12-2020

Globally Optimal Passive Compliance Control for Tasks Having Multiple Homotopy Classes

Jacob J. Rice
Marquette University

Joseph M. Schimmels
Marquette University, joseph.schimmels@marquette.edu

Mechanism and Machine Theory, Vol. 154 (December 2020): 104094. DOI.

Globally Optimal Passive Compliance Control for Tasks Having Multiple Homotopy Classes

Jacob J. Rice
Department of Mechanical Engineering, Marquette University, Milwaukee, WI
Joseph M. Schimmels
Department of Mechanical Engineering, Marquette University, Milwaukee, WI

Abstract

Redundant serial manipulators with variable stiffness actuators (VSAs) are capable of passive compliance control, in which the elastic behavior of the end-effector is controlled for robust interaction with a stiff environment. This paper addresses the problem of finding the globally optimal joint manipulation path (sequence of joint positions and compliances) that yields a desired task manipulation path (sequence of end-effector positions and compliances) when there is one degree of redundancy. The space of admissible joint paths can be very complex, with multiple bifurcations resulting in multiple homotopy classes of joint paths. Bifurcations due to singularities in the combined kinematic and compliance joint space are quickly identified using geometric conditions and root-finding. Bifurcations due to joint limits are identified using a novel path planner that traces the solution space boundary edges. With the bifurcations located, joint paths in all homotopy class are generated and deformed into locally optimal paths. The best of these is very likely the globally optimal joint path.

Concepts are illustrated in a case study for particle-planar passive compliance control with a 3R-VSA manipulator.

Keywords

Redundant manipulators, Variable stiffness actuation, Inverse kinematics, Manipulation planning, Compliance, Homotopy class

1. Introduction

Passive compliance control is an approach to regulate the force-displacement interaction between a manipulator and its environment by controlling the end-effector's multi-directional compliance (inverse of stiffness) directly, without active control using sensor feedback. Managing the mechanical interaction is very important in tasks with contact constraints, such as turning a crank, polishing a surface, or assembling parts.

Passive compliance control may be implemented using manipulators like that in Fig. 1, with variable stiffness actuators (VSAs), which have adjustable stiffness elements in series with the kinematic actuators [1]. Such manipulators can independently change the equilibrium joint position and physical joint compliance in real-time. Realization of passive compliance control, however, can be difficult even with these types of manipulators [2].

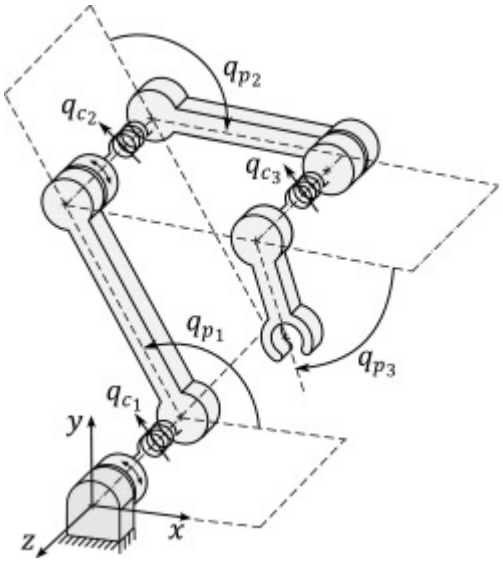

Fig. 1. Planar serial manipulator with variable stiffness actuators (VSAs). For each joint *i*, the primary actuator controls the kinematic joint position q_{p_i} and the VSA controls the joint compliance q_{c_i}.

The set of realizable end-effector compliances strongly depends on the kinematic joint configuration of the manipulator (i.e., spatial locations of the elastic joints). Because a redundant serial manipulator can adjust its kinematic configuration without changing the end-effector pose, the set of realizable end-effector compliances is much larger for a redundant serial manipulator compared to a non-redundant one.

