

# Passive Compliance Control of Redundant Serial Manipulators

Jacob J. Rice<sup>1</sup>

Department of Mechanical Engineering,  
Marquette University,  
Milwaukee, WI 53233  
e-mail: jacob.rice@marquette.edu

Joseph M. Schimmels

Department of Mechanical Engineering,  
Marquette University,  
Milwaukee, WI 53233  
e-mail: joseph.schimmels@marquette.edu

*Passive compliance control is an approach for controlling the contact forces between a robotic manipulator and a stiff environment. This paper considers passive compliance control using redundant serial manipulators with real-time adjustable joint stiffness. Such manipulators can control the elastic behavior of the end-effector by adjusting the manipulator configuration and by adjusting the intrinsic joint stiffness. The end-effector's time-varying elastic behavior is a beneficial quality for constrained manipulation tasks such as opening doors, turning cranks, and assembling parts. The challenge in passive compliance control is finding suitable joint commands for producing the desired time-varying end-effector position and compliance (task manipulation plan). This problem is addressed by extending the redundant inverse kinematics (RIK) problem to include compliance. This paper presents an effective method for simultaneously attaining the desired end-effector position and end-effector elastic behavior by tracking a desired variation in both the position and the compliance. The set of suitable joint commands is not unique; the method resolves the redundancy by minimizing the actuator velocity norm. The method also compensates for joint deflection due to known external loads, e.g., gravity. [DOI: 10.1115/1.4039591]*

## 1 Introduction

Stiffness control, a subset of impedance control [1], is an interaction control strategy for managing contact forces in tasks where the end-effector motion is constrained, such as opening doors, turning cranks, and assembling parts. Stiffness is the impedance component that characterizes the force–displacement relationship. The interaction force due to displacement of the end-effector from an equilibrium position depends on the end-effector's stiffness. Conversely, the end-effector displacement from equilibrium due to an interaction force depends on the end-effector's compliance, the inverse of stiffness.

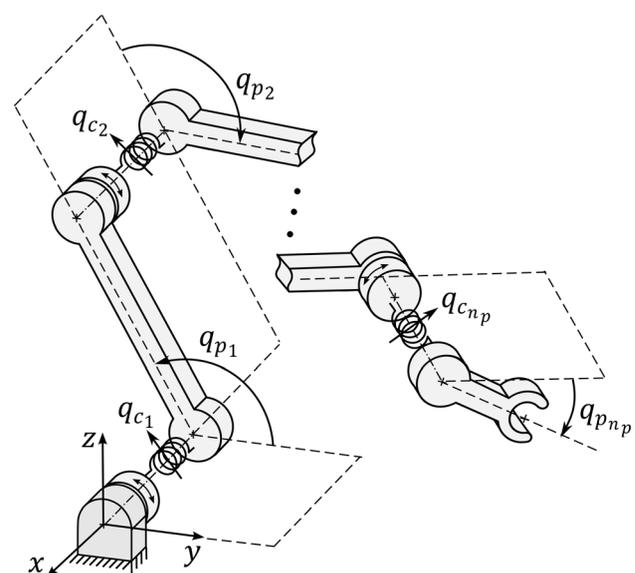
For constrained manipulation tasks, decreasing stiffness (increasing compliance) in the constrained direction(s) reduces the interaction force; increasing stiffness (decreasing compliance) in unconstrained directions reduces end-effector displacement due to known loads or to force disturbances.

Three methods of controlling task stiffness (impedance) [2] involve using: (1) active control with sensor feedback to impose a virtual joint stiffness matrix, (2) kinematic redundancy to adjust the joint configuration, which adjusts the task stiffness matrix, and (3) redundant actuation to adjust the intrinsic joint stiffness (a property of the human musculoskeletal system in con-

traction of antagonistic muscles). A large body of work [2–7] addresses active impedance control. Relatively little work addresses *passive stiffness control*, which is implemented using strategies 2 and 3 to adjust the task stiffness matrix without using sensor feedback. With passive stiffness control, each joint equilibrium position and joint stiffness are *independently* controlled. This contrasts with active stiffness control, where the primary actuators are controlled based on sensor feedback to emulate the desired stiffness at a commanded virtual equilibrium point. If it can implement the desired task stiffness, a passive stiffness controlled manipulator performs better [8] than a bandwidth-limited active controlled manipulator.

Passive stiffness control can be implemented using serial manipulators with elastic joints between consecutive links as illustrated in Fig. 1. A manipulator of this type has a diagonal joint stiffness matrix. If there is kinematic redundancy, the task stiffness may be controlled using strategy 2 even with nonadjustable joint stiffness. Control strategy 3 is possible if each joint is equipped with a variable stiffness actuator (VSA), an actuator that adjusts the intrinsic stiffness of the manipulator joint. A VSA has a real-time adjustable stiffness element in series with the primary actuator that controls the joint position (relative angle between links). Many VSAs [9–11] have the ability to adjust the joint stiffness without impacting the no-load equilibrium joint position. This paper presents a new method for controlling the end-effector position and end-effector stiffness of manipulators having VSAs of this type.

A limitation of passive stiffness control is the difficulty of its realization. As noted in Ref. [12], it is easy to implement an arbitrary task stiffness with a bandwidth-limited active control, but difficult, if not impossible, to implement the desired stiffness passively. The manipulator's ability to produce a desired end-effector passive stiffness strongly depends on its kinematic configuration. The set of implementable task stiffnesses is much larger for a redundant manipulator, where the manipulator can adjust its kinematic configuration without affecting the end-effector's position. Kinematic redundancy increases the capabilities of passive stiffness control, but complicates the problem of selecting the appropriate joint position and joint stiffness to realize the task manipulation plan. Prior work that addresses aspects of this problem is reviewed below.



**Fig. 1** Joint coordinates of an  $n_p$ -R serial manipulator with adjustable compliance. The kinematic position of each joint is given by  $q_{p_i}$  and the compliance of each joint is given by  $q_{c_i}$ .

<sup>1</sup>Corresponding author.

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the JOURNAL OF MECHANISMS AND ROBOTICS. Manuscript received July 25, 2017; final manuscript received January 31, 2018; published online May 31, 2018. Assoc. Editor: Hai-Jun Su.

**1.1 Prior Work in Passive Stiffness Control.** The problem of selecting the joint compliance values of planar 2R and 3R manipulators for realizing particle planar ( $E(2)$ ) task compliance matrices is addressed in Ref. [13]. Necessary and sufficient conditions for exactly realizing a  $2 \times 2$  compliance matrix are identified as well as joint compliance synthesis procedures to attain a desired end-point compliance. The results apply to any single specified kinematic configuration, not a manipulation plan (continuous sequence of positions and task compliances).

The problem of selecting the passive joint compliances for non-planar manipulators similar to the type illustrated in Fig. 1 is addressed in Refs. [14] and [15] using optimization. The optimization criterion minimizes the Frobenius norm of the difference between the desired task compliance and the passive end-effector compliance. The optimization in Ref. [14] selects the passive joint compliance without adjusting the kinematic configuration. The passive joint compliances attained by this optimization are not guaranteed to realize the desired task compliance. To compensate, the elastic behavior is supplemented with active stiffness control. In Ref. [15], a null space controller is used to adjust the kinematic configuration based on a local optimization for improving the task compliance. This approach adjusts the kinematic joint configuration without adjusting the passive joint compliance. These approaches [14,15] used for resolving the passive compliance and resolving the joint kinematics *separately* avoid using nonlinear constraints in the optimization, but sacrifice locally optimal joint control.

