

Trinity University

## Digital Commons @ Trinity

---

Engineering Faculty Research

Engineering Science Department

---

4-2021

# Vector Field Control Methods for Discretely Variable Passive Robotic Devices

Emma Treadway

Trinity University, [etreadwa@trinity.edu](mailto:etreadwa@trinity.edu)

R. Brent Gillespie

Trinity University

Follow this and additional works at: [https://digitalcommons.trinity.edu/engine\\_faculty](https://digitalcommons.trinity.edu/engine_faculty)



Part of the [Engineering Commons](#)

---

### Repository Citation

Treadway, E., & Gillespie, R. B. (2021). Vector field control methods for discretely variable passive robotic devices. *IEEE Transactions on Robotics*, 37(2), 375-389. doi: 10.1109/TRO.2020.3031255

This Post-Print is brought to you for free and open access by the Engineering Science Department at Digital Commons @ Trinity. It has been accepted for inclusion in Engineering Faculty Research by an authorized administrator of Digital Commons @ Trinity. For more information, please contact [jcostanz@trinity.edu](mailto:jcostanz@trinity.edu).

# Vector Field Control Methods for Discretely Variable Passive Robotic Devices

Emma Treadway *Member, IEEE*, and R. Brent Gillespie, *Member, IEEE*.

**Abstract**—Passive transmission-based robotic devices are capable of providing motion guidance while ensuring user safety and engagement. To circumvent some of the drawbacks associated with steering continuously variable transmissions based on rolling contacts, we are exploring a class of discretely variable devices, based on brakes and hydrostatic transmissions. Previously available control methods for discretely variable devices were built on velocity fields and only developed to stabilize a 1D target manifold. For  $n$ -DOF devices, methods to stabilize target manifolds of dimension 1 to  $n-1$  are of interest. In this paper we contribute constraint field methods that stabilize  $n-1$  dimensional target manifolds while leaving the orthogonal subspace free to the control of the operator. We also contribute force-modulated SDOF velocity fields, which add between 1 and  $n-2$  virtual DOF to the motion of devices whose physical constraints leave one DOF. Control performance is demonstrated in simulation for 3DOF devices capable of imposing 1D or 2D constraints and in experiment for 2DOF devices imposing 1D constraints. Our experimental apparatus features digital hydraulic transmissions that are easily configured for  $n$ -dimensional space and capable of imposing constraints of any dimension, thus motivating the contributed methods.

## I. INTRODUCTION

**I**DEAL motion guidance produces sufficient force to resist motion in certain directions yet allows the operator free motion along others. To the operator, ideal motion guidance feels like a hard wall, curve, or surface. Motion-guiding robots have been developed to facilitate human/robot collaborative assembly operations [1], to perform haptic rendering [2], to train manual skill [3], [4], and to rehabilitate motor function after neurological injury [5], [6], [7]. Both in manual skill training and robotic rehabilitation, it has been suggested that the human should take an active role to maximize therapeutic or training effects [8], [9], [10]. Energetically passive robotic devices are capable of fulfilling the goal of motion guidance by constraining motion in certain directions using transmissions in place of motors or other actuators found in more common

active robotic devices. They can thus perform motion guidance while benefiting from inherently safe user interaction, since the robot injects no power into the interaction. We are particularly interested in passive robotic devices in physical therapy, where paradigms that require the patient to generate the power can discourage so-called slacking behavior [11].

Passive motion guiding robots require users to power their own motion while the robot employs its variable transmission(s) under computer control to steer the user-generated motion onto and along a goal manifold in taskspace. Peshkin, Colgate, et al. [12] developed cobots, a class of nonholonomic guiding robots that rely on rolling contacts to realize continuously variable transmissions (CVTs) that reduce the available degrees of freedom (DOF). Cobots dictate the direction of motion by placing the steering (but not the rolling) of the wheels under computer control. Cobots can effectively enforce convergence to and motion along goal paths, but their realization involves certain design challenges: cobots require high preload on the rolling contacts to avoid sideslip when forces are applied perpendicular to the constrained direction. The need to structurally support these large preload forces can lead to devices that are quite large and heavy [2]. In some cases, routing power from the motion axes to be coupled to the CVT can also be challenging [13]. And furthermore, the inertia of the device and wheel mean that instantaneous direction change is not possible; this is a particular problem when trying to render freespace.

The alternative to a passive guiding robotic device based on CVTs is of course one based on discretely variable transmissions. A *discretely variable* passive robotic device has the ability to physically constrain motion in certain directions determined by the settings of the device's transmission(s), selecting transmission ratios from among a finite available set. We have previously proposed digital hydraulic transmissions [14] as a method for realizing discretely variable passive robotic devices. Digital hydraulic devices use hydrostatic transmissions in which multiple cylinders span each joint. Valves direct the flow from each cylinder into either the reservoir to allow free motion or into a common manifold that imposes a flow constraint. The resulting discretely variable transmission ratio is therefore based on the relative face areas at either end of the hydrostatic transmission that are connected via the manifold through switching. Digital hydraulic devices can overcome challenges associated with cobotic CVTs: hydraulic transmissions allow simple power routing through flexible tubing, enabling the realization of passive guiding exoskeletons. Hydraulics are also capable of overcoming the weight issues that are associated with high preload forces

The authors are with the Department of Engineering Science, Trinity University, San Antonio, TX 78212 USA and the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48109 USA, e-mail: etreadwa@trinity.edu and brentg@umich.edu

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. This work was supported by the both National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health, Grant No. R01EB019834, and by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1256260. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or NIH.

in rolling contact cobots, preventing slipping internally with pressurized fluid.

While digital hydraulic transmissions are capable of overcoming the design limitations associated with CVTs, they introduce a new control challenge. Due to the discrete set of continuous dynamics that these devices are capable of imposing, a hybrid control problem arises—we will discuss the challenges associated with this control problem more deeply in Section III. Certain brake-based robots can also be classified as discretely variable passive devices, and velocity-field-based controllers developed for these robots [15], [16] hold promise for other discretely variable passive devices.

When brakes are selectively used in an on/off manner to lock relative motion at joints, a braked linkage becomes a discretely variable passive robotic device, with the discretely variable paths through taskspace selected by locking motion in all but a single joint about which the device can rotate freely<sup>1</sup>. This approach came about when Sakaguchi et al. [17] proposed solving the problem of resistance encountered when moving *along* a virtual wall in a system with two brakes by alternating between the brakes, fully engaging one at a time such that an un-braked DOF was always available with no damping by rotating about the un-engaged brake.

This switching between two possible directions of motion was a precursor to what Swanson and Book later termed Single Degree of Freedom (SDOF) Velocity Field Control [15], [16], where the name “SDOF” refers to the single unconstrained 1-D path along which the operator can move freely when rotating about the un-braked joint. At each instant, the device employs the SDOF path that best matches the direction dictated by a continuous velocity field like the one in Fig. 1a. SDOF velocity field control is powerful because it provides a simple method of adapting established velocity field control methods developed for continuous active robots (e.g., [20], [21]) for use with discretely variable passive devices. However, SDOF methods were only developed for planar (2D) devices and are applicable only to creating 1D goal manifolds to which the user converges, even in devices of higher dimension. More generally, for  $n$ -dimensional taskspaces, it is desirable to be able to provide motion guidance towards and, once converged, along manifolds of any dimension between 1 and  $(n - 1)$ .

In this paper, we develop a control framework for inherently passive devices featuring discretely variable transmissions. We present a unified mathematical framework for describing different types of discretely variable devices in Section II. In Section III, we present potential control methods, focusing on a class of vector field-based controllers inspired by velocity fields. We present two extensions to single degree of freedom (SDOF) velocity field control [16] both of which are designed to enable the constraint of higher DOF manifolds. Both are illustrated thematically in Fig. 2: first, we add virtual DOF through force sensing. Second, we present a control method that is in a sense dual to the existing SDOF velocity field methods: it uses *constraint* fields (see Fig. 1b) to select from

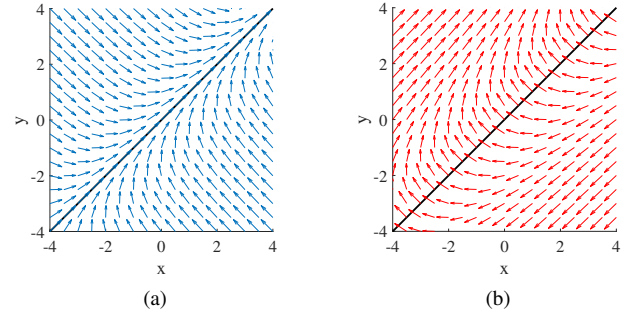


Fig. 1. (a) A continuous velocity field, created by blending with an exponential term [22]. At each point, the blue velocity field determines the desired tangent direction. (b) Continuous constraint field in 2D, using the same blending method. At each point, the red field determines the desired constrained direction. The velocity field in (a) and the constraint field in (b) encode the same path, generating identical control laws.

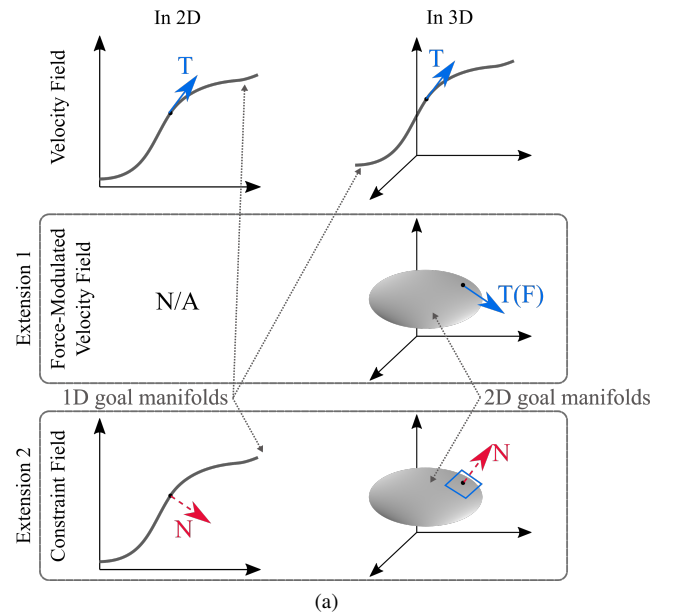


Fig. 2. Discretely passive devices can steer a user’s motion by discretely modulating the physically possible direction(s) of motion. SDOF velocity field control [16], creates a vector field of allowed headings that point towards a 1-dimensional goal manifold and uses the device’s 1-DOF transmission settings to approximate that field. We present two extensions to SDOF velocity field control approaches to impose goal manifolds of dimension higher than 1. (extension 1) Force modulation of the SDOF direction is an admittance-like modification in which the heading is modulated as a function of measured force as  $T(F)$ . (extension 2) A dual method to velocity fields involves modulation of a constraint vector perpendicular to the allowed direction(s) of motion.

manifolds with a single physically constrained direction of motion by modulating the constraint vector perpendicular to the allowed directions of motion. The controllers are then evaluated in simulation involving devices with 2 and 3 axes and in hardware experiments with two examples of digital hydraulic apparatus, each involving 2 axes.

## II. A MATHEMATICAL FRAMEWORK FOR DISCRETELY VARIABLE PASSIVE DEVICES

A discretely variable passive device will be capable of imposing certain constraints on the motion of its joints, possibly

<sup>1</sup>Many devices based on brakes ([17], [18], [19], [7], among others) have been proposed to guide motion or render virtual environments by resisting user motion. An important distinction is that devices that modulate braking force are **not** discretely variable.

applying more than one constraint at a time. We consider devices in which discretely-switched elements<sup>2</sup> determine the constraint(s) imposed during a given time interval. We propose in this section a unified mathematical framework for enumerating and describing the available constraints in a discretely variable passive device.

