the virtual task for the subject. Once that subject understood the task, he or she then practiced for approximately five minutes by performing 50 unrecorded trials (revolutions). This training period was meant to familiarize the participant with the current controller, and also to help alleviate any learning effects or differences within initial skill [26]. After the training phase, subjects executed 15 more trials, the last 10 of which were recorded for our data analysis. Next, the same protocol—demonstrating, training, and recording trials—was repeated for the second controller. Accordingly, every participant performed a total of 20 recorded trials, i.e., 10 with both controller types (IC and IC-PITD).

B. Data Analysis

For each recorded trial we computed the (a) applied torque, (b) interaction percent, (c) collision percent, (d) tracking error, and (e) movement smoothness. *Applied torque* was defined as $\int_0^{2\pi} \|\mathbf{f}_h(t)\| dt$, where $\mathbf{f}_h(t) \in \mathbb{R}^2$ is the human’s torque about the FE and RU axes at time t , as estimated by the disturbance observer. *Interaction percent* indicates to the fraction of the trial during which the magnitude of the human’s applied torque exceeded 0.5 Nm; in other words, $\frac{1}{2\pi} \int_0^{2\pi} (\|\mathbf{f}_h(t)\| > 0.5) dt$. We use applied torque and interaction percent to assess not only the subject’s total effort, but also how persistently the human interacts with the robot. *Collision percent* refers to the fraction of the trial during which the human’s torus is in contact with an obstacle. *Tracking error* considers the difference between $x(t)$ and $x_d^*(t)$, but only when $x_d^*(t)$ is more than 0.2 radians away from both of the obstacles. Let

us refer to the segments of time where $x_d^*(t)$ is in this free space as $\tau_{free} \subseteq 2\pi$; then, the tracking error is calculated as $\int_{\tau_{free}} \|x(t) - x_d^*(t)\| dt$. We use both collision percent and tracking error to determine how accurately the participant completed our virtual task. Finally, *movement smoothness* was measured by leveraging the spectral arc-length metric [42]. Within this spectral-arc length algorithm, an amplitude threshold of 0.05 and a maximum cut-off frequency of 10 Hz were selected. We use movement smoothness, as measured by spectral-arc length, to investigate the human-like quality of the robot’s movements.

We separately computed each subject’s mean trial metrics across the 10 recorded trials with IC, as well as their mean trial metrics across the 10 recorded trials using IC-PITD. As such, for an individual subject, we obtained their average torque, interaction, collision, error, and smoothness metrics per trial when IC was present, along with the same averaged behavior per trial when IC-PITD was employed. So as to statistically test the effects of controller type, we then used paired t-tests with a significance level of $\alpha = 0.05$. Analysis concerning statistical significance was conducted using SPSS (IBM).

C. Results and Discussion

Our findings are summarized by Fig. 11, and reported below in the form *mean* \pm *std*. For trials with IC, subjects applied an average torque of 4.64 ± 0.43 Nm, while the same subjects applied an average torque of 2.25 ± 0.20 Nm during IC-PITD trials. Similarly, the interaction percent for the IC case, $57.90 \pm 8.29\%$, exceeded the interaction percent with IC-PITD, $20.77 \pm 4.97\%$. We concluded that there was a statistically significant decrease in both applied torque ($t(9) = 15.907$, $p < 0.001$) and interaction percent ($t(9) = 11.530$, $p < 0.001$) when subjects performed our pHRI task using IC-PITD. In terms of collision percent, we observed a slight increase between the IC case, $0.71 \pm 1.17\%$, and the IC-PITD trials, $1.41 \pm 1.32\%$, but this difference was not statistically significant ($t(9) = -1.203$, $p = 0.260$). Interestingly, a similar pattern was discovered for tracking error; in the IC trials, subjects had an average tracking error of 0.016 ± 0.005 rad, which was actually lower than the average tracking error of 0.027 ± 0.015 rad within the IC-PITD trials. This increase in error, however, was not determined to be statistically significant ($t(9) = -2.025$, $p = 0.073$). Finally, subjects had an average movement smoothness of -2.91 ± 0.24 for the IC case and -2.66 ± 0.11 during the IC-PITD trials. Recalling that—within the spectral arc-length metric—more negative values indicate less smooth movements [42], we found that IC-PITD lead to a statistically significant improvement in movement smoothness ($t(9) = -2.640$, $p = 0.027$).