A method of simultaneous resolution of passive joint stiffnesses and joint kinematics is described in Ref. [16], where manipulation is treated as a series of configurations in static equilibrium. The method tracks the *change* in task stiffness with respect to time using a stiffness Jacobian matrix. Little attention, however, was given to deriving the stiffness Jacobian. In Ref. [16], the stiffness Jacobian is obtained from a function that maps joint stiffness to task stiffness. The stiffness mapping function used in Ref. [16] is the inverse of the mapping<sup>2</sup> commonly used in active stiffness control [17] for selecting the joint stiffness matrix. This choice of stiffness mapping, however, does not apply to redundant manipulators. For redundant manipulators, the joint stiffness for implementing the task stiffness is not unique and the inverse does not exist.

This paper presents means of selecting the joint position and passive joint stiffness for redundant serial manipulators implementing time-varying task manipulation plans. Unlike Refs. [14] and [15], our approach resolves the manipulator's joint position and joint stiffness *simultaneously*. Similar to the framework presented in Ref. [16], our approach tracks the required variation in the end-effector elastic behavior and position. Instead of using *stiffness*, however, we use a forward mapping of joint *compliance* to task *compliance*, which is computationally easier. More importantly, redundant manipulators can be considered in our passive compliance control approach because our compliance forward mapping does not require inverting the Jacobian matrix. As in Ref. [16], the manipulation plan is treated as a series of configurations in static equilibrium, but our approach compensates for joint deflection due to known external loads (not addressed in Ref. [16]).

**1.2 Overview.** This paper presents a new approach for generating actuation plans for redundant serial manipulators with VSAs to implement purely passive stiffness control. The approach is equivalent to solving the redundant inverse kinematics (RIK) problem extended to include compliance. In Sec. 2, the RIK problem is reviewed along with a standard local (differential) resolution approach using the weighted pseudoinverse. In Sec. 3, the inverse kinematics (IK) problem is generalized to

<sup>2</sup>This mapping selects a joint stiffness matrix, likely one with off-diagonal terms, that cannot be implemented passively with a manipulator like that in Fig. 1 even if a passive solution exists.

include compliance. In Sec. 4, local redundancy resolution (RR) of the compliance extended inverse kinematics problem is addressed by minimizing the norm of both the kinematic and compliance actuator motions. Compensation of joint deflection due to gravity is also addressed in Sec. 4. The manipulation planning approach for generating the actuator plan by numerically integrating the local RIK resolved joint motion is provided in Sec. 5. In Sec. 6, the manipulation planning approach is used to generate actuation commands for a 3R-VSA manipulator to perform a manipulation task in  $E(2)$  with time-varying task compliance. A brief summary of results and conclusions is provided in Sec. 7.

## 2 Review of Redundant Inverse Kinematics

Consider a noncompliant serial manipulator with  $n$  joint configuration coordinates where the end-effector performs a motion task described by  $m$  coordinates. The forward kinematic map  $f(\mathbf{q}) : Q \rightarrow X$  is a nonlinear function that defines the relationship between the manipulator's configuration in joint space  $\mathbf{q} \in Q$  (relative angles between consecutive links) and the end-effector configuration in task space  $\mathbf{x} \in X$  (position and orientation of the end-effector frame relative to the base frame).

The relationship between the joint velocity  $\dot{\mathbf{q}}$  and task velocity  $\dot{\mathbf{x}}$  is linear

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (1)$$

where  $\mathbf{J}(\mathbf{q}) = \partial f(\mathbf{q})/\partial \mathbf{q}$  is the manipulator's task Jacobian matrix evaluated at the local joint configuration  $\mathbf{q}$ .

Consider a task plan  $\mathbf{x}(t) \in X$ , a continuous set of desired task configurations corresponding to the desired motion of the manipulator's end-effector. The task plan is a path indexed by normalized time  $t \in [0, 1]$ . The IK path planning problem is to find a joint plan  $\mathbf{q}(t) \in Q$  such that  $f(\mathbf{q}(t)) = \mathbf{x}(t)$  for all time  $t$ , where  $\mathbf{q}(t)$  is a continuous set of joint configurations. For redundant manipulators ( $n > m$ ), there is an infinite number of solutions to the IK problem. Choosing a unique joint motion plan from the IK solution space is known as the RR problem. The RIK problem combines the IK and RR problems together.

The RIK problem is commonly solved locally at the velocity level [18] according to a quadratic optimization criterion, minimizing

$$G_{\text{local}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \quad (2)$$

where  $\mathbf{W}$  is an  $n \times n$  positive definite matrix of weighting values and the domain of the optimization is the set of joint velocities that produce the required task velocity given by linear constraints (i.e., Eq. (1)). The weighting matrix may be selected to minimize the norm of the joint velocities, minimize the kinetic energy of the manipulator [18], maximize the distance from joint limits [19], or some other criterion.

The unique joint motion that satisfies Eqs. (1) and (2) is easily identified using

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} \quad (3)$$

where  $\mathbf{J}^\dagger$  is the weighted pseudoinverse matrix of  $\mathbf{J}(\mathbf{q})$ , calculated by

$$\mathbf{J}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \quad (4)$$

when  $\mathbf{J}$  is full-rank ( $\text{rank}(\mathbf{J}) = m$ ). The weighted pseudoinverse simultaneously addresses the IK problem with  $\mathbf{J}$  and addresses the RR problem with  $\mathbf{W}$ .

The joint plan  $\mathbf{q}(t)$  is generated by integrating the local RIK solution

$$\mathbf{q}(t) = \mathbf{q}_0 + \int_{t=0}^{t=1} \mathbf{J}^\dagger(\mathbf{q}(t)) \dot{\mathbf{x}}(t) dt \quad (5)$$

where  $\mathbf{q}_0 \in Q \mid f(\mathbf{q}_0) = \mathbf{x}(0)$  is an admissible initial joint configuration.

By augmenting the joint configuration coordinates and task configuration coordinates to include compliance (inverse of stiffness), the weighted pseudoinverse method is extended below to track both a desired equilibrium task position and a desired task compliance.

### 3 Compliance Extended Inverse Kinematics

To include compliance in the local IK resolution approach, the joint space  $Q$  and task space  $X$  are augmented to include joint compliance coordinates and task compliance coordinates, respectively. The mapping function  $\mathbf{x} = f(\mathbf{q})$  from the joint space to the task space and the tangent space mapping  $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$  are augmented with additional mappings for compliance. This section provides the details. To emphasize the distinction between the typical kinematic and the augmented compliance coordinates and mappings, subscript  $(\cdot)_p$  indicates kinematic coordinates or mappings and subscript  $(\cdot)_c$  indicates compliance coordinates or mappings.

**3.1 Compliance Augmented Joint Space.** The class of manipulators considered in this paper are *serial* manipulators with  $n_p$  revolute joints where each joint has adjustable compliance (as depicted in Fig. 1). The joint kinematic configuration is  $\mathbf{q}_p = [q_{p1}, q_{p2}, \dots, q_{pn_p}]^T$  where the coordinate  $q_{pi} \in (-\infty, \infty)$  is the relative angle between link  $i$  and link  $(i-1)$  (link 0 is the ground). The joint kinematic configuration  $\mathbf{q}_p$  is the static equilibrium configuration of the manipulator when no disturbance acts on the manipulator (i.e., no commanded position error due to constrained interaction).

The passive compliance of each joint is independently controlled; therefore, the joint compliance configuration is  $\mathbf{q}_c = [q_{c1}, q_{c2}, \dots, q_{cn_p}]^T$  where the coordinate  $q_{ci} \in (-\infty, \infty)$  is the compliance of joint  $i$ . Note, the joint compliance must have a positive value to be realized; this issue is treated the same as joint limits (discussed in Sec. 5). The joint compliance matrix  $\mathbf{C}_q$  is an  $n_p \times n_p$  diagonal matrix with  $\mathbf{q}_c$  as the diagonal elements.