The primary challenge in realizing passive compliance control in a redundant manipulator is determining an appropriate *joint manipulation path*, a continuous sequence of kinematic joint

configurations and joint compliances that realizes the desired *task manipulation path*, a continuous sequence of desired end-effector positions and compliances. The kinematic joint configurations and joint compliances of the joint manipulation path must be *feasible* for the manipulator hardware. For some task manipulation paths, there is no possible joint path that can perform the task. When a path does exist, because of redundancy, there is an infinite set of joint paths that can complete the task.

The problem of selecting a *unique* joint manipulation path is addressed in [3] by extending the redundant inverse kinematic (RIK) path planning problem to include compliance (RIKC). To do this, the joint space of the manipulator and the task space of the end-effector are augmented to include compliance coordinates. However, the path generation method in [3] does not generate the globally optimal path, nor does it guarantee to find a solution though one may exist. To find the globally optimal path or to prove no solution exists, the entire feasible RIKC solution space must be considered.

The *RIK(C) solution space* is a time indexed sequence of self-motion manifolds [4] (connected sets of joint configurations that yield the same task configuration of the end-effector). The RIK(C) solution space may also be constrained by boundary conditions of the joint path (e.g., fixed, free, or periodic). The *feasible* RIK(C) solution space is the subset of realizable joint configurations (those within the hardware limits of the manipulator). The RIK(C) path planning problem is more difficult when the feasible RIK(C) solution space is not simply connected.

The problem of finding the globally optimal solution to the passive compliance control problem has previously not been addressed; nor has the problem of characterizing the connection structure of the solution space. Our recent work, [5], characterizes the connectivity of RIK solution spaces and identifies all joint path homotopy classes by identifying bifurcation points in the RIK solution space. In [5], bifurcations from kinematic singularities are identified using root-finding with the geometric conditions associated with the singularities. The bifurcation branch algorithm cannot be immediately applied to RIKC problems because singularities in the combined kinematic and compliance space have not been investigated. Additionally, [5] does not provide a means of identifying bifurcations due to joint limits, but this must be considered for RIKC problems as VSAs have hardware limits. This work fills in these gaps to use the bifurcation branch algorithm [5] in the RIKC framework [3] for passive compliance control.

1.1. Prior work

There are three areas of prior work relevant to this paper: 1) work related to characterizing the solution space for passive compliance control, 2) work related to generating joint paths for passive compliance control, and 3) work related to finding the globally optimal joint path for RIK path planning problems.

Others have investigated the space of realizable end-effector compliances by sampling the total joint configuration space (kinematic and compliance coordinates) and calculating the end-effector compliance. The space of realizable compliance is visualized using scatter plots [6] or “stiffablity maps” [7]. These relatively sparse sampling approaches, however, do not provide information about the connectivity of the joint space configurations.

Necessary and sufficient conditions for a manipulator to achieve a specified passive compliance have been identified for a variety of manipulator types. The conditions and geometric interpretations for 2

and 3 joint manipulators realizing a particle planar compliance have been identified for revolute joints [8] and for a combination of revolute or prismatic joints [9]. The conditions and the geometric interpretations have been identified for 3R [10], 4R [11], 5R [12], and 6R [13] manipulators realizing a general planar compliance. The conditions and the geometric interpretations have also been identified for 6R [14] manipulators realizing a general spatial compliance. For each type of manipulator, the conditions on manipulator geometry are decoupled from the conditions on the joint compliance values. Results apply to finding a single joint configuration at a single task instance. The connectivity of joint configurations at a single task instance is not addressed, nor is the generation of a time indexed joint path through the RIKC solution space.