Let  $n$  denote the taskspace DOF of a given device and let  $m$  denote the number of DOF removed by discretely variable constraints, leaving  $p = n - m$  taskspace DOF under control of the operator. Let  $d$  describe the number of possible discrete switch settings available to be selected. And let  $a \leq n$  describe the maximum number of constraints that the device is able to impose (by construction) within any given time interval (over all  $d$  switch settings)<sup>3</sup>. For a device with  $N$  joints whose configuration is expressed in the vector  $\mathbf{q} \in \mathbb{R}^N$ , a constraint matrix  $\mathbf{C}_{\mathbf{q},k} \in \mathbb{R}^{N \times a}$  may be used to describe, over a given time interval, the imposed relationship

$$\mathbf{C}_{\mathbf{q},k}^T \dot{\mathbf{q}} = 0. \quad (1)$$

The subscript  $k \in [1, 2, \dots, d]$  selects which one of the  $d$  switch settings pertains within the given time interval.

Furthermore, let the  $n$ -dimensional taskspace configuration  $\mathbf{R} \in \mathbb{R}^n$  be defined by the forward kinematics  $\mathbf{R} = \psi(\mathbf{q})$ . Then we define the Jacobian matrix  $\mathbf{J}_2 \in \mathbb{R}^{n \times N}$  as  $\mathbf{J}_2(\mathbf{q}) = \partial\psi/\partial\mathbf{q}$ . In taskspace, the constraint is expressed as

$$\mathbf{C}_k^T \dot{\mathbf{R}} = 0. \quad (2)$$

The Jacobian  $\mathbf{J}_2$  can of course be used to relate the taskspace and jointspace constraint matrices, with

$$\mathbf{C}_{\mathbf{q},k}^T = \mathbf{C}_k^T \mathbf{J}_2. \quad (3)$$

To describe the number of constraints imposed during a time interval with a given switch setting  $k$ , we use the symbol  $m_k$ . Thus Eq. (2) constrains  $m_k$  DOF, where  $m_k \leq a$  is the row rank of  $\mathbf{C}_k^T$ . Then there remain  $p_k = n - m_k$  allowed directions of motion for each setting. In taskspace, these allowed directions lie in the nullspace  $\mathbf{C}_k^\perp \in \mathbb{R}^{n \times p_k}$  of the constraint matrix; in jointspace, they lie in  $\mathbf{C}_{\mathbf{q},k}^\perp$ . It will often be convenient to enumerate allowed motion directions in jointspace ( $\mathbf{C}_{\mathbf{q},k}^\perp$ ) and then transform them to taskspace rather than directly enumerating allowable settings for  $\mathbf{C}_k^\perp$  in taskspace<sup>4</sup>.

All  $d$  settings of the device may not yield unique values for allowed directions, as some settings may be redundant [23]. We define  $j \leq d$  to be the number of unique allowed direction settings in taskspace (i.e., independent values of  $\mathbf{C}_k^\perp$  that the device can take on) that yield  $1 \leq p_k < n$ . To prepare for the process of designing controllers that manage the switch settings, we further subdivide the  $j$  unique settings by the DOF  $p_k$  that remain for a switch setting  $k$ . In addition to possible freespace ( $p_k = n$ ) and locked ( $p_k = 0$ ) settings, a given device design will have  $j_{(1)}$  SDOF allowed motions (with  $p_k = 1$ ),  $j_{(2)}$  2-DOF settings (with  $p_k = 2$ ), and so forth, with  $\sum_{i=1}^{n-1} j_{(i)} = j$ .

<sup>2</sup>The valve or brake settings, in our examples.

<sup>3</sup>In a digital hydraulic device,  $a$  is the number of separate fluid manifolds.

<sup>4</sup>If  $\mathbf{J}_2$  is full rank, the inverse or pseudoinverse may be employed to find  $\mathbf{C}_k$  given  $\mathbf{C}_{\mathbf{q},k}$ , but this is computationally more expensive.

We will make use of two special cases, to which we assign special names. When  $p_k = 1$ , we define the single allowed direction  $\mathbf{T}_k = \mathbf{C}_k^\perp$  to be an enforced *heading* in taskspace. On the other hand, when  $m = 1$ , we define a *constraint vector*  $\mathbf{N}_k$  in taskspace, where  $\mathbf{N}_k^T$  is a normalized nonzero column<sup>5</sup> of  $\mathbf{C}_k$ ;  $\mathbf{N}_k$  will be normal to all  $n - 1$  allowed directions of motion. The vectors  $\mathbf{T}_{\mathbf{q},k}$  and  $\mathbf{N}_{\mathbf{q},k}$  are the corresponding values expressed in jointspace.

Headings can be transformed back and forth between jointspace ( $\mathbf{T}_{\mathbf{q}}$ ) and taskspace ( $\mathbf{T}$ ) via the Jacobian  $\mathbf{J}_2$ :

$$\mathbf{T} = \mathbf{J}_2 \mathbf{T}_{\mathbf{q}} / |\mathbf{J}_2 \mathbf{T}_{\mathbf{q}}|. \quad (4)$$

The transformations for the constraint vector  $\mathbf{N}$  are derived from the relationship (2), since  $\mathbf{N}$  is simply a normalized column of  $\mathbf{C}_k$ :

$$\mathbf{N}_{\mathbf{q}} = \mathbf{J}_2^T \mathbf{N} / |\mathbf{J}_2^T \mathbf{N}|. \quad (5)$$

When  $\mathbf{J}_2$  is full rank, its inverse (or pseudoinverse, for  $n \neq N$ ) can be employed for transformations in the other direction:

$$\mathbf{T}_{\mathbf{q}} = \mathbf{J}_2^{-1} \mathbf{T} / |\mathbf{J}_2^{-1} \mathbf{T}|, \quad (6)$$

and

$$\mathbf{N} = (\mathbf{J}_2^{-1})^T \mathbf{N}_{\mathbf{q}} / |(\mathbf{J}_2^{-1})^T \mathbf{N}_{\mathbf{q}}|. \quad (7)$$

Note that we have dropped the subscript  $k$  in equations (4)-(7), since they can be used to transform *any* heading or constraint vector, including a desired heading/constraint, not just the ones available for a given device.

#### A. Application to Locking Linkages

We take the simplest interesting example for illustration here: an open-loop linkage<sup>6</sup> with motion in the plane. We assume the presence of a brake or clutch at each of the  $N$  joints that is capable of locking the relative motion. Thus,  $a = N$ , and the constraint matrix becomes

$$\mathbf{C}_{\mathbf{q},k} = \text{diag}(s_1(k), s_2(k), \dots, s_N(k)), \quad (8)$$

where  $s_i(k)$  takes on a value of 1 if joint  $i$  is locked, or 0 if joint  $i$  is free to move for a particular setting  $k$ . This results in  $d = 2^N$  possible settings. In cases where  $N > n$ , each setting will remove at most  $m_k = n$  DOF.

In a 2-link device, for example,  $\mathbf{C}_{\mathbf{q},k}^\perp \in \{\mathbf{I}_{2 \times 2}, [1 \ 0]^T, [0 \ 1]^T, [0 \ 0]^T\}$ . Thus we see that  $d = 4$  and  $j = 2$ . This device has the ability to achieve freespace, to achieve two  $j_{(1)} = 2$  SDOF motions that correspond to motion of one joint alone or the other joint alone, and the ability to lock motion entirely. In general, a planar  $N$  link open linkage with  $N$  brakes (one at each joint) might be able to constrain  $j_{(1)} = N$  unique SDOF directions, each corresponding to rotation about a single unlocked joint with the other  $N - 1$  brakes engaged to prevent relative motion between adjacent links.

<sup>5</sup>In the case that there is more than one nonzero column, they will be linearly dependent in any case, since  $m = 1$ .

<sup>6</sup>Extension to closed-loop linkages like PTER will require proper selection of unique jointspace variables and description of the imposed constraints.



### B. Application to Digital Hydraulics

In a digital hydraulic cobot [14], a set of cylinders with face areas collected into a diagonal matrix  $\mathbf{A}$  are connected to  $N$  joints. The nature of the connection and the moment arms between the cylinders and the joints  $\mathbf{q}$  are encoded in the Jacobian matrix  $\mathbf{J}_1(\mathbf{q})$ . Fluid from the cylinders is routed by solenoid valves, directing the flow from each cylinder either into a fluid manifold or a reservoir; these connections are encoded in the switching matrix  $\tilde{\mathbf{S}}$ , where each column represents a separate fluid manifold and each row represents a cylinder. Each element is a 1 if the cylinder is connected into the corresponding manifold and a 0 otherwise; if no manifold connections are present for a given cylinder, it is connected into the reservoir. The constraints in the system arise from the requirement that flow into each manifold must sum to 0:

$$\tilde{\mathbf{S}}^T \mathbf{A} \mathbf{J}_1 \dot{\mathbf{q}} = \mathbf{0}. \quad (9)$$

Thus, for each of our  $k$  possible switch settings, the constraint matrix takes on the value

$$\mathbf{C}_{\mathbf{q},k}^T = \tilde{\mathbf{S}}_k^T \mathbf{A} \mathbf{J}_1. \quad (10)$$

Depending on the dimension  $n$  of the device taskspace and the dimension of  $\mathbf{C}_{\mathbf{q},k}^\perp$ , motion with a certain valve setting may be allowed in all directions ( $p = n$ , as when all cylinders are connected to the reservoir), a single heading ( $p = 1$ ), or a higher dimensional manifold<sup>7</sup> (e.g., when  $\mathbf{C}_{\mathbf{q},k}^\perp \in \mathbb{R}^{3 \times 2}$ , a  $p = 2$  plane through a 3D space is imposed).

Note that, as defined in [14], we do not allow kinematic redundancy in the jointspace for digital hydraulic devices since they already have redundancy of cylinders at the joints—barring singularities,  $\mathbf{J}_2$  should therefore be invertible. In our examples here, we will work with digital hydraulic devices with  $\mathbf{J}_1$  being a constant matrix (achieved by mounting cylinders in opposing pairs about a pulley), such that  $\mathbf{C}_{\mathbf{q},k}$  is not a function of  $\mathbf{q}$ ; while not strictly necessary, this enables enumeration of the constraints in jointspace where they can more easily be visualized and understood.

### III. CONTROL METHODS

Switching between sets of passive dynamics is not a problem unique to discretely variable passive haptic devices. The switching sequence synthesis problem has been explored more abstractly in the field of hybrid control. While many techniques exist for hybrid control synthesis, the unique conditions for this application rule out quite a few common solutions; the main challenges in this case are the potentially large number of discrete modes ( $j$ ) and the fact that mode selection is the only input under computer control. Many techniques for synthesizing switching sequences break down with more than 2 or 3 discrete modes (e.g., [24]). And common hybrid model predictive control (MPC) techniques such as [25] require control over continuous inputs. Another relevant approach in the hybrid control literature is the work by Prabhakar and Garcia Soto to find stability of polyhedral switched systems through model-checking [26], [27], [28]. However,

the proposed methods apply only to analyzing the stability of a system, rather than to synthesizing a path that converges.

The most promising approach we have found in the hybrid control literature is to create a mixed integer MPC problem with a set of auxiliary binary variables, inspired by [29]. The resulting formulation is a set of constraints for a mixed integer quadratic programming problem, which can be used to optimize a cost function penalizing, for example, error and the number of switches. Unfortunately, as the number of available discrete modes becomes large, the problem can become infeasible to solve in real time. While we acknowledge the potential of hybrid control approaches, nothing in our survey of the current literature suggests a method which will yield practical methods for synthesis of switching sequences to guide motion with a discretely variable passive device. We therefore focus our efforts in this paper on extending the SDOF velocity field methods previously used on braked devices.

SDOF control of a linkage-based haptic display [16], [15], [30] makes use of redundancy in a linkage. Switching brakes on and off allows selection from the set of  $j_{(1)}$  available SDOF paths by locking relative motion between adjacent links as described in Section II-A. The existence of at least  $n$  joints also enables free motion in any direction when the brakes are not engaged. Early work [15] defined a priori shapes (obstacles) made up of possible SDOF paths into which penetration was not allowed—once a boundary was crossed, the corresponding constraint was engaged; however, this method cannot be used to converge to a desired path nor does it enable steering out of an obstacle if penetration occurs. The method described in [16] by Reed and Book makes use of a velocity field to encode a desired path. At each point, the field dictates the desired heading; the closest available approximation of that heading is selected by the controller; this is called velocity field SDOF control.