Since the joint kinematic configuration and the joint compliance configuration are controlled independently, the total joint space configuration  $\mathbf{q} \in Q$  of the manipulator is formed by concatenating the joint kinematic and joint compliance configurations

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_p \\ \mathbf{q}_c \end{bmatrix} = \begin{bmatrix} [q_{p1}, q_{p2}, \dots, q_{pn_p}]^T \\ [q_{c1}, q_{c2}, \dots, q_{cn_p}]^T \end{bmatrix}$$

**3.2 Compliance Augmented Task Space.** The task manipulation plan for compliance control involves both kinematics and compliance. The robot end-effector has a specific position/orientation given by the  $m_p \times 1$  vector  $\mathbf{x}_p$ , where  $m_p$  is the number of task kinematic degrees-of-freedom. The end-effector has a specific elastic behavior given by  $\mathbf{C}_x$ , an  $m_p \times m_p$  positive definite task compliance matrix.

Since  $\mathbf{C}_x$  is symmetric, the task compliance is fully and uniquely described by the upper triangular elements. Therefore, these elements are selected as the task compliance coordinates of the task compliance configuration  $\mathbf{x}_c = \text{sort}^\Delta(\mathbf{C}_x)$  where  $\text{sort}^\Delta(\cdot)$  is a sorting operation for arranging the upper triangular elements of  $\mathbf{C}_x$  into an  $m_c \times 1$  vector based on element position in the

matrix. The number of task compliance coordinates is  $m_c = (1/2)m_p(m_p + 1)$ .

The augmented task configuration  $\mathbf{x} \in X$  is formed by concatenating the task kinematic and task compliance configurations  $\mathbf{x} = [\mathbf{x}_p^T, \mathbf{x}_c^T]^T$ .

**3.3 Compliance Augmented Forward Maps.** The augmented joint space coordinates  $\mathbf{q}$  and augmented task space coordinates  $\mathbf{x}$  are each a concatenation of kinematic and compliance coordinates. Because of this structure, the forward mapping function  $f(\mathbf{q})$  is a concatenation of the forward kinematics and forward compliance functions

$$f(\mathbf{q}) = \begin{bmatrix} f_p(\mathbf{q}) \\ f_c(\mathbf{q}) \end{bmatrix}$$

where the forward kinematic function  $f_p$  gives the task kinematic configuration  $\mathbf{x}_p$  and the forward compliance function  $f_c$  gives the task compliance configuration,  $\mathbf{x}_c$ .

In our compliance resolution approach, compliance and kinematics are resolved simultaneously at the local (differential) level. Rather than finding the appropriate joint compliance configuration for each task instance, we find the best infinitesimal change in the joint configuration (kinematic and compliance) to produce the desired infinitesimal change in the task configuration (kinematic and compliance). The differential task requirements are a set of linear constraints given by the task Jacobian matrix  $\mathbf{J} = \partial f(\mathbf{q})/\partial \mathbf{q}$ .

Because the joint space and task space coordinates are concatenations of kinematic and compliance coordinates, the compliance augmented Jacobian may be partitioned as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_p(\mathbf{q})}{\partial \mathbf{q}_p} & \frac{\partial f_p(\mathbf{q})}{\partial \mathbf{q}_c} \\ \frac{\partial f_c(\mathbf{q})}{\partial \mathbf{q}_p} & \frac{\partial f_c(\mathbf{q})}{\partial \mathbf{q}_c} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{pp} & \mathbf{J}_{pc} \\ \mathbf{J}_{cp} & \mathbf{J}_{cc} \end{bmatrix} \quad (6)$$

where  $\mathbf{J}_c = [\mathbf{J}_{cp}, \mathbf{J}_{cc}]$  is the compliance Jacobian. The partition  $\mathbf{J}_{pp} = \partial f_p(\mathbf{q})/\partial \mathbf{q}_p$  is the standard kinematic Jacobian used in IK literature. The partition  $\mathbf{J}_{pc} = \partial f_p(\mathbf{q})/\partial \mathbf{q}_c$  describes how the end-effector task position changes with changes in the joint compliance variables. Since robot kinematics do not depend on the joint compliance variables,  $\mathbf{J}_{pc} = \mathbf{0}$ . The partition  $\mathbf{J}_{cp} = \partial f_c(\mathbf{q})/\partial \mathbf{q}_p$  describes how the end-effector task compliance changes with changes in the joint kinematic variables. The partition  $\mathbf{J}_{cc} = \partial f_c(\mathbf{q})/\partial \mathbf{q}_c$  describes how the end-effector task compliance changes with changes in the joint compliance variables.

**3.4 Constructing the Compliance Jacobian.** In this section, the compliance forward map and the compliance Jacobian are constructed using the kinematic Jacobian and the joint compliance matrix.

Serial manipulators with independently controlled compliant joints have a simple mapping from the manipulator's joint compliance matrix to its end-effector task compliance matrix. The end effector's task compliance is the superposition of the task compliance contribution from each joint and is given by [20]

$$\mathbf{C}_x(\mathbf{q}) = \mathbf{J}_{pp} \mathbf{C}_q \mathbf{J}_{pp}^T \quad (7)$$

where  $\mathbf{J}_{pp}$  is the kinematic Jacobian and  $\mathbf{C}_q$  is the diagonal joint compliance matrix with  $\mathbf{q}_c$  as the diagonal elements. The end-effector's task compliance matrix  $\mathbf{C}_x(\mathbf{q})$  is symmetric and positive definite. Note that Eq. (7) does not require calculating the inverse Jacobian matrix (a step needed in the mapping function described in Ref. [16]).

The forward compliance function as a *configuration* mapping is readily constructed by sorting the upper triangular elements of  $\mathbf{C}_x$  into a vector

$$f_c(\mathbf{q}) = \text{sort}^\Delta \left( \mathbf{J}_{pp} \mathbf{C}_q \mathbf{J}_{pp}^T \right) \quad (8)$$

The compliance Jacobian is  $\mathbf{J}_c = \partial f_c(\mathbf{q}) / \partial \mathbf{q}$  and the variation in task compliance configuration is  $\delta \mathbf{x}_c = \mathbf{J}_c \delta \mathbf{q}$ , where  $\delta \mathbf{q}$  is a variation in the augmented joint configuration. The compliance Jacobian, composed as a set of column vectors, is

$$\mathbf{J}_c = \left[ \frac{\partial f_c(\mathbf{q})}{\partial q_1}, \frac{\partial f_c(\mathbf{q})}{\partial q_2}, \dots, \frac{\partial f_c(\mathbf{q})}{\partial q_n} \right] = \sum_{k=1}^n \frac{\partial f_c(\mathbf{q})}{\partial q_k} \quad (9)$$

where  $\partial f_c(\mathbf{q}) / \partial q_k$  is the  $k$ th column of  $\mathbf{J}_c$  and  $q_k$  is the  $k$ th joint space variable.