Joint paths through the RIKC solution space have been generated [3] using *instantaneous* resolution. Optimal joint velocities are obtained by minimizing a velocity-based cost function (e.g., the weighted norm of the joint velocity) over the set of joint velocities that produce the desired task velocity. Starting from an admissible initial joint configuration, the instantaneously optimal joint path is generated by integrating the optimal joint velocity over the entire path. Joint limits were addressed using the Saturation in the Null Space (SNS) algorithm [15]. The velocity-based instantaneous RIKC path planner [3] has two limitations: 1) it may proceed to a premature termination, failing to complete the task manipulation path even when viable paths exist, and 2) when a path that does complete the task is found, it is not, in general, the globally optimal path or even a locally optimal path. A locally optimal path is a path for which any infinitesimal deformation yields a higher global cost value (according to an integral cost criterion).

Locally optimal solutions to the RIK path planning problem can be obtained using global resolution techniques in which a global cost function (e.g., integral of the instantaneous cost function over the whole path) is minimized over the set of admissible paths. Global resolution is equivalent to an optimal control problem which may be solved: 1) “indirectly” using Pontryagin’s maximum principle [16] or the calculus of variations [17], [18], 2) “directly” using nonlinear programming [5], [19], [20], or 3) using dynamic programming [21], [22]. Dynamic programming identifies the globally optimal path, but only for fixed boundary conditions, in which the joint configurations at the task endpoints are specified. The globally optimal path for fixed boundary conditions will, in general, have a higher cost than that of free (unconstrained) boundary conditions.

For RIK path planning problems with one degree of redundancy, the bifurcation branch algorithm (bb-algorithm) [5] provides the most complete solution. Briefly, the bb-algorithm decomposes the RIK solution space such that all joint path homotopy classes are identified, generates sub-optimal joint paths in each homotopy class, and deforms these joint paths using nonlinear programming and self-motion paths. The best of these joint paths (deformed into locally optimal paths) is very likely the globally optimal joint path.

Fig. 2 provides a very simple example of how the bb-algorithm decomposes the RIK solution space and captures the connection structure in a directed graph called the bifurcation branch roadmap (bb-roadmap). As the task progresses (normalized time increases from 0 to 1), the self-motion manifolds bifurcate at the hyperbolic singularity [23]. (Bifurcations may also occur from joint limits, but [5] does not provide a means of identifying that type of bifurcation.) Bifurcation branches (self-motion paths on either side of the bifurcation point) correspond to the black nodes in the bifurcation branch roadmap.

The bb-roadmap also includes the self-motion manifolds at the task start and end points corresponding to the white nodes. Each homotopy class is uniquely identified by its *route* (sequence of edges connecting a start node to a terminal node). There can be many homotopy classes when multiple bifurcations exist. The bb-roadmap, together with the self-motion paths of its nodes, provide a means of systematically searching the entire RIK solution space for the globally optimal path.

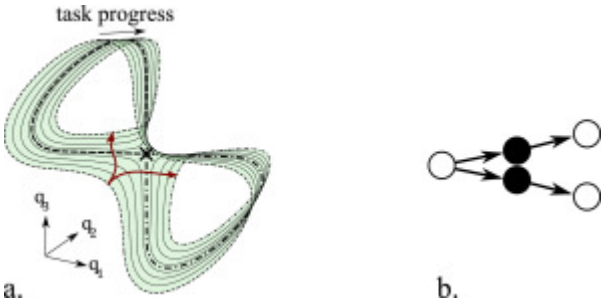

Fig. 2. An RIK solution space with a single bifurcation from a kinematic singularity and its bifurcation branch roadmap. (a) The shaded solution surface embedded in the joint configuration space is a set a self-motion manifolds, a few shown as thin solid lines. The × is a bifurcation point, dashed and dashed-dotted lines are feasible self-motion paths on one side of the bifurcation point or self-motion paths of task endpoints. Solid lines with arrow endpoints are joint paths in different homotopy classes. (b) White nodes correspond to the feasible self-motion paths of task endpoints, pairs of black nodes correspond to the feasible self-motion paths on either side of a singularity bifurcation point.