Observing the low collision percent and tracking error, it is clear that subjects were able to successfully avoid the virtual obstacles and then return to following x_d^* while using either IC or IC-PITD. On the other hand, controller type did have a statistically significant effect on the subjects’ efficiency; when physically interactive trajectory deformations was combined with the impedance controller, participants could complete the task while applying less total torque, and did not need to

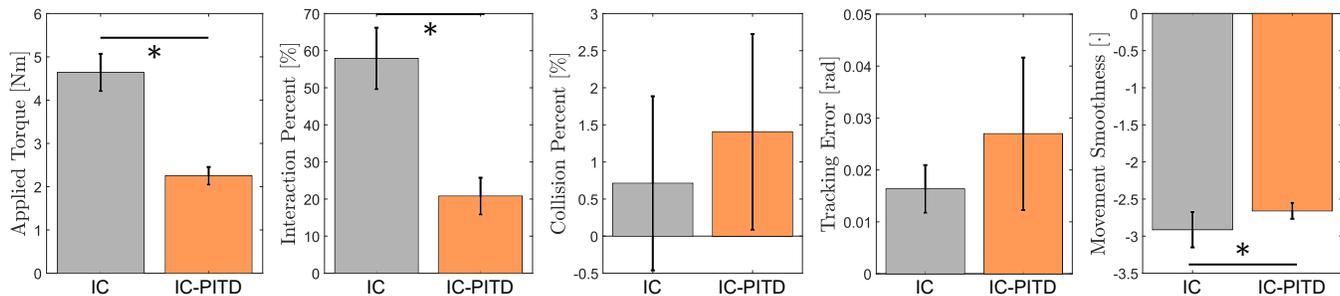


Fig. 11. Experimental results. Each plot shows the mean across subjects when impedance control was used (IC, gray), and when impedance control with physically interactive trajectory deformations was used (IC-PITD, orange). Error bars indicate the standard deviation, and an asterisk (*) denotes $p < 0.05$.

physically interact with the robot as frequently. Indeed, the robot’s resultant movements with IC-PITD were found to be smoother, and thus natural and human-like. The results from the described experiment therefore support our hypotheses: pHRI with IC-PITD was more efficient and smooth, without noticeably reducing accuracy as compared to IC. It must be recognized, however, that the same behaviors may not necessarily occur during other tasks, or for individual users with different levels of proficiency. As such, the results from our comparison of IC and IC-PITD should be understood as practical trends, and not absolute truths. Interestingly, based on the informal feedback we received from participants after the experiment, we recommend tuning the impedance controller to be more stiff when implementing IC-PITD. Subjects might have informally preferred higher K_d gains because IC-PITD provides two sources of compliance—both the impedance controller and the trajectory deformations—and so the stiffness of the impedance controller should be increased in compensation.

VIII. CONCLUSION

We proposed an impedance control algorithm which allows humans to simultaneously deform the robot’s actual and desired trajectories during pHRI. Under our approach, the human utilizes physical interactions to alter *how* the robot completes some collaborative task. Since the deformed desired trajectory smoothly returns to the original desired trajectory in the absence of human interactions, the robot maintains a level of autonomy, and both human and robot can meaningfully contribute towards collaborative movements. We first considered a single trajectory deformation, which was defined by selecting the variation; in particular, endpoint constraints on this variation were identified to prevent interference with impedance control. Next, the trajectory deformation’s energy was written as a balance of the work done to the human and the variation’s total jerk. After applying the method of Lagrange multipliers, we found the optimal variation shape which, in practice, could additionally be made invariant to the sampling period. We then introduced a real time algorithm that combined impedance control with physically interactive trajectory deformations, where the human now continuously deforms the robot’s desired trajectory by applying wrenches at the robot’s end-effector. This algorithm is intended for multi-DoF robots, and scales linearly with the dimensionality of the task space. To verify our algorithm, and intuitively understand the effects of its user-specified arbitration parameters, we performed 1-DoF simulations. Finally, we experimentally

demonstrated on human subjects and a 2-DoF haptic device that impedance control with physically interactive trajectory deformations yields efficient and smooth pHRI.