Consider the partial derivative of the task compliance function with respect to the  $k$ th joint variable

$$\frac{\partial f_c(\mathbf{q})}{\partial q_k} = \frac{\partial}{\partial q_k} \text{sort}^\Delta \left( \mathbf{J}_{pp} \mathbf{C}_q \mathbf{J}_{pp}^T \right) \quad (10)$$

Both the sorting operation and the partial derivative of a matrix with respect to a single variable are element-by-element operations. Therefore, sorting may occur after taking the partial derivative. Substituting Eq. (7) into Eq. (10) yields

$$\frac{\partial f_c(\mathbf{q})}{\partial q_k} = \text{sort}^\Delta \left( \frac{\partial \mathbf{C}_x(\mathbf{q})}{\partial q_k} \right) \quad (11)$$

Using the product rule, the partial derivative of Eq. (7) expands to<sup>3</sup>

$$\frac{\partial \mathbf{C}_x(\mathbf{q})}{\partial q_k} = \begin{cases} \left( \frac{\partial \mathbf{J}_{pp}}{\partial q_k} \right) \mathbf{C}_q \mathbf{J}_{pp}^T + \mathbf{J}_{pp} \mathbf{C}_q \left( \frac{\partial \mathbf{J}_{pp}}{\partial q_k} \right)^T & 1 \leq k \leq n_p \\ \mathbf{J}_{pp} \left( \frac{\partial \mathbf{C}_q}{\partial q_k} \right) \mathbf{J}_{pp}^T & n_p + 1 \leq k \leq n \end{cases} \quad (12)$$

where  $q_k$  is a kinematic joint variable for  $1 \leq k \leq n_p$  and a compliance joint variable for  $n_p + 1 \leq k \leq n$ . This decomposition is similar to that used in the conservative congruence transform [21] for finite deflection of a mechanism stiffness. This decomposition, however, is on compliance and it includes an additional term that accounts for variation in the joint compliance.

The columns of the compliance Jacobian are constructed using Eq. (12) after generating  $\partial \mathbf{J}_{pp} / \partial q_k$  and  $\partial \mathbf{C}_q / \partial q_k$ , which are evaluated element by element.

The partial derivatives of the joint compliance matrix components have a simple form given by

$$\left( \frac{\partial \mathbf{C}_q}{\partial q_k} \right)_{ij} = \begin{cases} 1 & i = j = k - n_p \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

With these results, the compliance augmented Jacobian (total Jacobian)  $\mathbf{J}$  is readily calculated and used for locally resolving the compliance extended inverse kinematics problem using the weighted pseudoinverse method.

<sup>3</sup>Equation (12) may be separated into cases because  $\mathbf{C}_q$  is independent of the kinematic variables  $\mathbf{q}_p$ , and  $\mathbf{J}_{pp}$  is independent of the compliance variables  $\mathbf{q}_c$ .

## 4 Compliance Extended Redundancy Resolution

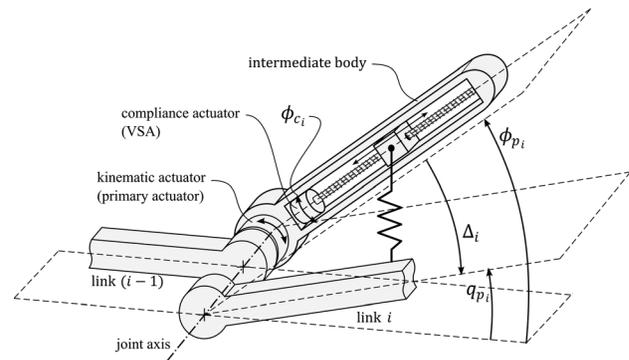
When a solution to the compliance extended local IK problem exists, the solution is not unique for redundant manipulators. There are many kinematic and compliance *joint motions* that produce the desired kinematic and compliance *task motion*. Here, the redundancy is resolved by minimizing the velocity norm of the actuators used to obtain the desired kinematic and compliance behavior.

For traditional industrial manipulators with rigid joints, the joint coordinates (relative angle between consecutive links) and the actuator coordinates (primary actuator positions) are the same. This is not that case for manipulators with elastic joints; actuator coordinates form a separate space  $\Phi$  from the joint space  $\mathcal{Q}$ . The actuator coordinates of serial manipulators with VSAs are separate from its joint coordinates for two reasons: (1) externally applied loads displace the joint position from the no-load equilibrium position commanded by the primary actuators, and (2) the VSA actuator position and the joint compliance have different units and therefore separate configuration spaces. The differences between the actuator coordinates and joint coordinates for both kinematic and compliance are illustrated in Fig. 2.

In this section, a mapping function from joint space to actuator space for both kinematic and compliance variables is described. A linear mapping from the joint tangent space to the actuator tangent space (the actuator Jacobian) is also described. This actuator Jacobian is used in the pseudoinverse weighting matrix for minimizing the actuator motion.

**4.1 Actuator Coordinates.** As stated previously and illustrated in Fig. 1, manipulators having variable stiffness actuators [9–11] allow kinematic joint position and joint compliance to be independently controlled. Each joint of the manipulator can be modeled as having two revolute actuators, a kinematic actuator (primary actuator) for adjusting joint kinematic position, and a compliance actuator for adjusting joint compliance.

An example of a joint with separate kinematic and compliance actuators is illustrated in Fig. 2. Each link  $i$  is connected to link  $(i - 1)$  by a revolute joint. The kinematic actuator, attached to link  $(i - 1)$ , drives an intermediate body that is elastically coupled to link  $i$ . In Fig. 2, the spring illustrated is a simple elastic coupling between two points. The compliance actuator illustrated in Fig. 2 adjusts the effective joint compliance by controlling the orientation of the elastic element using a motor and ball screw. For each joint  $i$ ,  $\phi_{p_i}$  is the kinematic actuator position and  $\phi_{c_i}$  is the compliance actuator position. The total actuator configuration of the manipulator is



**Fig. 2 Actuator coordinates of joint  $i$ .** The kinematic actuator position  $\phi_{p_i}$  is the no-load joint position. External loads cause a joint displacement  $\Delta_i$  resulting in the static equilibrium joint position  $q_{p_i}$ . The amount of displacement depends on the joint compliance  $\phi_{c_i}$  that is determined by the compliance actuator position  $\phi_{c_i}$ .

$$\phi = \begin{bmatrix} \phi_p \\ \phi_c \end{bmatrix} = \begin{bmatrix} [\phi_{p_1}, \phi_{p_2}, \dots, \phi_{p_{n_p}}]^T \\ [\phi_{c_1}, \phi_{c_2}, \dots, \phi_{c_{n_p}}]^T \end{bmatrix}$$

There is a nonlinear function  $\phi = \mathbf{h}(\mathbf{q})$  that maps a *joint* configuration to an *actuator* configuration. Since the compliance actuator coordinates are separate from the kinematic coordinates, the actuator mapping function is partitioned as

$$\mathbf{h}(\mathbf{q}) = \begin{bmatrix} \mathbf{h}_p(\mathbf{q}) \\ \mathbf{h}_c(\mathbf{q}) \end{bmatrix}$$

The kinematic actuator mapping  $\mathbf{h}_p$  defines the relationship between the joint configuration  $\mathbf{q}$  and the kinematic actuator configuration  $\phi_p$ . The compliance actuator mapping  $\mathbf{h}_c$  defines the relationship between the joint configuration  $\mathbf{q}$  and the compliance actuator position  $\phi_c$ . These mappings are described in detail below.

**4.2 Kinematic Actuator Mapping.** In the absence of external loads, kinematic joint positions and kinematic actuator positions are the same. However, gravitational loads are normally present and may cause significant joint deflection from the commanded position  $\mathbf{q}_p - \phi_p \neq \mathbf{0}$ . Commanded kinematic joint positions may be compensated for known loads (e.g., gravity) such that the deflected joint position is the desired position.

When joint compliance is independent of joint deflection, the kinematic actuator command for gravity compensation is [4].

$$\mathbf{h}_p(\mathbf{q}) = \mathbf{q}_p + \mathbf{C}_q \mathbf{g}(\mathbf{q}_p) \quad (14)$$

where  $\mathbf{g}(\mathbf{q}_p)$  is a vector of applied torques at the manipulator joints. For gravitational loads, the torque at each joint from the load is a function of the joint configuration determined by the link mass and moment arm. Since  $\mathbf{C}_q = \text{diag}(\mathbf{q}_c)$ , both  $\mathbf{q}_c$  and  $\mathbf{q}_p$  are used in the kinematic actuator mapping function.