The bb-algorithm requires high-level information (obtained, for example, from geometric analysis) about: 1) the number and structure[1] of the self-motion manifolds along the task, 2) a characterization of how bifurcations in the self-motion manifolds impact the number of joint path homotopy classes, and 3) a means of identifying the locations of the bifurcation points along the task. This information, used to algorithmically construct a bb-roadmap, has not been provided for any RIKC problem.

1.2. Approach

In this paper, the bb-algorithm previously used in RIK resolution is used in the combined kinematic and compliance (RIKC) solution space to generate the globally optimal joint path for passive compliance control with one degree of redundancy. Here, the analysis and developments needed to construct the bb-roadmap for RIKC problems are provided.

The challenges associated with developing the bb-roadmap in the combined solution space are: 1) identifying the number and structure of self-motion manifolds and the feasible self-motion paths on them, 2) identifying bifurcation nodes associated with singularities, 3) identifying bifurcations associated with joint limits, and 4) identifying edges connecting nodes. The first challenge relates directly to identifying endpoint nodes, but also relates to identifying edges as this information is important for understanding the bifurcation structure and its impact on homotopy classes.

The number of self-motion manifolds impacts the number of endpoint nodes in the bb-roadmap (white nodes). The structure of the self-motion manifolds impacts the number of edges connecting nodes. This information has been identified for the *kinematic* self-motion manifolds of many types of manipulators (e.g., [4], [24], [25]). However, the number and the structure of self-motion manifolds in the combined kinematic and compliance configuration space has not been identified for any type of

manipulator. This paper provides a method for obtaining this information by combining kinematic self-motion manifolds with compliance realization conditions (e.g., [8], [9], [10], [11], [12], [13], [14]).

The identification of singular joint configurations (and their corresponding end-effector configurations) is necessary for finding the bifurcation branches of singularities (black nodes). In [5], these features were identified using known geometric conditions for singularities (and for the task instances at which they occur). However, the geometric conditions for singularities and their task instances in the combined kinematic and compliance configuration (RIKC) spaces are not known for any type of manipulation with VSA robots (e.g., a 3R planar VSA-manipulator performing a particle planar task). This paper provides a method for obtaining this information using compliance realization conditions (e.g., [8], [9], [10], [11], [14]) for simpler manipulators that can achieve the same elastic behavior as the redundant manipulator.

For problems with bifurcations in the feasible RIKC solution space induced by joint limits, it is necessary to identify the bifurcation points to build the bb-roadmap. Because joint compliances are always lower and upper bounded, joint limits must be accounted for in the compliance subspace. An example of bifurcations from joint limits is illustrated in Fig. 3. Joint limit bifurcation points are identified here for differentiable task manipulation paths by tracking the boundary edges of the RIK(C) solution space using an instantaneous path planner based on a modified Saturation in the Null Space (SNS) algorithm [15]. The boundary edge path planner terminates at: 1) a task endpoint (white node), 2) a premature termination point (⊗ node), or 3) a joint limit bifurcation point (gray node). The boundary edge path planner can be used in both RIK and RIKC path planning problems.

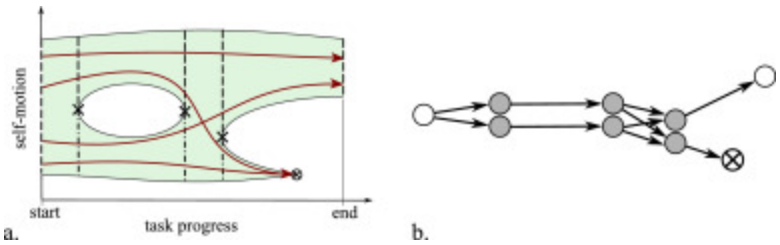

Fig. 3. Bifurcations and pre-mature termination from joint limits on a simple RIK(C) solution space with a single open manifold. (a) The solution space is parameterized by task progress and self-motion. A single joint path (solid line with arrow end-point) is shown in each homotopy class. (b) The bifurcation branch roadmap. White nodes correspond to the feasible self-motion paths of task endpoints (dashed lines at start and end), pairs of gray nodes correspond to the feasible self-motion paths on either side of a joint limit bifurcation point (×), and ⊗ corresponds to a premature termination point.