While velocity field SDOF control can be used to impose  $p = 1$  DOF goal manifolds in an  $n$ -dimensional space, certain applications (e.g., rendering surfaces within a 3D virtual environment) would benefit from the ability to impose higher DOF goal manifolds. In the following we first recap the velocity field SDOF control method as adapted to our framework for a generic discretely variable device, and we then propose two extensions that follow a generic vector-field-based control framework (Fig. 3a): one which creates the illusion of additional DOF through feedback control while employing the available SDOF paths, and a second which makes use of multiple physical DOF in a discretely variable haptic device. The first method will take inspiration from cobot freespace rendering [12] to alter a velocity field on-the-fly in response to force inputs to add  $\ell$  virtual DOF to an  $m = n - 1$  system; the second will rely on the constraint vectors which can be defined for  $m = 1$  systems, allowing free motion across the remaining  $n - 1$  directions that remain physically unconstrained. As illustrated in Fig. 3b, each extension of velocity field methods is capable of creating constraints of different DOF depending on the native dimension of the space.

Based on our particular motivating application (rehabilitation), we have chosen to implement controllers in this paper such that the user is always guided towards the desired goal

<sup>7</sup>A mathematical manifold, not to be confused with a hydraulic manifold.

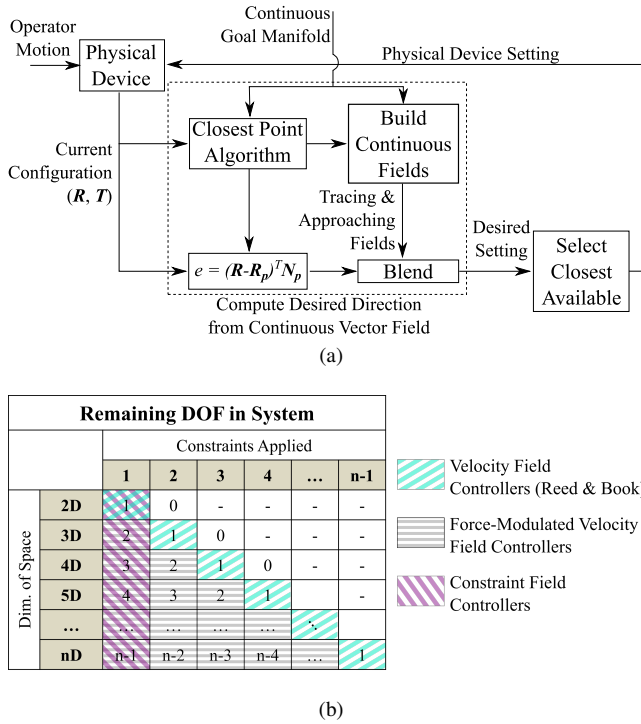


Fig. 3. Overview of discussed controllers. (a) Field-based control methods at a high level contain the illustrated steps, common to the original SDOF velocity field method [16] and our two proposed extensions. (b) SDOF velocity field control relies on the application of  $n - 1$  constraints in an  $n$ -dimensional space to impose a single remaining DOF. Force-modulated velocity fields (extension 1) add  $\ell$  DOF through force sensing to create constraints of arbitrary dimension  $> 1$ , while constraint fields (extension 2) achieve  $n - 1$  DOF without force sensing.

manifold. For haptic rendering, these controllers could of course be paired with freespace (i.e., a selection of none of the constraints) outside of the goal manifolds.

We will develop the controllers in taskspace coordinates,  $\mathbf{R}$ . However, some desired paths or manifolds might be described in jointspace instead, particularly in the case of rehabilitation applications. For desired paths defined in jointspace, it is simple to define  $\mathbf{R} = \mathbf{q}$ , yielding the Jacobian  $\mathbf{J}_2 = \mathbf{I}$ .

#### A. Velocity Field SDOF Control

Methods for designing velocity fields have been proposed in several different areas of study—for example, machining [31], quad-rotor control [22], and rehabilitation robotics [21], [32]. Velocity field methods are desirable since they can enforce a path without predetermining timing, and they are convergent by design—traits which are also desirable for passive rendering. Some methods developed for active manipulators have specifically tried to mimic passive interaction for safety and stability reasons [20], [31]. For use with discretely variable SDOF paths, a continuous is first designed, and then a closely-matched SDOF path is selected from those available, as illustrated in Fig. 3a.

1) *Designing Velocity Fields:* Generally, velocity fields designed for path following share the following characteristics:

- 1) The field along the desired path is tangent to that path. This tangent direction is encoded in a tracing field  $\mathbf{V}_{tr}$ .

- 2) Far from the desired path, the velocity field points toward (normal to) the desired path. An approaching field  $\mathbf{V}_{ac}$  encodes this normal direction.
- 3) At intermediate distances from the desired path, the velocity field blends between these tangent and normal directions (based on the distance from the path).

In this way, the distance-weighted blend of  $\mathbf{V}_{tr}$  and  $\mathbf{V}_{ac}$  yields the desired velocity field heading  $\mathbf{V}_{des}$  to encode the path.

The closest point  $\mathbf{R}_p$  along the goal curve is associated with the curve's local path tangent vector  $\mathbf{T}_p$ . These are used to define

$$\mathbf{V}_{ac} = (\mathbf{R}_p - \mathbf{R}) / \|\mathbf{R}_p - \mathbf{R}\| \quad (11)$$

and

$$\mathbf{V}_{tr} = \mathbf{T}_p. \quad (12)$$

In 2D ( $n = 2$ ), Eq. (11) is equivalent to  $\mathbf{V}_{ac} = -\text{sign}(e)\mathbf{N}_p$ , where  $\mathbf{N}_p \perp \mathbf{T}_p$  is the normal to the goal path and

$$e = (\mathbf{R} - \mathbf{R}_p)^T \mathbf{N}_p \quad (13)$$

is the taskspace configuration error from the goal path. If motion along the path is to be allowed in either direction based on the user's applied force  $\mathbf{F}$ , then the tracing field can be modified<sup>8</sup> as

$$\mathbf{V}_{tr} = \text{sign}(\mathbf{F}^T \mathbf{T}_p) \mathbf{T}_p. \quad (14)$$

There are multiple ways to blend between the approaching and tracing fields as an approach is made to the goal path. In the method from [16], a “path boundary” is defined at a certain distance from the desired path. Outside that boundary,  $\mathbf{V}_{des} = \mathbf{V}_{ac}$ ; inside the boundary, a linear blend is made from  $\mathbf{V}_{ac}$  to  $\mathbf{V}_{tr}$  as the Euclidean distance to the path goes to 0. Here, we make use of a second method, described in [22] which uses a continuous blending via a parameter  $\gamma$  that can be tuned to determine the desired heading

$$\mathbf{V}_{des} = w_1 \mathbf{V}_{ac} + w_2 \mathbf{V}_{tr}, \quad (15)$$

where

$$w_1 = \frac{2}{1 + e^{-\gamma|e|}} - 1, \quad w_2 = 1 - w_1. \quad (16)$$

Such a velocity field was shown in Fig. 1a with  $\gamma = 1$ .

While most papers pertaining to velocity fields use desired paths defined along lines or circles for simplicity of finding the closest point  $\mathbf{R}_p$ , the method is more broadly applicable; closest-point algorithms and extensions to more generic curves such as NURBS have been made [33], [34], along with extensions for dealing with self-intersecting contours [35]. While workspace envelope constraints are not automatically considered, it would also be relatively trivial to modify the vector field near the limits of travel to point back into the center of the workspace<sup>9</sup>.

<sup>8</sup>Or, alternatively, the user's instantaneous heading  $\mathbf{T}$  may be used in place of  $\mathbf{F}$  to determine the direction of  $\mathbf{V}_{tr}$ . Naturally, the discretely variable device enforces  $\mathbf{T} = \pm \mathbf{T}_k$ .

<sup>9</sup>Since the velocity field selects only a heading while allowing the user to control the velocity along it, it is possible for the operator to select negative “combative” velocities as well as positive “cooperative” ones. Convergence will occur along the velocity field when cooperative velocities are generated, while combative velocities will cause the user to travel directly away from the desired direction, eventually reaching a motion limit in one or more axes.

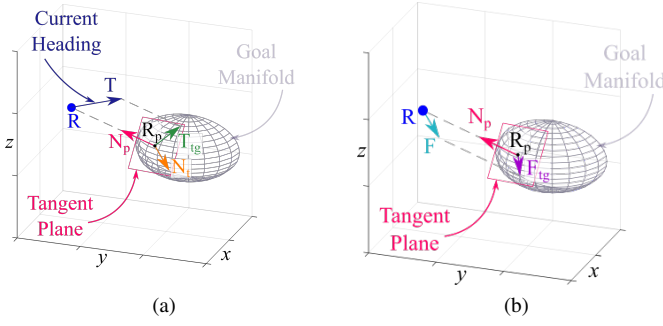


Fig. 4. Steps in creating a force-modulated velocity field, illustrated schematically for  $n = 3$ ,  $\ell = 1$ , (a) the current heading  $T$  is projected into the goal manifold to produce  $T_{tg}$  using the surface normal  $N_p$  at the closest point. (b) The user-applied force can be similarly be projected into the goal manifold to produce  $F_{tg}$ . The planar steering force magnitude  $F_{\perp}$  is  $F^T N_t$ .

2) *Selection from Discrete Directions*: Once  $V_{des}$  has been determined, a further step must be taken which distinguishes SDOF control from continuous velocity field methods: the desired heading must be matched to the available headings  $T_{av} \in \{T_1, \dots, T_{j(1)}\}$ , and the best candidate selected. This can be accomplished by finding which available heading has the largest projection onto  $V_{des}$ .

If the available headings are enumerated in jointspace as  $T_{q,av}$ , a transformation using the Jacobian  $J_2$  is required for goal paths defined in taskspace, either by transforming the available headings to taskspace via Eq. (4), or by transforming  $V_{des}$  to jointspace with Eq. (6). In [16], this matching was done in taskspace—for redundant linkages with non-square  $J_2$  ( $n \neq N$ ), this application of (4) to the enumerated  $T_{q,av}$  in jointspace circumvents the need to perform inverse kinematics.

### B. Extension 1: Force-Modulated Velocity Field

This method enables the constraint of  $n-m+\ell$  dimensional manifolds in  $n$  dimensional spaces using the  $j(1)$  SDOF constraints available from the device. It can therefore be used to impose 2D manifolds in 3D space. Superficially, it is very similar to the standard velocity control method described in Section III-A, blending between an approaching field  $V_{ac}$  and a tracing field  $V_{tr}$ . However, to enable motion in additional DOF, the selected transmission ratio must also use the applied user force, as was also the case for the addition of extra DOF in freespace control of cobots [36]. Note that this adds an additional input to the “Build Continuous Fields” block in Fig. 3a.