**4.3 Compliance Actuator Mapping.** The torque experienced at each VSA may be described by  $\tau_{\text{VSA}_i}(\phi_{c_i}, \Delta_i)$ , a nonlinear function of the VSA actuator coordinate  $\phi_{c_i}$ , and the link deflection  $\Delta_i = \mathbf{q}_{p_i} - \phi_{p_i}$ . The shape of the function is determined by the VSA design.

If the torque–deflection relationship is modeled as linear such that the stiffness does not change significantly with link deflection  $\Delta_i$ , then the joint stiffness can be described by the compliance actuator position alone:  $k_i(\phi_{c_i})$ .

For VSA designs in which kinematic and compliance properties are independently controlled, when the VSA has a strictly increasing (or decreasing) compliance profile:  $q_{c_i} = 1/k_i(\phi_{c_i})$ , there exists an inverse function  $h_{c_i}$  such that  $\phi_{c_i} = h_{c_i}(q_{c_i})$ .

An exponential [11] compliance versus actuation profile allows a VSA to attain a very large range of elastic behaviors with only a small change in actuator position. The joint compliance is given by  $q_{c_i} = c_0 \exp(\xi \phi_{c_i})$ , where  $\xi$  and  $c_0$  are constants. The compliance actuator position, then, is given by

$$h_{c_i}(q_{c_i}) = \frac{1}{\xi} \ln\left(\frac{q_{c_i}}{c_0}\right) \quad (15)$$

Since each VSA is independently controlled, the compliance actuator mapping function is

$$\mathbf{h}_c(\mathbf{q}) = [h_{c_1}(q_{c_1}), h_{c_2}(q_{c_2}), \dots, h_{c_{n_p}}(q_{c_{n_p}})]^T \quad (16)$$

Assuming the mapping function  $\mathbf{h}(\mathbf{q})$  is continuous and differentiable, the linear mapping from the joint tangent space to the actuator tangent space is given by

$$\dot{\phi} = \mathbf{J}_\phi \dot{\mathbf{q}} \quad (17)$$

where  $\mathbf{J}_\phi = \partial \mathbf{h}(\mathbf{q}) / \partial \mathbf{q}$  is the actuator Jacobian matrix. The actuator Jacobian can be evaluated column by column, taking the partial derivative of the actuator mapping function with respect to each joint variable. If the actuator kinematic and compliance variables are independent, the actuator Jacobian is partitioned as

$$\mathbf{J}_\phi = \begin{bmatrix} \frac{\partial \mathbf{h}_p(\mathbf{q})}{\partial \mathbf{q}_p} & \frac{\partial \mathbf{h}_p(\mathbf{q})}{\partial \mathbf{q}_c} \\ \frac{\partial \mathbf{h}_c(\mathbf{q})}{\partial \mathbf{q}_p} & \frac{\partial \mathbf{h}_c(\mathbf{q})}{\partial \mathbf{q}_c} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\phi_{pp}} & \mathbf{J}_{\phi_{pc}} \\ \mathbf{J}_{\phi_{cp}} & \mathbf{J}_{\phi_{cc}} \end{bmatrix} \quad (18)$$

Note that, for the VSAs considered,  $\mathbf{h}_c$  only uses *compliance* joint variable as inputs. As such, the partition  $\mathbf{J}_{\phi_{cp}} = \mathbf{0}$ . This contrasts with the task Jacobian where the diagonally opposite component  $\mathbf{J}_{pc} = \mathbf{0}$ .

The redundancy resolution criterion used to minimize the norm of the actuator velocities is equivalent to minimizing the quadratic objective function  $G_{\text{local}} = \frac{1}{2} \dot{\phi}^T \dot{\phi}$ .

Substituting Eq. (17) into this expression yields

$$G_{\text{local}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{J}_\phi^T \mathbf{J}_\phi \dot{\mathbf{q}} \quad (19)$$

Note that Eq. (19) and Eq. (2) have the same form. As such, the pseudoinverse weighting matrix is selected to be

$$\mathbf{W}(\mathbf{q}) = \mathbf{J}_\phi^T \mathbf{J}_\phi \quad (20)$$

This selection of the weighting matrix minimizes the norm of the *actuator* velocity, while resolving the inverse kinematics in *joint* tangent space.

The compliance extended weighted pseudoinverse is calculated by substituting Eqs. (20) and (6) into Eq. (4). This weighted pseudoinverse is used to solve the *local* compliance extended RIK problem. The procedure for integrating the local RIK solution for the whole ( $t=0 \rightarrow 1$ ) task manipulation plan is provided below.

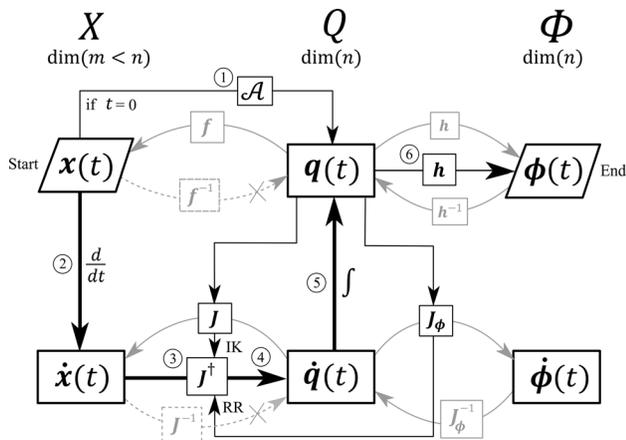
## 5 Manipulation Planning Approach

Passive compliance control of redundant serial manipulators is implemented by executing an actuator plan containing the kinematic and compliance actuator positions for producing the desired task manipulation plan. In this section, a basic numerical procedure for generating the actuation plan is provided.

A summary of the weighted pseudoinverse RIK resolution approach is depicted in Fig. 3. Three configuration spaces are relevant: (1) task space  $X$ , (2) joint space  $Q$ , and (3) actuator space  $\Phi$ . The desired task manipulation plan  $\mathbf{x}(t) \in X$ , including both position and compliance, is provided as input to the procedure. The actuator plan  $\phi(t) \in \Phi$ , needed for implementing passive stiffness control, is the output of the procedure.

For redundant manipulators considered here, a direct mapping  $f^{-1}: X \rightarrow Q$  from the task space to joint space does not exist;  $\mathbf{J}^{-1}$  does not exist either. However, an admissible joint motion that produces the desired task motion is identified with the weighted pseudoinverse  $\mathbf{J}^\dagger$ , which resolves redundancy by minimizing the actuator velocity norm. Joint space  $Q$  serves as an intermediary between task space and actuator space in which the IK and RR problems are resolved at the differential level using the weighted pseudoinverse.

The manipulation planning approach uses the following steps:



**Fig. 3** Local IK resolution of redundant manipulators with elastic joints using the weighted pseudoinverse  $\mathbf{J}^\dagger$  at a single instant in time. User-provided input is  $\mathbf{x}(t)$  and procedure output is  $\boldsymbol{\phi}(t)$ . Thick straight arrows show the integration cycle to generate the joint plan  $\mathbf{q}(t)$ . Solid straight arrows show necessary calculations. Solid curved arrows represent mappings between spaces, whereas dashed curved arrows with a  $\times$  indicate non-existent mappings between spaces.