In addition to describing new procedures for identifying roadmap nodes, new procedures are also developed for identifying roadmap edges. The methods described in [5] for implicitly identifying edges of the bb-roadmap are limited to RIK problems with 1 or 2 closed self-motion manifolds at each task instance. New methods of identifying edges are needed to address the greater number of disjoint self-motion manifolds in RIKC problems. This paper presents an algorithm for constructing the bb-roadmap edges for RIK(C) problems with any number of *open* self-motion manifolds and with bifurcations from both singularities and joint limits.

1.3. Paper overview

This paper provides means of constructing the bifurcation branch roadmap (bb-roadmap) that identifies the connectivity of the solution space for RIKC path planning with one degree of redundancy. Strategies for identifying the number and the structure of self-motion manifolds, and their bifurcations from singularities and their bifurcations from joint limits are described. These strategies are used to generate the nodes of the bb-roadmap. Strategies for identifying the edges of the bb-roadmap are also provided.

Section 2 reviews the technical background needed to solve the RIKC path planning problem. Section 3 provides means of identifying the number and the structure of self-motion manifolds in the total joint space (kinematic and compliance coordinates). Section 4 describes procedures for identifying self-motion bifurcations from singularities in the total joint space. Section 5 provides a method for quickly identifying bifurcation points that result from joint limits. Section 6 describes a new algorithm for generating the bb-roadmap for manipulation with open self-motion manifolds and with bifurcations from both singularities and joint limits. Section 7 demonstrates the procedures and provides the results of a case study in which a 3R-VSA manipulator is used to turn a crank despite geometric uncertainties. Section 8 provides a brief summary and conclusion.

2. Background

This section reviews technical background used in solving the redundant inverse kinematics and compliance (RIKC) path planning problem. The augmentation of the RIK problem to include compliance (RIK $\rightarrow$ RIKC) is described first. With the RIKC framework [3], joint manipulation paths for passive compliance control are obtained using the bb-algorithm [5] to find the best joint path in the combined solution space. Technical descriptions of the RIK(C) path planning problem and its solution space are provided. To use the bb-algrithm, connectivity characteristics of the RIKC solution space are required. Bifurcations from singularities and bifurcations from joint limits must be identified. Bifurcations from joint limits are identified using a boundary edge path planner based on a *modified* Saturation in the Null Space (SNS) algorithm [15]. The basic[2] version of the SNS algorithm is briefly reviewed here.

2.1. Compliance extended redundant inverse kinematics

Serial manipulators with VSAs can control each joint position and its compliance independently. As such, the total joint configuration space includes both kinematic coordinates and compliance coordinates. The *kinematic* joint configuration is $\mathbf{q}_p = \left[q_{p_1}, q_{p_2}, \dots, q_{p_{n_p}}\right]^T$,, where the coordinate $q_{p_i} \in \left[q_{p_{i_{\min}}}, q_{p_{i_{\max}}}\right]$ is the relative angle between link *i* and link $(i-1)$ (link 0 is the ground) and n_p is the number of joints. The *compliance* joint configuration is $\mathbf{q}_c = \left[q_{c_1}, q_{c_2}, \dots, q_{c_{n_p}}\right]^T$,, where the coordinate $q_{c_i} \in \left[q_{c_{i_{\min}}}, q_{c_{i_{\max}}}\right]$ is the compliance of joint *i*. Note, unlike revolute kinematic joints, a passive compliance has a fundamental limit for which qcimin≥0. The *total* joint configuration is the concatenation of these: $\mathbf{q} = \left[\mathbf{q}_p^T, \mathbf{q}_c^T\right]^T$..