Future work includes comparing the deformation method presented in this paper to learning approaches that re-plan the robot’s desired trajectory. For instance, we are interested in explicitly learning the human’s energy function from pHRI—at each time step where our understanding of the human’s energy function is updated, we would then re-plan the robot’s desired trajectory to minimize that energy function. On the one hand, this learning and re-planning approach could lead to less pHRI and a desired trajectory that better matches the human’s preferences. On the other hand, our deformation approach may be more intuitive, and can be performed faster than re-planning the entire trajectory. Another topic for future work concerns the speed at which the task is completed; we recognize that a limitation of our current work is that the human cannot use pHRI to change the robot’s overall task timing. In future work, we would like to extend learning methods in order to infer the human’s desired timing through pHRI.

REFERENCES

- [1] N. Jarrassé, T. Charalambous, and E. Burdet, “A framework to describe, analyze and generate interactive motor behaviors,” *PLOS ONE*, vol. 7, no. 11, p. e49945, 2012.
- [2] M. E. Bratman, “Shared cooperative activity,” *Phil. Review*, vol. 101, no. 2, pp. 327–341, 1992.
- [3] N. Hogan, “Impedance control: An approach to manipulation,” *J. Dyn. Syst. Meas. Control*, vol. 107, no. 1, pp. 1–24, 1985.
- [4] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, “An atlas of physical human–robot interaction,” *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.
- [5] S. Haddadin and E. Croft, “Physical human-robot interaction,” in *Springer Handbook of Robotics*, 2nd ed., 2016, pp. 1835–1874.
- [6] N. Jarrassé, J. Paik, V. Pasqui, and G. Morel, “How can human motion prediction increase transparency?” in *Proc. IEEE Conf. Robot. Autom.*, 2008, pp. 2134–2139.
- [7] B. Corteville, E. Aertbeliën, H. Bruyninckx, J. De Schutter, and H. Van Brussel, “Human-inspired robot assistant for fast point-to-point movements,” in *Proc. IEEE Conf. Robot. Autom.*, 2007, pp. 3639–3644.
- [8] Y. Li and S. S. Ge, “Human–robot collaboration based on motion intention estimation,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.
- [9] M. S. Erden and T. Tomiyama, “Human-intent detection and physically interactive control of a robot without force sensors,” *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 370–382, 2010.
- [10] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, “The role of roles: Physical cooperation between humans and robots,” *Int. J. Robot. Res.*, vol. 31, no. 13, pp. 1656–1674, 2012.
- [11] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, “Continuous role adaptation for human–robot shared control,” *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 672–681, 2015.