- (1) Determine an admissible joint configuration  $\mathbf{q}_0 \mid f(\mathbf{q}_0) = \mathbf{x}(0)$  using an analytical process [13] or search algorithm (denoted by  $\mathcal{A}$  in Fig. 3).
- (2) Differentiate the task manipulation plan with respect to normalized time to identify the desired task motion  $\dot{\mathbf{x}}(t)$ .
- (3) Evaluate the compliance extended Jacobian  $\mathbf{J}$  and the weighting matrix  $\mathbf{W} = \mathbf{J}_\phi^T \mathbf{J}_\phi$  at the current joint configuration. Evaluate the weighted pseudoinverse matrix  $\mathbf{J}^\dagger$  using Eq. (4).
- (4) Identify the local optimal joint motion  $\dot{\mathbf{q}}(t)$  using Eq. (3).
- (5) Numerically integrate the joint motion using a small step size in  $t$ .
- (6) Determine the actuator plan using the actuator mapping:  $\boldsymbol{\phi}(t) = \mathbf{h}(\mathbf{q}(t))$ .

The integration loop (steps 2–5) repeats, drawing information from the task motion plan  $\mathbf{x}(t)$  each cycle until  $t = 1$ . Steps 2–5 describe the execution of Eq. (5) to generate the joint manipulation plan.

The IK and RR problems are resolved in joint space rather than actuator space because  $\mathbf{h}^{-1}: \Phi \rightarrow Q$  cannot be computed directly when gravitational loads are present; it may be approximated, however, by iterative calculation [4]. Evaluating the Jacobian  $\mathbf{J}_\phi^{-1} = \partial \mathbf{h}^{-1}(\boldsymbol{\phi}) / \partial \boldsymbol{\phi}$  is even more difficult. This difficulty prevents effective gravity compensation when the IK problem is resolved in the actuator space. The joint motion and *stiffness* resolution method [16] resolves the IK problem in the actuator space, but does not address gravity compensation.

**5.1 Additional Details.** The basic manipulation planning approach, shown in Fig. 3, may exhibit drift in the task manipulation plan due to the finite step size used in the integration. Drift may be mitigated by using an adaptive time-step, or by using the forward map  $f$  to evaluate task error and correct the joint configuration using the Newton–Raphson method.

The algorithm presented in Fig. 3 does not address joint limits. Joint limits, however, *cannot* be ignored for passive compliance control. Every joint must have a positive compliance value regardless of the VSA design. The domain of  $q_{c_i}$  is  $(-\infty, \infty)$  but the feasible subset is given by  $q_{c_i} \in [c_{\min}, c_{\max}]$ , where  $0 < c_{\min} < c_{\max} < \infty$  are the minimum and maximum compliances afforded by the VSA.

Although not presented in Fig. 3, joint limits are addressed using the saturation in the null space (SNS) algorithm [22], where

redundancy is utilized to avoid joint limits only when the pseudoinverse method would produce a limit-violating joint motion. The SNS algorithm is used between steps 3 and 4. Alternatively, joint limits could be accounted for by using quadratic programming.

## 6 Particle Planar Example

Consider a 3R-VSA planar manipulator performing the constrained manipulation task in  $E(2)$  shown in Fig. 4. The manipulator has a total length (reach) of  $L$  and a total weight of  $f_g$ . The normalized link lengths are dimensionless proportions of the total length:  $l_1 = 0.46$ ,  $l_2 = 0.43$ , and  $l_3 = 0.11$  (anthropomorphic ratios [23]). The gravitational force experienced by each link is expressed as proportions of the total weight  $f_g$ . The weight of each link is proportional to its length and its mass center is located at its geometric center as illustrated in Fig. 4. The joint coordinates are given by

$$\mathbf{q} = [q_{p_1}, q_{p_2}, q_{p_3}, q_{c_1}, q_{c_2}, q_{c_3}]^T$$

The kinematic joint positions  $q_{p_1}$ ,  $q_{p_2}$ , and  $q_{p_3}$  are expressed in radians;  $q_{c_1}$ ,  $q_{c_2}$ , and  $q_{c_3}$  are joint compliances normalized by  $f_g$ . The normalized joint compliance matrix is

$$\mathbf{C}_q = \begin{bmatrix} q_{c_1} & 0 & 0 \\ 0 & q_{c_2} & 0 \\ 0 & 0 & q_{c_3} \end{bmatrix} \quad (21)$$

The constrained manipulation task is to slide a block along a rigid surface and contact a rigid wall. The block center is to follow the motion plan given by:

$$\mathbf{x}_p(t) = \begin{bmatrix} 0.3 + 0.4(t) \\ 0.15 \end{bmatrix} \quad (22)$$

where, for normalized time,  $t = 0$  at the start and  $t = 1$  at the end of the task. The task plan is dimensionless, normalized by  $L$ .

Since the block is in contact with a horizontal rigid surface, the task compliance in the vertical direction is high to limit the constraint force that may result from error in the commanded vertical position of the task. In moving the block, friction is considered a disturbance impacting the desired motion. The task compliance in the horizontal direction is low to limit the end-effector deflection from the planned path due to friction. The compliance in the horizontal direction increases as the block approaches the wall to limit the interaction force from wall contact. In Fig. 4, the desired compliance ellipse<sup>4</sup> is shown at four instants in time, where the major and minor radii of the ellipse are eigenvalues of the task compliance matrix.

The normalized task compliance plan (task compliance normalized by  $L/f_g$ ) in vector form is

$$\mathbf{x}_c(t) = \text{sort}^\Delta \left( \begin{bmatrix} \frac{\lambda_2}{\gamma(t)} & 0 \\ 0 & \lambda_2 \end{bmatrix} \right) \quad (23)$$

where the larger eigenvalue is  $\lambda_2 = 0.1$  and  $\gamma = \lambda_2 / \lambda_1$  is the ellipse aspect ratio that transitions from 10 to 1 according to the profile given by  $\gamma(t) = 11 - 10^t$ .

The goal for implementing passive stiffness control is to find an actuator plan  $\boldsymbol{\phi}(t)$  that executes the task manipulation plan  $\mathbf{x}(t) = [\mathbf{x}_p(t)^T, \mathbf{x}_c(t)^T]^T$ . To do this, the task forward mapping function  $f(\mathbf{q})$  and task Jacobian  $\mathbf{J}$  are needed to locally resolve the compliance extended IK problem.

<sup>4</sup>The compliance ellipse here is the end-effector's displacement image from the unit force ( $\mathbf{f}^T \mathbf{f} = 1$ ) applied to the end-effector with task compliance  $\mathbf{C}_c$ ,  $\Delta \mathbf{x} = \mathbf{C}_c \mathbf{f}$ .

**6.1 Mapping From Joint Space to Task Space.** The task forward mapping functions are easily derived from the distal position of each link relative to its joint position. Let  $\mathbf{p}_i = [p_{x_i}, p_{y_i}]^T$  be the position of link- $i$ 's distal point (end-effector or next joint) relative to joint- $i$ 's position. For the manipulator in this case study

$$\mathbf{p}_1 = [l_1 \cos(q_{p_1}), l_1 \sin(q_{p_1})]^T \quad (24a)$$

$$\mathbf{p}_2 = [l_2 \cos(q_{p_1} + q_{p_2}), l_2 \sin(q_{p_1} + q_{p_2})]^T \quad (24b)$$

$$\mathbf{p}_3 = [l_3 \cos(q_{p_1} + q_{p_2} + q_{p_3}), l_3 \sin(q_{p_1} + q_{p_2} + q_{p_3})]^T \quad (24c)$$

The forward kinematic mapping function is

$$\mathbf{f}_p(\mathbf{q}) = \mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3 \quad (25)$$

The kinematic Jacobian of this manipulator is found by taking the partial derivative of Eq. (25) with respect to each kinematic joint coordinate. Since the forward kinematic function is composed of sin and cos terms,  $\partial p_{x_i} / \partial q_k = -p_{y_i}$  and  $\partial p_{y_i} / \partial q_k = p_{x_i}$ , the resulting Jacobian is

$$\mathbf{J}_{pp} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} [(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3) \quad (\mathbf{p}_2 + \mathbf{p}_3) \quad \mathbf{p}_3] \quad (26)$$

The forward compliance function  $\mathbf{f}_c(\mathbf{q})$  is readily obtained by substituting the kinematic Jacobian matrix, Eq. (26), and the joint compliance matrix, Eq. (21), into Eq. (8).