Likewise, the end-effector position/orientation configuration $\mathbf{x}_p$ and the task compliance matrix $\mathbf{C}_x$ are independently controlled. These are similarly combined into the total task configuration $\mathbf{x} = \left[\mathbf{x}_p^T, \mathbf{x}_c^T\right]^T$„ where $\mathbf{x}_c$ is the set of upper triangular elements of $\mathbf{C}_x$ $C_{x_{ij}}, j \geq i$. This set contains the minimum number of elements needed to define a symmetric positive definite task compliance matrix.

Because of this structure, the forward mapping function $\boldsymbol{f}$: $Q \rightarrow X$ is

$$\boldsymbol{f}(\mathbf{q}) = \begin{bmatrix} \boldsymbol{f}_p(\mathbf{q}_p) \\ \boldsymbol{f}_c(\mathbf{q}) \end{bmatrix},$$

where the forward kinematic function $\boldsymbol{f}_p$ gives $\mathbf{x}_p$ and the forward compliance function $\boldsymbol{f}_c$ gives $\mathbf{x}_c$. Note that the kinematic function only depends on the kinematic joint variables; whereas, the compliance function depends on all joint variables.

The total forward mapping function $\boldsymbol{f}(\mathbf{q})$ and total Jacobian matrix $\mathbf{J}(\mathbf{q}) = \frac{\partial \boldsymbol{f}(\mathbf{q})}{\partial \mathbf{q}}$ are used to resolve the compliance extended inverse kinematics (RIKC) problem using the same resolution methods used in the RIK problem.

2.2. Redundant inverse kinematics path planning

Consider a task space with *m* coordinates and a manipulator joint space with *n* > *m* joint coordinates. The general optimal RIK path planning problem is

(1)

$$\begin{aligned} \text{min.} \quad & G_{\text{global}}\big(\boldsymbol{q}(t)\big) \\ \text{s.t.} \quad & \boldsymbol{f}\big(\boldsymbol{q}(t)\big) = \boldsymbol{x}(t) \\ & \mathbf{q}_{\text{min}} \leq \boldsymbol{q}(t) \leq \mathbf{q}_{\text{max}} \end{aligned} \quad ,$$

where $\boldsymbol{q}(t)$ and $\boldsymbol{x}(t)$ are paths parameterized by normalized time (i.e., task progress) in the joint and task spaces, respectively, $\mathbf{q}_{\text{min}}$ and $\mathbf{q}_{\text{max}}$ are lower and upper limits of the joint coordinates, and $G_{\text{global}}(\boldsymbol{q}(t))$ is the global cost function. The global optimization problem may also be constrained by boundary conditions, usually *fixed, free*, or *periodic*. Fixed boundary conditions specify the joint configurations at the task endpoints, whereas free (unconstrained) boundary conditions do not specify the endpoint joint configurations. Periodic boundary conditions (applicable for cyclic task paths), do not specify the endpoint configurations, but require them to be the same. Free boundary conditions yield the lowest cost path.

The domain of the optimization is the *feasible* RIK(C) solution space, where the RIK(C) solution space is a sequence of sets of self-motion manifolds for every $\mathbf{x} \in \boldsymbol{x}(t)$ [4] given by

(2)

$$\boldsymbol{f}^{-1}(\mathbf{x}) = \bigcup_{i}^{n_s} \boldsymbol{\mathcal{M}}_i(\boldsymbol{\psi})$$

where n_s is the number of disjoint self-motion manifolds $\boldsymbol{\mathcal{M}}_i(\boldsymbol{\psi})$,, and $\boldsymbol{\psi}$ is a set of self-motion parameters. For RIK(C) problems with one degree of redundancy $(n - m = 1)$, each self-motion

manifold is a 1-dimensional path $\boldsymbol{q}_i(\psi)$. The portions of the self-motion paths within the joint limits are feasible self-motion paths. The feasible RIK(C) solution space is the collection of all feasible self-motion paths at each instance in time.