The approaching field  $V_{ac}$  is still defined as in (11). However, a modification is made to the tracing field based on the user’s applied direction of force. We first project the current heading  $T = \dot{R}/\|\dot{R}\|$  onto the goal manifold

$$T_{tg} = (I - N_p N_p^T) T, \quad (17)$$

where  $N_p$  is the normal to the goal manifold at the closest point, as illustrated in Fig. 4a. Note that for goal manifolds embedded in spaces of dimension higher than 3, there may be more than one vector normal to the goal manifold at the closest point, and  $N_p$  in the above equation may be

replaced with a matrix assembled from these vectors,  $C_p = [N_{p1}, N_{p2}, \dots, N_{p(m-\ell)}]$ . In the absence of user-applied steering forces, the nominal tracing field should be

$$V_{tr,nom} = T_{tg}. \quad (18)$$

As shown in Fig. 4b for  $n = 3$ ,  $\ell = 1$ , the user-applied force  $F$  can be decomposed into a normal force (towards/away from the goal manifold),

$$F_n = -F^T N_p, \quad (19)$$

and the component representing the remaining tangential forces applied to steer across the goal manifold,

$$F_{tg} = (I - N_p N_p^T) F. \quad (20)$$

The tangential component of the force can then be used to adjust the direction of the tracing field  $V_{tr}$  to enable user-directed steering. Depending on the desired responsiveness, one could select (highly responsive)

$$V_{tr} = F_{tg}. \quad (21)$$

Or, for a less responsive controller,  $V_{tr}$  can be defined as the vector  $T_{tg}$  rotated in the direction of the forces applied across the surface of the goal manifold that do not align with  $T_{tg}$ . For example, to impose a 2D manifold in an  $n = 3$  space, a tunable weighting parameter  $\gamma_2$  could be introduced to rotate the field based on the magnitude  $F_{\perp}$  of the applied force in the direction of  $N_t = T_{tg} \times N_p$ :

$$V_{tr} = \lambda_1 \text{sign}(F_{\perp}) N_t + \lambda_2 V_{tr,nom}, \quad (22)$$

where the terms are weighted according to

$$\lambda_1 = \frac{2}{1 + e^{-\gamma_2 |F_{\perp}|}} - 1, \quad \lambda_2 = 1 - \lambda_1. \quad (23)$$

The disallowed forces  $F_{\perp}$  may then be expressed as

$$F_{\perp} = F^T N_t. \quad (24)$$

With  $V_{tr}$  and  $V_{ac}$  defined in this way,  $V_{des}$  is computed just as in Eq. (15), and then matched to the available SDOF constraints through projection of  $V_{des}$  onto each member of  $T_{av}$ . If steeper convergence down to the surface is desired when the operator pushes strongly down, the blending of  $V_{ac}$  and  $V_{tr}$  could be adjusted based on the magnitude of  $F_n$  (i.e., by changing the value of  $\gamma$  to dynamically alter the blend between the approaching and tracing fields); recall, however, that with or without this modification, the speed is always controlled by the user.

### C. Extension 2: Constraint Fields

In a second extension, we enable the imposition of goal manifolds of dimension  $n - 1$ , making use of the constraint vectors  $N_k$  corresponding to  $m_k = 1$ . While implementing a force-modulated velocity field requires force sensing, a constraint field is achievable with only position measurements; the tradeoff is the limitation to  $n - 1$ -dimensional goal manifolds, as was illustrated in Fig. 3b. The available unique constraint vectors from which our controller may select can be enumerated as  $\{N_1, \dots, N_{j(n-1)}\} \in N_{av}$ , which will

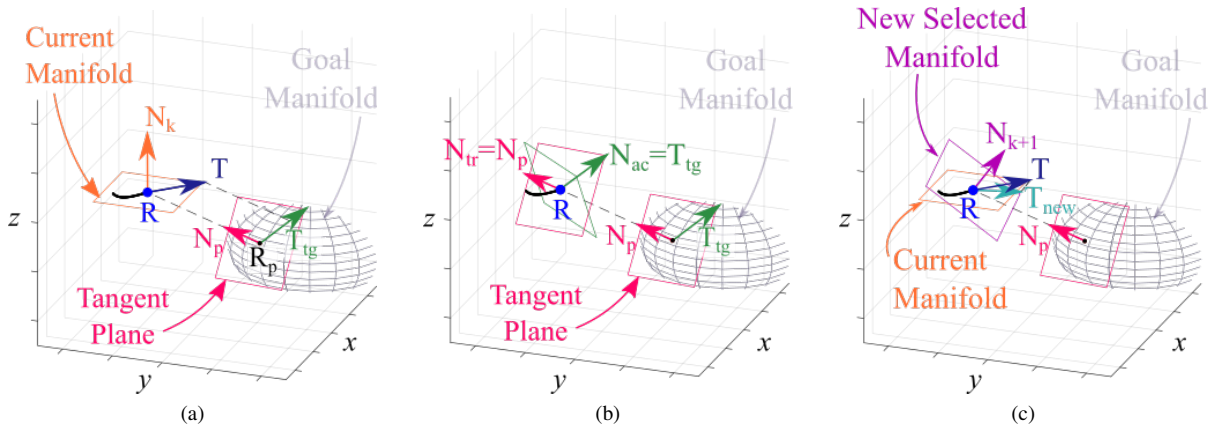


Fig. 5. Constraint field control, shown for  $n = 3$ ,  $e > 0$ . (a) Free motion by the operator across the currently imposed manifold (orange) has instantaneous direction  $T$ . (b) Using the geometry of the goal manifold and the projection of the current heading into the plane tangent to the goal manifold at the closest point, the approaching and tracing constraint fields are constructed. (c) Two consecutive selected manifolds are shown in 3D: selection of the new plane (purple) by blending of the approaching and tracing constraint vectors will redirect the user's motion to the direction  $T_{new}$ .

describe the direction perpendicular to the  $j_{(1)}$  unique 1DOF paths if  $n = 2$ , the  $j_{(2)}$  unique 2DOF surfaces if  $n = 3$ , and so forth.

Building on the general framework for vector-field-based control (Fig. 3a), we will define a *constraint* field rather than a velocity field that will determine the appropriate constraint vector based on the desired  $n - 1$  dimensional goal manifold to be imposed. Then, the two fields will be blended and the closest available constraint selected, still following the general outline of Fig. 3a. The goal manifolds can be defined in either task- or jointspace, just as in the velocity field method. In 2D, the method will in fact be precisely equivalent to the velocity field method, since the heading and constraint vectors uniquely determine one another. In 3D, however, it will enable convergence to 2D goal manifolds such as planes or the surface of spheres or other shapes.

We first define approaching and tracing constraint fields and blend them based on the distance  $e$  from the desired goal manifold, defined as in Eq. (13). The tracing constraint field is defined by the direction  $N_p$  perpendicular to the allowed directions in the goal manifold at the closest point  $R_p$ :

$$N_{tr} = N_p. \quad (25)$$

The approaching constraint field  $N_{ac}$  is selected to steer the user's current velocity down towards the desired surface. We use  $T_{tg}$ , the projection of  $T$  onto the goal manifold defined in (17), to form the approaching field:

$$N_{ac} = \text{sign}(e)T_{tg} \quad (26)$$

as illustrated in Fig. 5. In 2D,  $T_{tg} = T_p$  since  $N \perp T$ . If the motion is directly toward the closest point  $R_p$  such that  $\|T_{tg}\| = 0$ , any tangent direction in the plane can be selected as  $N_{ac}$ , since tilting in any direction should equally direct motion onto the goal manifold.

We then blend the two constraint fields as we did the velocity fields, for example using (16):

$$N_{des} = w_1 N_{ac} + w_2 N_{tr}. \quad (27)$$

An example of a constraint field in a 2D space was shown in Fig. 1b with  $\gamma = 1$ . Note that since  $\gamma$  here is selected to be

the same as in the velocity field of Fig. 1a, these two 2D fields are encoding the same goal path; each constraint vector in Fig. 1b is perpendicular to the heading vector at the same point in Fig. 1a, illustrating the equivalence of the two methods in 2D.

The constraint to be imposed is then selected by projecting  $N_{des}$  onto each member of the set of available constraints  $N_{av}$  and selecting the closest available constraint vector, which has  $\max(|N_{des}^T N_{av}|)$ . An example of how the motion is redirected towards the plane through this method is illustrated in Fig. 5c. Note that if the available constraint vectors are enumerated in jointspace as  $N_{q,av}$ , appropriate transformations of the constraint vectors need to be made, transforming  $N_{q,av}$  to  $N_{av}$  via Eq. (7) or transforming  $N_{des}$  to  $N_{q,des}$  via Eq. (5).

#### IV. TESTING METHODS

Both simulation and hardware testing were used to demonstrate the performance and features of these controllers. Simulations were performed in both 2D and 3D, constraining paths in joint- and taskspace. A 2D hardware apparatus was additionally used to test the controllers which are applicable in 2D (SDOF velocity field controller and constraint field controller), in both jointspace and taskspace hardware configurations.

##### A. Simulation Testing

Simulation results were generated using an ideal model of a transmission, assuming a positive user velocity along the path. The link lengths of the linkages used to produce taskspace transformations were selected to correspond to the upper and lower arm lengths of a human user of an exoskeleton linking the elbow and shoulder joints, inspired by our desired application in rehabilitation.

1) *2D Simulation*: A set of 2D simulations were performed assuming connection of single-acting cylinders with equal face area  $A$  connected in antagonistic pairs. Each pair forms a



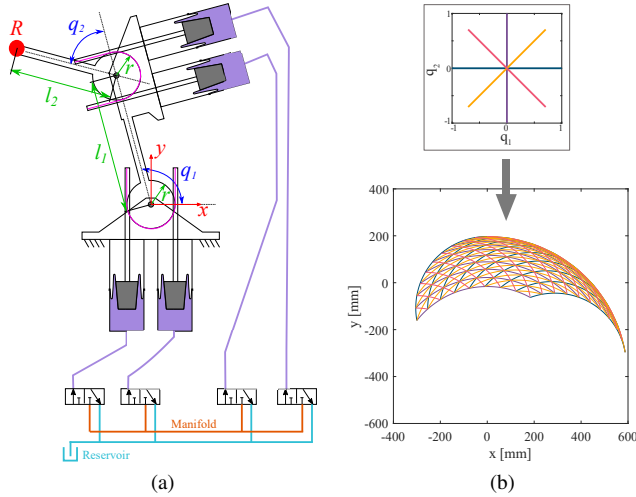


Fig. 6. (a) Two-link digital hydraulic arm configuration used for simulations, featuring two “modules” at the elbow and shoulder joints, each connecting two cylinders to the joint via cables wrapped around a pulley. (b) The 4 unique jointspace headings for the 2D configuration yield configuration-dependent SDOF lines through taskspace.

“module” of two cylinders connected in opposite directions to a pulley of radius  $r$  as shown in Fig. 6a. Thus,

$$\mathbf{A}\mathbf{J}_1 = \begin{bmatrix} Ar & 0 \\ -Ar & 0 \\ 0 & Ar \\ 0 & -Ar \end{bmatrix}. \quad (28)$$

Four valves  $\tilde{\mathbf{S}}^T = [s_1 \ s_2 \ s_3 \ s_4]$  imposed the variable manifold constraint (9), yielding  $d = 4$  unique possible values of the constraint matrix,  $\mathbf{C}_{q,k}^\perp \in \{[1 \ 0]^T, [1 \ 1]^T, [0 \ 1]^T, [-1 \ 1]^T\}$ .

For simulations in which paths were defined in taskspace rather than jointspace, the transformation from  $\mathbf{q}$  to taskspace coordinates  $\mathbf{R}$  was achieved through a linkage with link lengths  $l_1 = 292\text{mm}$  and  $l_2 = 305\text{mm}$ , yielding the Jacobian

$$\mathbf{J}_2 = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}, \quad (29)$$

where  $s_1 = \sin(q_1)$ ,  $c_2 = \cos(q_2)$ ,  $s_{12} = \sin(q_1 + q_2)$  and so forth. We assumed joint travel limits of  $15^\circ \leq q_1 \leq 85^\circ$  and  $0^\circ \leq q_2 \leq 100^\circ$  to correspond to the physical device described in the following hardware testing section. For the 4 unique selectable headings, the allowed SDOF motions are as shown in Fig. 6b.

2) *3D Simulation*: For simplicity and clarity, we started with the planar linkage (2 modules shown in Fig. 6a). We added two additional modules to connect the elbow motion to the shoulder ab/adduction joint. The three joint positions were therefore, in order,  $q_1$  measuring horizontal shoulder rotation,  $q_2$  measuring vertical shoulder ab/adduction, and  $q_3$  measuring elbow flexion. This connection of cylinder modules to the joints yielded the relationship

$$(\mathbf{A}\mathbf{J}_1)^T = Ar \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix}. \quad (30)$$

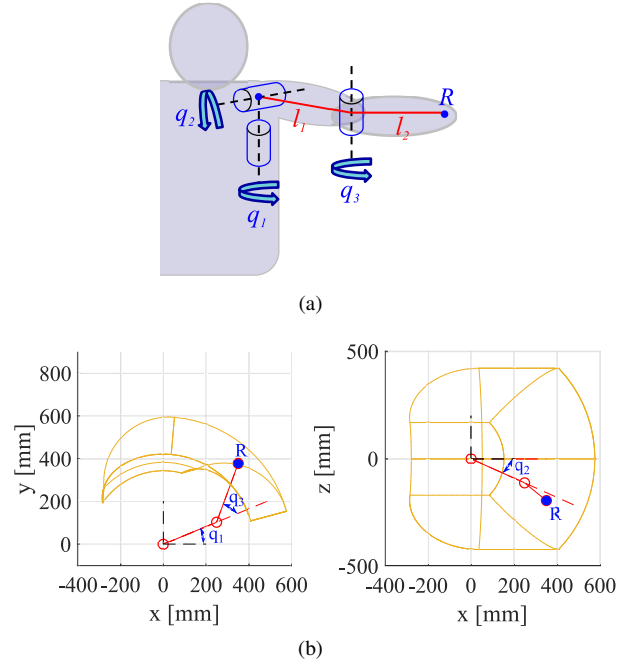


Fig. 7. Taskspace geometry used for 3D simulations. (a) The endpoint configuration  $\mathbf{R}$  is determined by the jointspace configuration  $\mathbf{q}$ , with  $q_1$  measuring horizontal shoulder rotation,  $q_2$  measuring vertical shoulder ab/adduction, and  $q_3$  measuring elbow flexion. (b) The envelope for the given joint travel limits is shown outlined in gold.