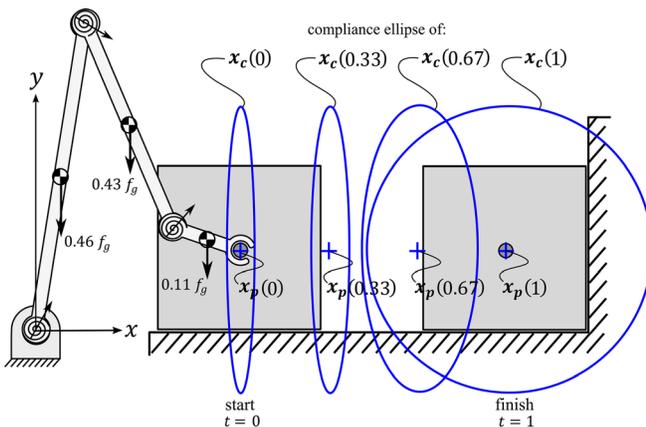
The compliance Jacobian matrix  $\mathbf{J}_c$  is constructed one column at a time using Eq. (12) substituted into Eq. (11). The partition  $\mathbf{J}_{cp}$  is readily evaluated with the partial derivatives of the kinematic Jacobian with respect to the kinematic joint variables:

$$\frac{\partial \mathbf{J}_{pp}}{\partial q_1} = [-(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3) \quad -(\mathbf{p}_2 + \mathbf{p}_3) \quad -\mathbf{p}_3] \quad (27a)$$

$$\frac{\partial \mathbf{J}_{pp}}{\partial q_2} = [-(\mathbf{p}_2 + \mathbf{p}_3) \quad -(\mathbf{p}_2 + \mathbf{p}_3) \quad -\mathbf{p}_3] \quad (27b)$$

$$\frac{\partial \mathbf{J}_{pp}}{\partial q_3} = [-\mathbf{p}_3 \quad -\mathbf{p}_3 \quad -\mathbf{p}_3] \quad (27c)$$

Equations (27a)–(27c) are used in Eq. (12), for  $k=1, 2$ , and  $3$  to construct the first three columns of the compliance Jacobian, respectively. The last three columns,  $\mathbf{J}_{cc}$ , are evaluated with Eq. (13) and used in Eq. (12), for  $k=4, 5$ , and  $6$ .



**Fig. 4** Planar 3R manipulator and manipulation plan. Four equally spaced instants in time illustrate the continuous end-effector motion plan  $\mathbf{x}_p(t)$  and the continuous compliance manipulation plan  $\mathbf{x}_c(t)$ .

The compliance augmented Jacobian relating augmented joint motion to augmented task motion is sufficient for solving the local inverse kinematics problem.

**6.2 Mappings From Joint Space to Actuator Space.** The actuator mapping and actuator Jacobian are defined below for gravity compensation and for joint redundancy resolution according to the minimum norm actuator motion criterion.

From the joint configuration, the kinematic actuator mapping  $\mathbf{h}_p(\mathbf{q})$  is used to determine the actuator configuration that yields the joint configuration at static equilibrium. At static equilibrium, the external torques at each joint are balanced by the internal torque from the VSA elastic element. The normalized external torque is given by

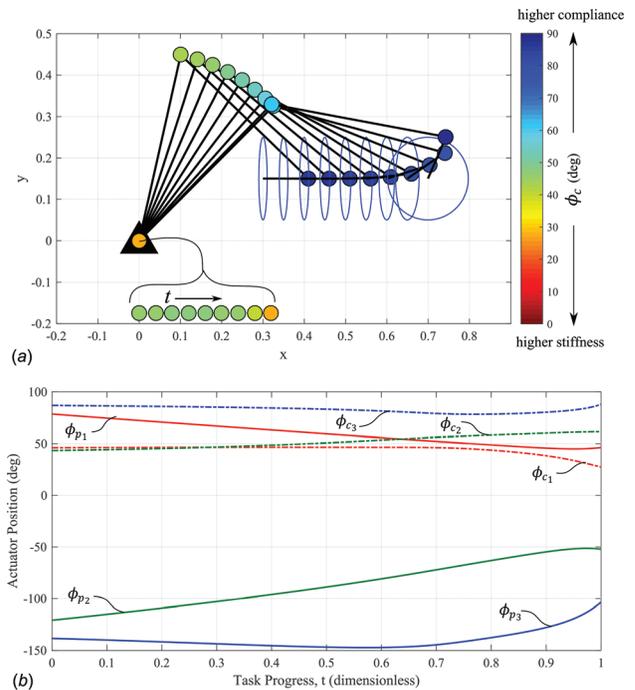
$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} \frac{p_{x_1}}{2} & \left(p_{x_1} + \frac{p_{x_2}}{2}\right) & \left(p_{x_1} + p_{x_2} + \frac{p_{x_3}}{2}\right) \\ 0 & \frac{p_{y_2}}{2} & \left(p_{y_2} + \frac{p_{y_3}}{2}\right) \\ 0 & 0 & \frac{p_{y_3}}{2} \end{bmatrix} \begin{bmatrix} 0.46 \\ 0.43 \\ 0.11 \end{bmatrix} \quad (28)$$

where  $[0.46, 0.43, 0.11]^T$  are the link weights normalized by  $f_g$ .

The kinematic actuator function  $\mathbf{h}_p(\mathbf{q})$  is formed by substituting Eqs. (28) and (21) into Eq. (14).

The VSAs of this example have an exponential compliance versus actuation profile like that of Ref. [10], where  $h_{c_i}(q_{c_i})$  is given by Eq. (15) with  $\xi = 5.86$  and  $c_0 = 0.001$ . The same VSA is used for each joint in this example.

The actuator Jacobian is calculated by  $\mathbf{J}_\phi = \partial \mathbf{h}(\mathbf{q}) / \partial \mathbf{q}$  and is evaluated one column at a time similar to the task compliance Jacobian construction. The weighing matrix used in Eq. (4) is  $\mathbf{W} = \mathbf{J}_\phi^T \mathbf{J}_\phi$ . This weighing matrix resolves the redundancy by minimizing the norm of the actuator motion.



**Fig. 5** Locally resolved manipulation plan starting at  $\mathbf{q}_0$ . (a) Stroboscopic image of the manipulator performing the task. The color of each joint indicates the compliance actuator position. The compliance ellipse is shown at each snapshot. (b) Position profiles of each joint's kinematic and compliance actuator.

**6.3 Results.** Locally resolved joint plans using the weighted pseudoinverse are generated with the numerical integration procedure presented in Sec. 5. An adaptive time-step is used to ensure that the task error is below a specified tolerance and that the numerically resolved joint motion accurately follows the true locally optimal plan.

The first step in the procedure is to find an admissible initial joint configuration from the initial task configuration using an analytical procedure (or search algorithm) denoted by  $\mathcal{A}$  in Fig. 3. For the initial task point  $\mathbf{x}(0)$ , the set of admissible kinematic joint configurations is given by the analytical inverse kinematic solution of a 4 bar mechanism with the end-effector as a grounded joint at  $\mathbf{x}_p(0)$ . The orientation of the end-effector, given by  $\psi = q_{p1} + q_{p2} + q_{p3}$ , is the input angle that resolves the kinematic joint configuration. The results here consider the “elbow-up” pose. The joint compliance configuration is uniquely determined by the desired task compliance and the kinematic joint configuration; formulas for the joint compliance are presented in Ref. [13]. Therefore, the set of admissible initial joint configurations is a one-dimensional set defined by and  $\psi$  at  $\mathbf{x}(0)$ .