- [12] A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan, "Intention recognition for dynamic role exchange in haptic collaboration," *IEEE Trans. Haptics*, vol. 6, no. 1, pp. 58–68, 2013.
- [13] J. R. Medina, T. Lorenz, and S. Hirche, "Synthesizing anticipatory haptic assistance considering human behavior uncertainty," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 180–190, 2015.
- [14] L. Peternel, N. Tsagarakis, and A. Ajoudani, "Towards multi-modal intention interfaces for human-robot co-manipulation," in *Proc. IEEE/RSJ Conf. Intelligent Robots Syst.*, 2016, pp. 2663–2669.
- [15] C. Masone, A. Franchi, H. H. Bühlhoff, and P. R. Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *Proc. IEEE/RSJ Conf. Intelligent Robots Syst.*, 2012, pp. 2641–2648.
- [16] C. Masone, P. R. Giordano, H. H. Bühlhoff, and A. Franchi, "Semi-autonomous trajectory generation for mobile robots with integral haptic shared control," in *Proc. IEEE Conf. Robot. Autom.*, 2014, pp. 6468–6475.
- [17] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. IEEE/RSJ Conf. Intelligent Robots Syst.*, 2013, pp. 299–306.
- [18] C. Chao and A. Thomaz, "Timed petri nets for fluent turn-taking over multimodal interaction resources in human-robot collaboration," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1330 – 1353, 2016.
- [19] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration," *Hum. Factors*, vol. 57, no. 1, pp. 21–33, 2015.
- [20] Q.-C. Pham and Y. Nakamura, "A new trajectory deformation algorithm based on affine transformations," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 1054–1063, 2015.
- [21] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.
- [22] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *Int. J. Robot. Res.*, vol. 32, no. 9–10, pp. 1164–1193, 2013.
- [23] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Conf. Robot. Autom.*, 2011, pp. 4569–4574.
- [24] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [25] C. Park, J. Pan, and D. Manocha, "ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments," in *Proc. Conf. Autom. Planning Scheduling*, 2012, pp. 207–215.
- [26] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 790–805, 2013.
- [27] N. Diolaiti, G. Niemeyer, F. Barbagli, and J. K. Salisbury, "Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 256–268, 2006.
- [28] J. J. Abbott and A. M. Okamura, "Effects of position quantization and sampling rate on virtual-wall passivity," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 952–964, 2005.
- [29] J. E. Colgate and J. M. Brown, "Factors affecting the Z-width of a haptic display," in *Proc. IEEE Conf. Robot. Autom.*, 1994, pp. 3205–3210.
- [30] J. M. Lee, *Riemannian manifolds: An introduction to curvature*. New York, NY, USA: Springer Science & Business Media, 2006.
- [31] K. B. Reed and M. A. Peshkin, "Physical collaboration of human-human and human-robot teams," *IEEE Trans. Haptics*, vol. 1, no. 2, pp. 108–120, 2008.
- [32] E. Noohi, M. Žefran, and J. L. Patton, "A model for human-human collaborative object manipulation and its application to human-robot interaction," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 880–896, 2016.
- [33] S. S. Rao, *Engineering optimization: Theory and practice*, 4th ed. New York, NY, USA: John Wiley & Sons, 2009.
- [34] R. Chipalkatty, G. Droge, and M. B. Egerstedt, "Less is more: Mixed-initiative model-predictive control with human inputs," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 695–703, 2013.
- [35] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985.
- [36] Y. Meirovitch, D. Bennequin, and T. Flash, "Geometrical invariance and smoothness maximization for task-space movement generation," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 837–853, 2016.
- [37] A. D. Dragan, "Legible robot motion planning," Ph.D. dissertation, Robotics Institute, CMU, Pittsburgh, PA, 2015.
- [38] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. New York, NY, USA: John Wiley & Sons, 2006, vol. 3.
- [39] E. Pezent, C. G. Rose, A. D. Deshpande, and M. K. O'Malley, "Design and characterization of the openwrist: A robotic wrist exoskeleton for coordinated hand-wrist rehabilitation," in *Proc. IEEE Conf. Rehab. Robot.*, 2017, pp. 720–725.
- [40] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 932–938, 2000.
- [41] A. Gupta and M. K. O'Malley, "Disturbance-observer-based force estimation for haptic feedback," *J. Dyn. Syst. Meas. Control*, vol. 133, no. 1, p. 014505, 2011.
- [42] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 8, pp. 2126–2136, 2012.



Dylan P. Losey (S'14) received the B.S. degree in mechanical engineering from Vanderbilt University, Nashville, TN, USA, in 2014, and the M.S. degree in mechanical engineering from Rice University, Houston, TX, USA, in 2016.

He is currently working towards the Ph.D. degree in mechanical engineering at Rice University, where he has been a member of the Mechatronics and Haptic Interfaces Laboratory since 2014. In addition, between May and August 2017, he was a visiting scholar in the Interactive Autonomy and Collaborative Technologies Laboratory at the University of California, Berkeley. He researches physical human-robot interaction; in particular, how robots can learn from and adapt to human corrections.

Mr. Losey received an NSF Graduate Research Fellowship in 2014, and the 2016 IEEE/ASME Transactions on Mechatronics Best Paper Award.



Marcia K. O'Malley (SM'13) received the B.S. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 1996, and the M.S. and Ph.D. degrees in mechanical engineering from Vanderbilt University, Nashville, TN, USA, in 1999 and 2001, respectively.

She is currently a Professor of mechanical engineering and of computer science and of electrical and computer engineering at Rice University, Houston, TX, USA, and directs the Mechatronics and Haptic Interfaces Laboratory. She is adjunct faculty in the Departments of Physical Medicine and Rehabilitation at the Baylor College of Medicine and the University of Texas Medical School at Houston, and is the Director of Rehabilitation Engineering at TIRR-Memorial Hermann Hospital. Her research addresses issues that arise when humans physically interact with robotic systems, with a focus on training and rehabilitation in virtual environments.

Prof. O'Malley is a Fellow of the American Society of Mechanical Engineers and serves as an Associate Editor for the ASME Journal of Mechanisms and Robotics, the ACM Transactions on Human Computer Interaction, and the IEEE Transactions on Robotics.