Along with upper arm length  $l_1$  and lower arm length  $l_2$ , the joint positions were used to determine the position of the endpoint in space as shown in Fig. 7, forming the taskspace variable  $\mathbf{R}$ :

$$\mathbf{R} = \begin{bmatrix} l_1 c_1 c_2 + l_2 (c_1 c_2 c_3 - s_1 s_3) \\ l_1 s_1 c_2 + l_2 (s_1 c_2 c_3 + c_1 s_3) \\ -l_1 s_2 - l_2 s_2 c_3 \end{bmatrix}.$$

We assumed joint travel limits of  $15^\circ \leq q_1 \leq 85^\circ$ ,  $-45^\circ \leq q_2 \leq 45^\circ$ , and  $0^\circ \leq q_3 \leq 100^\circ$  to match limits for  $q_1$  and  $q_3$  with the limits used in the 2D planar case, and adding a reasonable human range of motion in  $q_2$ .

The 8 cylinders on the device were connected to two separate manifolds via valves:

$$\tilde{\mathbf{S}}^T = \begin{bmatrix} s_{11} & s_{12} & 0 & 0 & s_{15} & s_{16} & 0 & 0 \\ 0 & 0 & s_{23} & s_{24} & 0 & 0 & s_{27} & s_{28} \end{bmatrix}, \quad (31)$$

yielding  $k = 2^8$  possible settings. This setup ensured that  $q_1$  and  $q_3$  were linked through one fluid manifold and that  $q_2$  and  $q_3$  were linked through the second fluid manifold, without the possibility that the two manifolds were ever connected together. Overall, this manifold connection scheme yields  $j_{(1)} = 11$  straight SDOF paths and  $j_{(2)} = 7$  2D planes that can be constrained in the 3D jointspace.

## B. Hardware Testing

By testing on physical devices, we sought to verify that these controllers could enable path guidance in non-ideal conditions, since the unrealistic assumption was made in the simulations that a digital hydraulic system functions as a

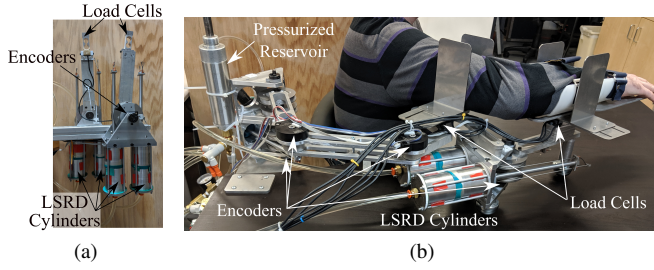


Fig. 8. Experimental apparatus for (a) jointspace testing, and (b) taskspace testing (exoskeleton apparatus).

perfect transmission without any parasitic effects. We demonstrated the functionality of these controllers on an  $n = 2$  physical digital hydraulic apparatus. This allowed hardware testing of both the SDOF velocity field controller and the constraint field controller.

As described in section IV-B, the force-modulated velocity field controller was proposed to impose manifolds of dimension  $n - m + l$ . In a space with dimension  $n = 2$  where the transmission can constrain  $m = 1$  DOF, this would mean imposing a manifold of dimension 2 (freespace)—the result would be a trivial demonstration of steering in the direction of applied force. This controller therefore cannot be demonstrated in any meaningful way with  $n < 3$ ; we leave hardware demonstration of this controller, along with higher dimensional implementations of the other two controllers, as future work.

1) *Jointspace Testing*: Four cylinders were connected via valves (ASCO 8320G212-24/DC) to a single manifold, with the valves powered by MOSFET switches (SparkFun DEV-10618 ROHS). Custom long stroke rolling diaphragm (LSRD) cylinders [37] were used in this setup, with a 4in stroke and a 1.75in effective diameter (average of the 1.95in bore and the 1.55in minimum piston diameter, between which the diaphragm connects), with food-grade mineral oil (SAE viscosity grade 20) as the hydraulic fluid. A pressurized fluid reservoir maintained at 30-40psi (pressure set via a manual regulator knob) was created by filling one side of a hydraulic cylinder (Bimba Original Line Stainless Steel) with the oil, and pressurizing the opposite side of the cylinder with compressed air. The cylinders were connected to two joints via pulleys of radius  $r = 1.5$ in as shown in Fig. 8a, resulting in the same transformation  $\mathbf{AJ}_1$  as Eq. (28), used in simulation. Polyurethane tubing (1/4in inner diameter, 3/8in outer diameter) was used for all hydraulic lines.

The operator interacted with the two joints via levers on each joint; each lever was equipped with a load cell for force sensing (Transducer Techniques LSP-5), and the pulleys were instrumented with encoders (US Digital E2-5000-315-IE-H-D-B). Data acquisition and control were performed with a Sensoray 626 card and Simulink Desktop Real-Time, with a sampling frequency of 1 kHz. Cylinder switching was limited to 20Hz due to solenoid response time. Forces were filtered in post-processing with a 5Hz first order zero-lag lowpass filter.

The current position and desired goal path were displayed to the operator in a Matlab plot window on a screen placed in front of the apparatus, scaled by the factor  $G = 2000/\pi$  to

fit the screen—this value resulted in the need for very small values of  $\gamma$ , since scaling distances that enter the blending calculation Eq. (16) are equivalent to using an effective blending parameter  $\gamma_e = G\gamma$  on the unscaled jointspace configuration.

2) *Taskspace Testing*: A taskspace transformation was introduced via a custom exoskeleton design, shown in Fig. 8b. The same modules and DAQ/control system were used as in the benchtop setup described above. One module was connected to the shoulder motion via a 4-bar linkage design, and the second module located below the elbow. The connection to the shoulder via a linkage was designed to minimize discomfort during any translation of the shoulder joint, since the human shoulder is not a simple rotational joint [38]; similar designs have been used in devices like the KINARM and MEDARM [39], [40]. We assumed for the calculations that the operator's shoulder remains fixed in place and that the arm does not slip in the cuffs (i.e., that the measured shoulder angle at the cylinder module is the same as the physical shoulder angle), such that the relationship

$$\mathbf{R} = \begin{bmatrix} l_1 \cos(q_1 + \beta) + l_2 \cos(q_1 + \beta + q_2) \\ l_1 \sin(q_1 + \beta) + l_2 \sin(q_1 + \beta + q_2) \end{bmatrix} \quad (32)$$

holds, where  $\beta = 42^\circ$  is a fixed offset between the angle measured by the encoder at the module and the shoulder angle. Note that this is the same as the transformation introduced by the linkage of Fig. 6a, with the simple addition of the joint offset  $\beta$  to  $q_1$ —we therefore expect the experimental results to match the simulation results for the 2D linkage (link lengths are the same).

Two 3-axis load cells (Galcoe GPB160, 50N range) were mounted beneath the cuffs for the upper and lower arm, and the forces were filtered in post-processing with a 5Hz first order zero-lag lowpass filter.

User interaction was provided by an individual familiar with the apparatus and interface, with an upper arm length of  $l_1 = 11.5$ " and a lower arm length of  $l_2 = 12$ " (measured to the center of the fist), and the device adjusted to match these lengths. The visual display showed both the current configuration of the arm and the desired path.

## V. RESULTS AND DISCUSSION

Results for each controller follow, arranged with the simulation and experimental results presented together for each of the three controllers, labeled in figures as *VF* (SDOF velocity field), *MVF* (force-modulated velocity field), and *CF* (constraint field). Simulation results are labeled *SIM*, and hardware results labeled *EXP*; the results for goal manifolds defined in jointspace are labeled *JOINT*, and those defined in taskspace are labeled *TASK*.

### A. SDOF Velocity Field Control

1) *2D Results*: Simulation and experimental results using a velocity field (*VF*) defined in jointspace to follow a straight line are shown in Fig. 9. Simulation results in Fig. 9a and Fig. 9b illustrate how the desired velocity field  $\mathbf{V}_{des}$  is approximated by the hardware with the SDOF headings that match most closely (gold), resulting in convergence along

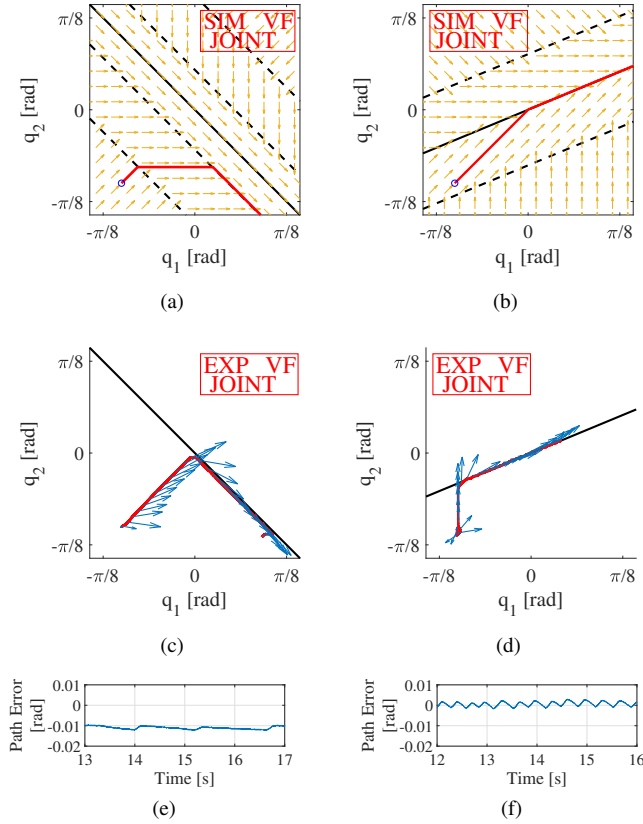


Fig. 9. SDOF velocity field control for convergence to straight lines through joint space. (a) and (b), simulation results for two different headings, generated with  $\gamma = 5$ . The discrete SDOF approximation of the continuous velocity field is shown in gold; distinct regions matching individual headings emerge, divided by the dashed black lines. The starting configuration is indicated by the blue circle, and the resulting path shown in red, converging towards the desired path (black) along the SDOF headings. The heading in (a) is one of the available directions  $T_{av}$ , while (b) shows an approximation of a line at an unavailable heading. (c-d) Experimental SDOF velocity field control ( $\gamma_e = 50.9$ ) in joint space imposing the same two desired paths. Desired constraint is drawn in black, and experimental trajectory in red, with applied joint torques overlaid as blue vectors. In (e) and (f), the tracking error over a period of time corresponding to steady-state tracking is shown for each experiment.

straight lines (red). As illustrated, distinct regions that match most closely to a certain available heading can be demarcated; the location of the boundaries between these regions is determined by the parameter  $\gamma$ . In steady-state, the behavior will depend on the desired path heading. Steady-state behavior will typically feature back-and-forth switching between two closest headings. If the path to be constrained is in the direction of one of the available device transmission ratios, there will be steady-state error with a magnitude determined by the selection of  $\gamma$ , since the region around the goal manifold will share this discrete direction, as can be seen in Fig. 9a.