Consider the initial joint configuration for  $\psi = -\pi$  rad:

$$\mathbf{q}_0 = [77.3^\circ, -121^\circ, -136^\circ, 0.111, 0.0840, 7.16]^\top$$

Starting at this configuration, the manipulation plan is generated by integrating Eq. (5); the resulting plan is shown in Fig. 5. Figure 5(a) shows equally time-spaced snapshots of the manipulator performing the task, including the initial and final task times. The color of each joint indicates the compliance actuator position  $\phi_c$  for that joint. Because joint 1 does not translate; its compliance actuator position is indicated by the colored circles below the manipulator. Figure 5(b) shows the continuous actuator position profiles for the task.

Figure 5(a) shows the manipulator *exactly* tracking the task compliance *when there is no disturbance*. Disturbance from uncharacterized friction causes displacement of the end-effector from the commanded task position. The joint configuration, also displaced, yields a slightly different task compliance since task compliance depends on joint kinematics. Since the direction of undesirable forces is known for this task, the *task* compliance is designed to be stiff in the direction of motion to keep the end-effector displacement small (as stated in this example).

## 7 Conclusion

This paper identified a method of passive stiffness control of redundant serial manipulators to achieve effective interaction for performing constrained manipulation tasks. Actuator commands that produce the desired time-varying end-effector position and stiffness were obtained by extending a standard RIK approach to include compliance. Joint compliances were resolved simultaneously with the joint kinematics using the weighted pseudoinverse of the total Jacobian (including the compliance Jacobian). Redundancy was resolved using the minimum norm actuator motion criterion, where the actuator coordinates include kinematic actuators and compliance actuators. Gravity compensation was also addressed.

## Funding Data

- National Science Foundation (Grant No. IIS-1427329).

## References

- [1] Hogan, N., 1985, “Impedance Control: An Approach to Manipulation—Part I: Theory,” *ASME J. Dyn. Syst., Meas., Control*, **107**(1), pp. 17–24.
- [2] Hogan, N., 1985, “Impedance Control: An Approach to Manipulation—Part II: Implementation,” *ASME J. Dyn. Syst., Meas., Control*, **107**(1), pp. 8–16.
- [3] Caccavale, F., Natale, C., Siciliano, B., and Villani, L., 1999, “Six-DOF Impedance Control Based on Angle/Axis Representations,” *IEEE Trans. Rob. Autom.*, **15**(2), pp. 289–300.
- [4] Ott, C., Albu-Schäffer, A., Kugi, A., and Hirzinger, G., 2008, “On the Passivity-Based Impedance Control of Flexible Joint Robots,” *IEEE Trans. Rob.*, **24**(2), pp. 416–429.
- [5] Albu-Schäffer, A., Ott, C., Frese, U., and Hirzinger, G., 2003, “Cartesian Impedance Control of Redundant Robots: Recent Results With the DLR-Light-Weight-Arms,” IEEE International Conference on Robotics and Automation (ICRA’03), Taipei, Taiwan, Sept. 14–19, pp. 3704–3709.
- [6] Dietrich, A., Ott, C., and Albu-Schäffer, A., 2015, “An Overview of Null Space Projections for Redundant, Torque-Controlled Robots,” *Int. J. Rob. Res.*, **34**(11), pp. 1385–1400.
- [7] Ferraguti, F., Secchi, C., and Fantuzzi, C., 2013, “A Tank-Based Approach to Impedance Control With Variable Stiffness,” IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, May 6–10, pp. 4948–4953.
- [8] Kim, B.-S., Kim, Y.-L., and Song, J.-B., 2011, “Preliminary Experiments on Robotic Assembly Using a Hybrid-Type Variable Stiffness Actuator,” IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, Hungary, July 3–7, pp. 1076–1080.
- [9] Jafari, A., Tsagarakis, N. G., and Caldwell, D. G., 2011, “AwAS-II: A New Actuator With Adjustable Stiffness Based on the Novel Principle of Adaptable Pivot Point and Variable Lever Ratio,” IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9–13, pp. 4638–4643.
- [10] Groothuis, S. S., Rusticelli, G., Zucchelli, A., Stramigioli, S., and Carloni, R., 2014, “The Variable Stiffness Actuator vsaUT-II: Mechanical Design, Modeling, and Identification,” *IEEE/ASME Trans. Mechatronics*, **19**(2), pp. 589–597.
- [11] Schimmels, J. M., and Garces, D. R., 2015, “The Arched Flexure VSA: A Compact Variable Stiffness Actuator With Large Stiffness Range,” IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, May 26–30, pp. 220–225.
- [12] Albu-Schäffer, A., Fischer, M., Schreiber, G., Schoeppe, F., and Hirzinger, G., 2004, “Soft Robotics: What Cartesian Stiffness Can Obtain With Passively Compliant, Uncoupled Joints?,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. Sendai, Japan, Sept. 28–Oct. 2, 3295–3301.
- [13] Huang, S., and Schimmels, J. M., 2016, “Realization of Point Planar Elastic Behaviors Using Revolute Joint Serial Mechanisms Having Specified Link Lengths,” *Mech. Mach. Theory*, **103**, pp. 1–20.
- [14] Petit, F., and Albu-Schäffer, A., 2011, “Cartesian Impedance Control for a Variable Stiffness Robot Arm,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, Sept. 25–30, pp. 4180–4186.
- [15] Petit, F. P., 2014, “Analysis and Control of Variable Stiffness Robots,” *Ph.D. thesis*, ETH Zürich, Zürich, Switzerland.
- [16] Howard, M., Braun, D. J., and Vijayakumar, S., 2011, “Constraint-Based Equilibrium and Stiffness Control of Variable Stiffness Actuators,” IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9–13, pp. 5554–5560.
- [17] Salisbury, J. K., 1980, “Active Stiffness Control of a Manipulator in Cartesian Coordinates,” 19th IEEE Conference on Decision and Control Including the Symposium on Adaptive Processes, Albuquerque, NM, Dec. 10–12, pp. 95–100.
- [18] Whitney, D. E., 1969, “Resolved Motion Rate Control of Manipulators and Human Prostheses,” *IEEE Trans. Man-Mach. Syst.*, **10**(2), pp. 47–53.
- [19] Chan, T. F., and Dubey, R. V., 1995, “A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators,” *IEEE Trans. Rob. Autom.*, **11**(2), pp. 286–292.
- [20] Huang, S., and Schimmels, J. M., 2002, “The Duality in Spatial Stiffness and Compliance as Realized in Parallel and Serial Elastic Mechanisms,” *ASME J. Dyn. Syst., Meas., Control*, **124**(1), pp. 76–84.
- [21] Chen, S.-F., and Kao, I., 2000, “Conservative Congruence Transformation for Joint and Cartesian Stiffness Matrices of Robotic Hands and Fingers,” *Int. J. Rob. Res.*, **19**(9), pp. 835–847.
- [22] Flacco, F., De Luca, A., and Khatib, O., 2015, “Control of Redundant Robots Under Hard Joint Constraints: Saturation in the Null Space,” *IEEE Trans. Rob.*, **31**(3), pp. 637–654.
- [23] Gordon, C. C., Churchill, T., Clauser, C. E., Bradtmiller, B., McConville, J. T., Tebbetts, I., and Walker, R. A., 1989, “Anthropometric Survey of US Army Personnel: Summary Statistics,” Anthropology Research Project, Yellow Springs, OH, Technical Report.