The experimental SDOF velocity field control results for the same two lines are shown in Fig. 9c and Fig. 9d. Steady-state error profiles for these two experiments are shown in Fig. 9f and Fig. 9e. Compared to the simulation conditions, the blend from the approaching to the tracing field is much steeper with the selected value of  $\gamma$ , resulting in tighter tracking in Fig. 9c than Fig. 9a. As expected from simulation, Fig. 9e displays

steady-state error, since there is an SDOF region around the desired line that matches the desired heading; however, the chatter between two adjacent headings around a steady-state value is a result of fluid compliance, and was not observed in simulation.

Simulation results for tracking a circle in joint space are shown in Fig. 10a. Although the partitions are not drawn and are more complex in shape than for a line, it is clear that a similar division of the space into regions of uniform discrete heading have appeared in the discretized velocity field. It is worth noting that the performance will depend not only on the value of  $\gamma$ , but also on the path speed provided by the user, since switching occurs at a fixed time interval.

Experimental results for SDOF velocity field control for circular paths in joint space are shown in Fig. 10b and Fig. 10c—by controlling the blend between the approaching and tracing fields (via the parameter  $\gamma$ ), it is possible to control how tightly the desired path is followed. Good fit between the simulation and experimental results can be seen by comparing Fig. 10a with Fig. 10b, which both have the same  $\gamma$ . For baseline comparison, Fig. 10c shows experimental traces of both controlled and freehand drawing of a circle with the joint space apparatus. Freespace was implemented by setting  $\tilde{S}^T = [0 \ 0 \ 0 \ 0]$ , and the experienced operator used visual feedback only to trace the circle displayed on the screen without haptic guidance.

The SDOF velocity field method is robust to transformation between spaces as long as singularities are avoided. In Fig. 11a, the simulated goal path is defined in taskspace. The desired heading  $V_{des}$  is transformed into joint space ( $V_{q,des}$ ), where it is matched to the closest available heading from  $T_{q,av}$ ; this can result in abrupt turns in taskspace, as seen near the edge of the workspace in the top pane of Fig. 11a. A smoother path through taskspace would have resulted if  $T_{q,av}$  had instead been transformed to  $T_{av}$ , and the matching performed in taskspace. The approximation of the circle is limited by both the number of available SDOF settings and the selected value of  $\gamma$  for the simulation.

Fig. 11b shows experimental results gathered using the exoskeleton with the SDOF velocity field controller. Matching of the desired heading to the available headings was performed in taskspace. The controller imposes a series of SDOF curves that approximate the desired straight line.

2) *3D Results:* For 1DOF goal paths in higher dimensions, the SDOF velocity field method can be used just as in 2D. Simulation results for the imposition of a line through taskspace are shown in Fig. 11c, with  $V_{des}$  matched to the available headings  $T_{av} \in \{T_1, \dots, T_{j(1)}\}$  in taskspace.

### B. Force-Modulated SDOF Velocity Field Control

Force-modulated SDOF velocity field control (MVF) is applicable only for  $n > 2$ . Figure 12 shows results of simple 3D simulations of force-modulated SDOF velocity field control, using  $V_{tr}$  as defined in Eq. (22). The examples shown are for two different values of  $F_{\perp}$  applied throughout the simulation. The larger force causes a change in direction of rotation about the constrained sphere, while the smaller force



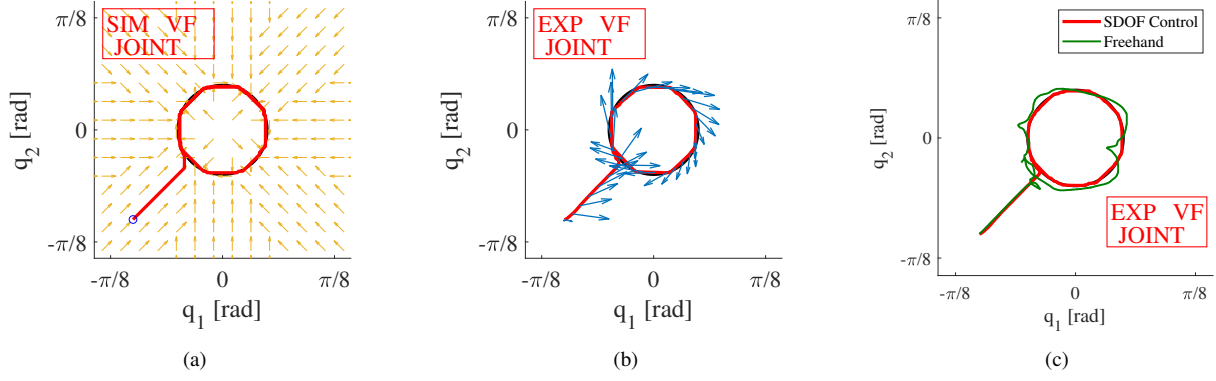


Fig. 10. SDOF velocity field control can also be used to trace other shapes - shown here is a circle. (a) Simulation results with the SDOF velocity field approximation shown in gold and the resulting path in red,  $\gamma = 50.9$ . (b) Experimental SDOF velocity field control in jointspace imposing a circle,  $\gamma_e = 50.9$ . (c) Experimental SDOF velocity field control in jointspace,  $\gamma_e = 127$  is shown in red. Note the contrast with Fig. 10b due to the different values of  $\gamma$ . For comparison, the controlled path is shown overlaid with freehand (green) shape tracing.

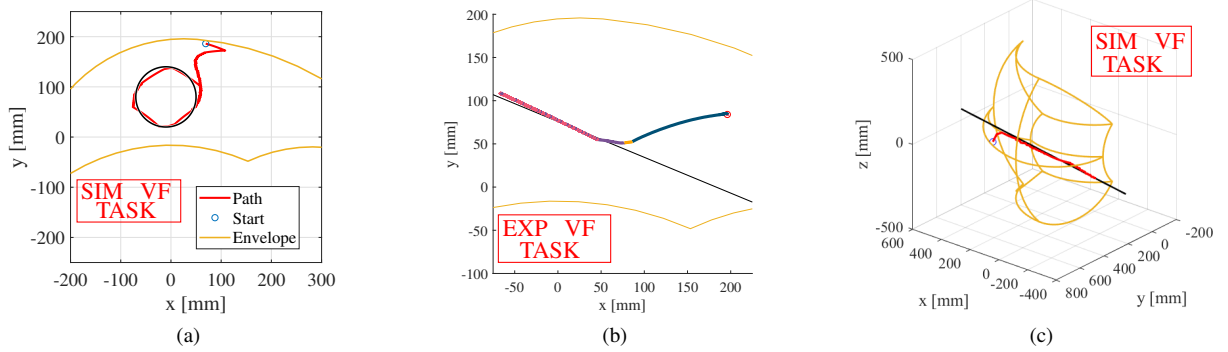


Fig. 11. SDOF velocity field control for goal paths in taskspace. (a) Simulation, with  $\gamma = 0.08$ . The desired vector field is defined in taskspace; the desired heading is transformed into jointspace, where the closest available heading is selected. (b) Experimental results ( $\gamma = 0.05$ ) on the exoskeleton. Colors correspond to the SDOF lines from Fig. 6b. (c) Simulation ( $\gamma = 0.04$ ) in 3D taskspace, with the desired line shown in black and the path taken shown in red. The work envelope is shown in gold for reference.

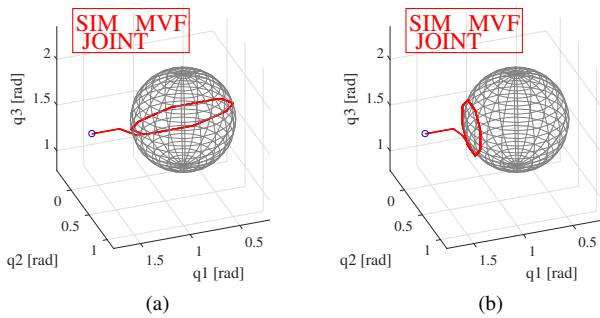


Fig. 12. Simulation results for force-modulated velocity field SDOF control,  $\gamma = 10$ ,  $\gamma_2 = 8$ , under two different user-applied levels of user-applied steering force: (a)  $F_{\perp} = 0.002$  and (b)  $F_{\perp} = 0.05$ . Units of applied force are irrelevant, since it acts in combination with the value  $\gamma_2$  in determining the weights for Eq. (22).

is not enough to alter the selected headings from the behavior with no force applied (not shown). Of course in a physical realization, varying steering forces could be applied at any time, and the controller would respond, changing direction any time the applied  $F_{\perp}$  was large enough to alter the largest projection onto  $V_{des}$ . Even in a more complex treatment of

applied force, for example where  $F_{\perp}$  is integrated over time to cause a response to small sustained steering forces, there cannot be immediate responsiveness to small applied forces due to the single physical DOF employed.

### C. Constraint Field Control

1) *2D Results:* Experimental results for imposing a line with the constraint field controller (CF) are shown in Fig. 13a. The equivalence of the SDOF velocity field and constraint field controllers in 2D can be seen in the similarity between this plot and Fig. 9c.

Match between the simulation and experimental results for the constraint field controller can be seen in Fig. 13b and 13c, where the controller is shown imposing circles in jointspace. As with the line, the constraint field controller performs identically to the SDOF velocity field controller under the same conditions—compare Fig. 13b with Fig. 10a.

Results for the constraint field controller in taskspace are shown in Fig. 14. Comparison shows a very good match in performance with simulation results (not pictured—see Table II) as well as significantly better performance than freehand. The freehand condition involved opening all valves to the manifold while the experienced operator used visual feedback

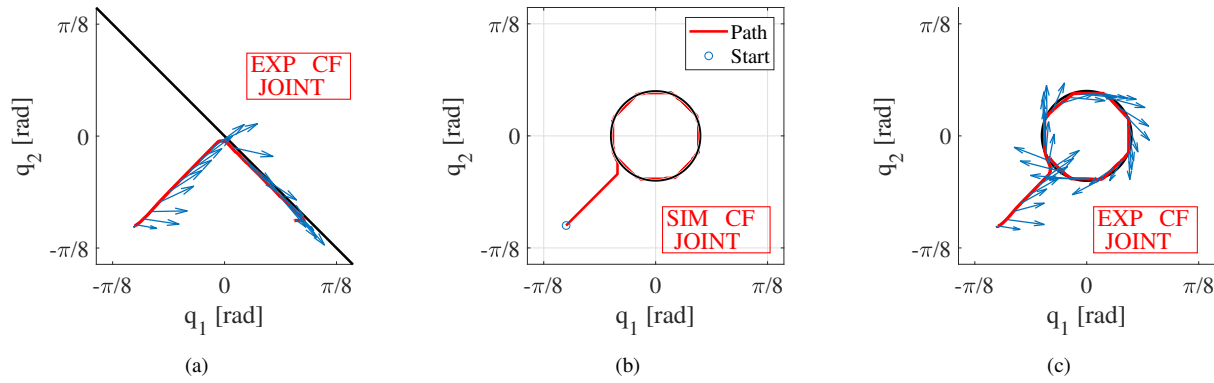


Fig. 13. Constraint field control in 2D jointspace: (a) experimentally imposing a line ( $\gamma_e = 50.9$ ), (b) simulating ( $\gamma = 50.9$ ) a circular path and (c) experimentally imposing the same circular path ( $\gamma_e = 50.9$ ).

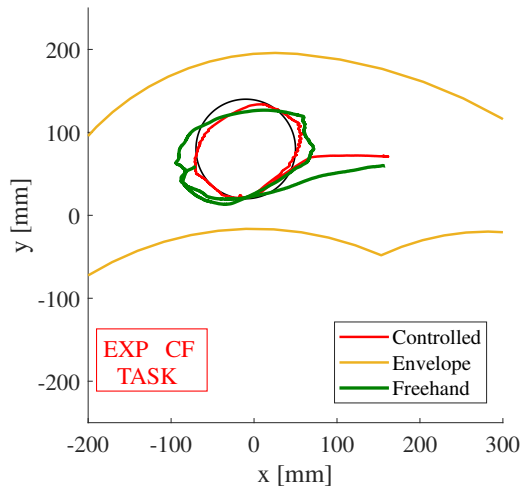


Fig. 14. Experimental exoskeleton circle tracing results with the constraint field controller,  $\gamma = 0.08$ . Constrained performance is shown overlayed with freehand performance.

only to trace the circle displayed on the screen without haptic guidance.

Looking at the experimental exoskeleton results presented both here in Fig. 14 for the constraint field controller and in Fig. 11b for SDOF velocity field control, the kinematic results were quite good—both controllers were able to impose upon the exoskeleton more or less the path predicted by simulation. However, looking at the interaction force and torque data measured by the load cells below the cuffs, we observed high loads (up to 10-12 Nm of torque applied about each joint, with forces above 50N measured in the axial direction under the cuffs) applied to achieve these motions. Thus, while the controller is able to overcome the parasitics in the device in the sense that the desired path is enforced, the nature of a passive device is such that there is no way to compensate for the losses due to damping. More detail on the recorded forces can be found in [41].

2) *3D Results*: Although motion would be allowable in any direction across the constrained planes in a physical realization, for the sake of simulation of the constraint field

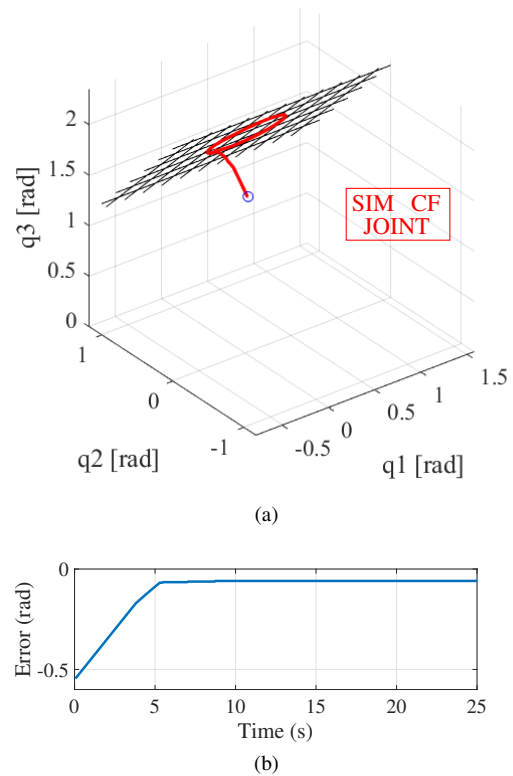


Fig. 15. (a) Constraint field controller simulation results ( $\gamma = 10$ ) for convergence to a plane through the point  $(0.5, 1, 1.3)$  with normal direction  $[0, 1, 1]^T$ . Planar curvature is generated with and a small steering force applied to simulate free motion across the constrained plane. (b) Error during the simulation.

controller, linear motion in the initial direction is acted on by simulated steering forces  $F_{tg}$  acting on a unit mass, and the heading is subsequently projected onto each plane at the transition to become  $T_{new}$ , as in Fig. 5c. Simulation results using the constraint field controller in 3D jointspace are shown in Fig. 15 and Fig. 16a and 16c. In Fig. 15, the goal manifold is a plane oriented in the direction of one of the available 2DOF manifolds enabled by the hardware—this results in very smooth motion across the surface, and a level of steady-state error. In Fig. 16a, the available planes are employed to

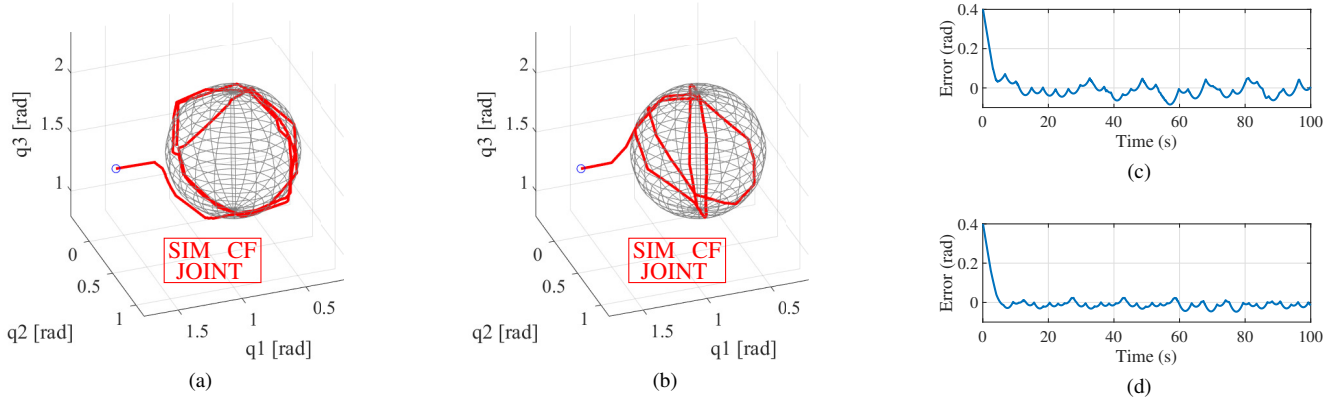


Fig. 16. Constraint field controller simulation ( $\gamma = 10$ ) imposing a sphere of radius 0.5 radians (a), and (c) path and error with 7 available planes. (b) and (d), the same controller with more available planes in the device by design ( $j_{(2)} = 21$  instead of 7), with all other settings the same. Both simulations generated with  $F_{\perp} = 0.002$ , assuming a device with point mass  $m = 1$ .

approximate a sphere.

The simulation of convergence to a sphere in Fig. 16a uses only the 7 available planes that were enabled by the manifold connections of (31), resulting in the relatively high error shown in (c). In a digital hydraulic device, there can be flexibility available by design in  $a$ , the maximum number of restricted DOF: a set of cylinders spanning three different joints may be interconnected via two completely separate fluid manifolds capable of restricting up to  $a = 2$  DOF, or connected via a single fluid manifold, which has  $a = 1$ . These two configurations might be used to maximize either  $j_{(1)}$  or  $j_{(2)}$ . Depending on the desired application, the designer might seek to maximize the available number of switch settings with different DOF. If the system had instead been designed to maximize  $j_{(2)}$  (the number of 2DOF planes available) by plumbing the cylinders into a single manifold instead of separating them into separate manifolds<sup>10</sup> to enable 1DOF constraints, tracking with lower error would be possible. Instead of the connections in (31), the same cylinder setup could be used with connections to a single manifold:

$$\tilde{\mathbf{S}}^T = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} \end{bmatrix}. \quad (33)$$

This enables the performance shown in Fig. 16 (b) and (d), with reduced error; however, there are *no SDOF constraints* possible in this configuration of  $\tilde{\mathbf{S}}$ , as a single manifold with  $a = 1$  can only remove  $m = 1$  DOF at a time.

Having the larger number of available 2DOF selections that result from the manifold configuration described by (33) also enables effectively enforcing taskspace constraints with the apparatus described in Fig. 7, as in convergence to a plane shown in Fig. 17. It is worth noting that, since the controller does not explicitly check for envelope limitations, we found our simulations to be highly susceptible to instability if initial conditions and the parameter  $\gamma$  are chosen to direct motion outside of the work envelope or near singularities.

<sup>10</sup>As an alternative to redesigning the system, connecting the two independent manifolds via an additional solenoid valve could allow flexibility in the dimension  $a$  of the constraint vector.

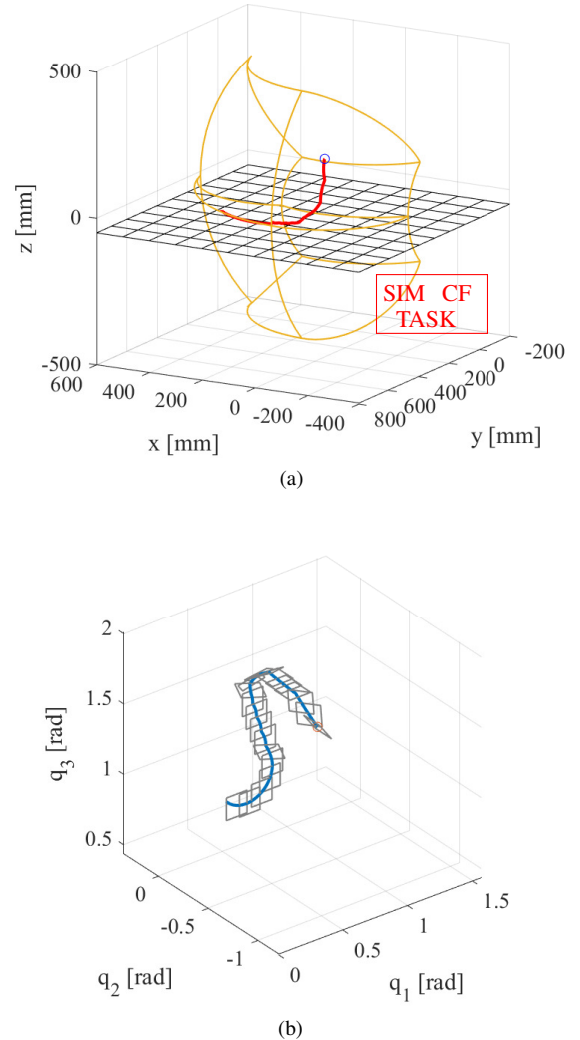


Fig. 17. (a) Constraint field controller convergence to a plane in taskspace (with jointspace relationship shown in Fig. 7), across which the operator can steer freely. Shown with  $\gamma = 0.02$ , converging to a plane with the surface normal  $[0 \ 0 \ 1]^T$  50mm above the shoulder. (b) The same path plotted through jointspace, with snapshots (shown at 0.75s intervals) of the imposed manifolds selected to steer the motion overlaid in grey.

Controller	Type	$\gamma$	RMSE	See
VF	SIM	50.9	0.0040 rad	Fig. 10a
VF	EXP	50.9	0.0059 rad	Fig. 10b
VF	EXP	127	0.0022 rad	Fig. 10c
CF	SIM	50.9	0.0040 rad	Fig. 13b
CF	EXP	50.9	0.0062 rad	Fig. 13c
Freehand	EXP	—	0.0191 rad	Fig. 10c

TABLE I  
RMS ERROR FOR CIRCLES CONSTRAINED IN JOINTSPACE.

Controller	Type	$\gamma$	Match Space	Fig. Ref.	RMSE
VF	SIM	0.08	JOINT	Fig. 11a	5.78 mm
CF	SIM	0.08	TASK	not pictured	6.44 mm
CF	EXP	0.08	TASK	Fig. 14	6.44 mm
Freehand	EXP	—	—	Fig. 14	15.81 mm

TABLE II  
RMS ERROR FOR CIRCLES CONSTRAINED IN TASKSPACE. “MATCH SPACE” REFERS TO THE SPACE IN WHICH THE CLOSEST AVAILABLE SETTING WAS SELECTED.

#### D. Comparative Performance

Direct comparison of the SDOF velocity field controller and the constraint field controller in 2D illustrates the equivalence of these controllers for imposing 1D constraint paths in a 2D space, as pointed out in the previous discussion. Numerically, this is shown in Tables I and II, which report the RMS error tracing a full revolution of a circle in each case (after convergence is achieved). Force-modulated constraint fields are not applicable to 2D spaces.

Comparison of Fig. 16a with Fig. 12a yields insight into the differences between using the two different extended control methods to impose  $n - 1$  dimensional goal manifolds—the simulations are generated with the same initial conditions, and the same applied values of  $F_{\perp}$ . As compared to the force-modulated SDOF velocity field controller, the constraint field method in 3D can provide a better sense of movement across a 2D surface, since instantaneous free motion is always available to the operator in two directions—by necessity, motion across the sphere’s surface in Fig. 12 was only possible when applied forces were large enough to force a change in the selected  $T_k$ , as was the case in Fig. 12b. The two methods naturally have different hardware requirements, as well, since they require different physical DOF. In 3D, no direct comparison is possible between the velocity field controller, which imposes 1D constraints, and the two extended methods, which both impose 2D constraints.

## VI. CONCLUSION

Vector field-based control methods are an attractive and practical solution to the control of discretely variable passive haptic devices since they specify a desired direction that can be matched to available motion constraints without requiring computationally expensive optimization. The closed-loop controllers were capable of compensating for some of the parasitic effects in the system, most notably the switching drift due to fluid compressibility that has been previously demonstrated in the characterization and open-loop control of digital hydraulic circuits in [42].

A drawback of controllers based on vector fields like those proposed here is that they can be susceptible to problems with limited workspace. In our introduction of velocity fields, we noted that modifications could be made to direct motion back into the allowable workspace near the edges; taking on implementation of this modification is important future work to make these controllers feasible for exoskeletons, particularly if they are to be used with patient populations whose range of motion is limited.

Many therapeutic motions (e.g., reaching, drawing circles, or exercising the edges of workspace) might be achieved through the imposition of goal manifolds in joint- or taskspace that can be encoded in velocity or constraint fields. Additional functionality can easily be achieved by taking advantage of the fact that velocity field-based methods can work with time-varying path definitions. For example, a center-out reaching task might be achieved by varying the heading of a constrained linear path every time a return to center is made. These simple modifications will enable further testing of self-powered rehabilitation paradigms with a variety of therapeutic or coordinated motions, building on preliminary findings about reduced slacking in [11].

A few challenges remain in the design and implementation of the exoskeleton apparatus before it is fully functional as a device suitable for testing with patient populations. First, while kinematic performance matches quite well with the simulation results, the damping losses need to be reduced to bring interaction forces down to a comfortable level and enable higher motion speeds—the forces were anecdotally high enough to both cause discomfort and confusion about the imposed direction of motion. These losses are primarily due to fluid damping in the hydraulic lines. In this work, we were limited by the materials used to make the rolling diaphragm cylinders to using mineral oil. Further exploration of alternative materials or cylinder designs (e.g., [43]) compatible with less viscous fluids like water is needed in order to address this limitation.

A second limitation of the experiments performed on the exoskeleton is the assumption that the operator’s shoulder remains fixed in the correct location to create a parallelogram-shaped 4-bar linkage from the shoulder to the cylinder module where the cylinders are mounted. Future iterations of the controller could make use of the additional encoders at the linkage joints to more accurately pinpoint the location of the shoulder with a more complex kinematic equation. After undertaking some simple exoskeleton design changes, the experiments to test the controller described in this work give us confidence to undertake future human subject tests to more rigorously test the controllers’ ability to function in more realistic scenarios and achieve therapeutic goals.

## ACKNOWLEDGMENT

The authors would like to thank Chin-Wei Lin and Tzu-Yuan (Justin) Lin for their work assembling the exoskeleton, Saeed Hashemi for providing the LSRD cylinders, and Necmiye Ozay for recommending several references in the hybrid control literature.

## REFERENCES

- [1] M. Peshkin and J. E. Colgate, "Cobots," *Industrial Robot: An International Journal*, vol. 26, no. 5, pp. 335–341, 1999.
- [2] E. L. Faulring, J. E. Colgate, and M. A. Peshkin, "The Cobot Hand Controller: Design, control and performance of a novel haptic display," *The International Journal of Robotics Research*, vol. 25, no. 11, pp. 1099–1119, 2006.
- [3] R. B. Gillespie, M. S. O'Modhrain, P. Tang, D. Zaretzky, and C. Pham, "The virtual teacher," in *Proceedings of the ASME Dynamic Systems and Control Division*. Anaheim: ASME, 1998, pp. 171–178.
- [4] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li, "Shared control in haptic systems for performance enhancement and training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 75–85, 2006.
- [5] N. Jarrasse, T. Proietti, V. Crocher, J. Robertson, A. Sahbani, G. Morel, and A. Roby-Brami, "Robotic exoskeletons: A perspective for the rehabilitation of arm coordination in stroke patients," *Frontiers in Human Neuroscience*, vol. 8, no. December, pp. 1–13, 2014.
- [6] L. Jabre, R. McGrew, R. B. Gillespie, and P. Goleski, "An assistive cobot for aid in self care activities," in *IFAC Conference on Mechatronic Systems*. Berkeley: Elsevier, 2002, pp. 1–6.
- [7] C. K. Chang, E. P. Washabaugh, A. Gwozdziowski, C. D. Remy, and C. Krishnan, "A semi-passive planar manipulandum for upper-extremity rehabilitation," *Annals of Biomedical Engineering*, vol. 46, no. 7, pp. 1047–1065, 2018.
- [8] J. L. Patton, M. E. Stoykov, M. Kovic, and F. A. Mussa-Ivaldi, "Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors," *Experimental Brain Research*, vol. 168, pp. 368–383, 2006.
- [9] D. J. Reinkensmeyer, O. M. Akoner, D. P. Ferris, and K. E. Gordon, "Slacking by the human motor system: Computational models and implications for robotic orthoses," in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine*. Minneapolis: IEEE, 2009, pp. 2129–2132.
- [10] A. Kaelin-Lang, L. Sawaki, and L. G. Cohen, "Role of voluntary drive in encoding an elementary motor memory," *Journal of Neurophysiology*, vol. 93, pp. 1099–1103, 2005.
- [11] E. P. Washabaugh, E. Treadway, R. B. Gillespie, C. David Remy, and C. Krishnan, "Self-powered robots to reduce motor slacking during upper-extremity rehabilitation: A proof of concept study," *Restorative Neurology and Neuroscience*, vol. 36, no. 6, pp. 693–708, 2018.
- [12] M. A. Peshkin, J. E. Colgate, W. Wannasupphorasit, C. A. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.
- [13] P. Bosscher and E. LaFay, "Haptic cobot exoskeleton: Concepts and mechanism design," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2006. Philadelphia: ASME, 2006, pp. 1–10.
- [14] E. Treadway, Z. Gan, C. D. Remy, and R. B. Gillespie, "Toward controllable hydraulic coupling of joints in a wearable robot," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 748–763, 2018.
- [15] D. Swanson and W. Book, "Obstacle avoidance methods for a passive haptic display," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Como: IEEE, 2001, pp. 1187–1192.
- [16] M. R. Reed and W. J. Book, "Modeling and control of an improved dissipative passive haptic display," in *IEEE International Conference on Robotics and Automation*. New Orleans: IEEE, 2004, pp. 311–318.
- [17] M. Sakaguchi, J. Furusho, and N. Takesue, "Passive force display using er brakes and its control experiments," in *Proceedings of the Virtual Reality 2001 Conference*. Yokohama: IEEE, 2001, pp. 7–12.
- [18] C. Cho, M. Kim, and J. B. Song, "Performance analysis of a 2-link haptic device with electric brakes," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS*. Los Angeles: IEEE, 2003, pp. 47–53.
- [19] D. Senkal, H. Gurocak, and E. I. Konukseven, "Passive haptic interface with mr-brakes for dental implant surgery," *Presence: Teleoperators and Virtual Environments*, vol. 20, no. 3, pp. 207–222, 2011.
- [20] P. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 751–763, 1999.
- [21] A. Erdogan, A. C. Satıcı, and V. Patoglu, "Passive velocity field control of a forearm-wrist rehabilitation robot," in *IEEE International Conference on Rehabilitation Robotics*. Zurich: IEEE, 2011, pp. 1–8.
- [22] J. Rascon-Enriquez, L. A. Garcia-Delgado, J. R. Noriega, and A. Garcia-Juarez, "Quad-rotor control using velocity field method," in *Congreso Nacional de Control Automatico*. San Luis Potosi: AMCA, 2018, pp. 400–405.
- [23] Z. Gan, K. Fry, R. B. Gillespie, and C. D. Remy, "A novel variable transmission with digital hydraulics," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*. Hamburg: IEEE, 2015, pp. 5838 – 5843.
- [24] S. C. Benga and R. A. Decarlo, "Optimal control of switching systems," *Automatica*, vol. 41, pp. 11–27, 2005.
- [25] D. Q. Mayne and S. Rakovic, "Model predictive control of constrained piecewise affine discrete-time systems," *Int. J. Robust Nonlinear Control*, vol. 279, no. August 2002, pp. 261–279, 2003.
- [26] P. Prabhakar and M. G. Soto, "Abstraction based model-checking of stability of hybrid systems," in *25th International Conference on Computer Aided Verification*. Berlin: Springer-Verlag, 2013, vol. 8044, pp. 280–295.
- [27] —, "An algorithmic approach to stability verification of polyhedral switched systems," in *Proceedings of the American Control Conference*. Portland: AACC, 2014, pp. 2318–2323.
- [28] —, "Counterexample guided abstraction refinement for stability analysis," in *Computer Aided Verification 2016*, S. Chaudhuri and A. Farzan, Eds. Springer International Publishing Switzerland, 2016, vol. LNCS9779, pp. 495–512.
- [29] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [30] D. K. Swanson and W. J. Book, "Path-following control for dissipative passive haptic displays," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Los Angeles: IEEE, 2003, pp. 101–108.
- [31] P. Y. Li and R. Horowitz, "Passive velocity field control (PVFC): Part i-geometry and robustness," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1346–1359, 2001.
- [32] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An assist-as-needed velocity field control scheme for rehabilitation robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Madrid: IEEE, 2018, pp. 3322–3327.
- [33] C.-Y. Chen and M.-Y. Cheng, "Velocity field control and adaptive virtual plant disturbance compensation for planar contour following tasks," *IET Control Theory & Applications*, vol. 6, no. 9, pp. 1182–1191, 2012.
- [34] A. Erdogan and V. Patoglu, "Online generation of velocity fields for passive contour following," in *2011 IEEE World Haptics Conference*. Istanbul: IEEE, 2011, pp. 245–250.
- [35] P. Y. Li and R. Horowitz, "Passive velocity field control (PVFC). part II. application to contour following," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, pp. 1360–1371, 2001.
- [36] C. A. Moore, M. A. Peshkin, and J. E. Colgate, "Cobot implementation of virtual paths and 3-D virtual surfaces," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 347–351, 2003.
- [37] S. Hashemi, S. Sobjinski, and W. K. Durfee, "Low-friction antagonist hydraulic transmission using long-stroke rolling diaphragm cylinders," in *Proceedings of the ASME/BATH 2017 Symposium on Fluid Power and Motion Control*. Sarasota: ASME, 2017, pp. 1–7.
- [38] A. Schiele, P. Letier, R. van der Linde, and F. van der Helm, "Bowden cable actuator for force-feedback exoskeletons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing: IEEE, 2006, pp. 3599–3604.
- [39] S. H. Scott, "Apparatus for measuring and perturbing shoulder and elbow joint positions and torques during reaching," *Journal of Neuroscience Methods*, vol. 89, no. 2, pp. 119–127, 1999.
- [40] S. J. Ball, I. E. Brown, and S. H. Scott, "MEDARM: A rehabilitation robot with 5DOF at the shoulder complex," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. Zurich: IEEE, 2007, pp. 1–6.
- [41] E. Treadway, "Perception, analysis, and control of parasitic effects in active and passive physical human-robot interaction," Ph.D. dissertation, University of Michigan, 2019.
- [42] Z. Gan, H. Tang, E. Treadway, R. B. Gillespie, and C. D. Remy, "Modelling and experimental evaluation of a variable hydraulic transmission," *IEEE Transactions on Mechatronics*, vol. Early Access, pp. 1–11, 2020.
- [43] A. Gruebele, S. Frishman, and M. R. Cutkosky, "Long-stroke rolling diaphragm actuators for haptic display of forces in teleoperation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1478–1484, 2019.



**Emma Treadway** received the B.S. degree in Engineering Science from Trinity University in 2011, and her M.S.E. and Ph.D. degrees in Mechanical Engineering from the University of Michigan, Ann Arbor in 2017 and 2019, respectively. She is an Assistant Professor of Mechanical Engineering in the Department of Engineering Science at Trinity University, San Antonio. Her research interests include human-robot interaction, rehabilitation robotics, prosthetics, and haptics.



**R. Brent Gillespie** received the BS degree in mechanical engineering from the University of California, Davis, in 1986, and the MS and PhD degrees in mechanical engineering from Stanford University, Stanford, CA, in 1992 and 1996, respectively. He is currently with the Department of Mechanical Engineering, University of Michigan, Ann Arbor. His current research interests include haptic interface and teleoperator control, human motor control, and robot-assisted rehabilitation after neurological injury.