Technically, the manifold description of the preimage (2) is only valid for *regular values*, task configurations for which its preimage does not contain a singularity, i.e., a joint configuration qs such that $\text{rank}\big(\mathbf{J}(\mathbf{q}_s)\big) < m$. Each *coregular value* is a task configuration $\mathbf{x}_{cr}$ for which the preimage contains both singularities and non-singular joint configurations [4]. Although the preimage of a coregular value is not a collection of manifolds, it is a collection of self-motion paths for problems with one degree of redundancy. These paths are split at the singularity into separate branches (e.g., dashed and dashed-dotted lines in Fig. 2a).

2.3. SNS algorithm

Joint paths through the RIK(C) solution space can be generated using instantaneous resolution methods where the optimal joint velocity that yields the desired task velocity is identified [26] using

(3)

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\dot{\mathbf{q}}_N,$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{J}^{\dagger}$ is the weighted pseudoinverse of $\mathbf{J}(\mathbf{q})$,, and $\dot{\mathbf{q}}_N$ is a "preferred" joint motion, often selected to be $\mathbf{0}$.

Starting at a joint configuration of the task start point, (3) may be integrated over the task manipulation path $\boldsymbol{x}(t)$. The generated joint path, however, may violate joint limits.

The basic Saturation in the Null Space (SNS) algorithm [15] first identifies a joint motion using (3) still with $\dot{\mathbf{q}}_N = \mathbf{0}$,, but with the additional restriction

(4)

$$\frac{\mathbf{q}_{\text{min}} - \boldsymbol{q}(t)}{\Delta t} \leq \dot{\mathbf{q}} \leq \frac{\mathbf{q}_{\text{max}} - \boldsymbol{q}(t)}{\Delta t},$$

where Δt is the path integration step size and $\boldsymbol{q}(t)$ is the current joint configuration of the generated path. This prevents the generated joint path from violating joint limits.

To satisfy restriction (4), the SNS algorithm first calculates $\dot{\mathbf{q}}$ using (3), then checks if the inequality constraints of (4) are satisfied. If they are not, the most violated joint value, q_i is saturated at its limiting value by setting element *i* of $\dot{\mathbf{q}}_N$ to element *i* of the inequality constraint (4). Eq. (3) is re-calculated, replacing $\mathbf{J}^{\dagger}$ with $\mathbf{J}^{\dagger}_{\text{SNS}} = (\mathbf{JS})^{\#}$,, where

$(\cdot)$

is the Moore-Penrose pseudoinverse and $\mathbf{S}$ is the saturation matrix: an $n \times n$ diagonal matrix with element $S_{ii} = 1$ for non-saturated joints or $S_{ii} = 0$ for saturated joints. This forces the non-saturated joints to compensate. The process is repeated until either a feasible joint velocity is identified or until $\text{rank}(\mathbf{JS}) < m$.

In this paper, the SNS algorithm is modified to track a boundary edge of the RIKC solution space. By tracking the boundary edge until the path can no longer be generated (i.e., rank($\mathbf{JS}$) $< m$), the joint limit bifurcation points and premature termination points are identified.

3. RIKC self-motion manifolds

As stated previously, the main challenge in passive compliance control is to solve the redundant inverse kinematics and compliance (RIKC) path planning problem. There are multiple ways for a joint path to traverse the RIKC solution space's complicated connection structure. Characterizing the connectivity of the RIKC solution space is important for identifying the globally optimal joint path for passive compliance control.

The first aspect in characterizing the connectivity of the RIKC solution space is to identify the number and structure of the self-motion manifolds of regular task points on the task manipulation path. This information relates directly to generating the bb-roadmap nodes at the task endpoints (white start and terminal nodes). This section describes a strategy for identifying the number and structure of self-motion manifolds for RIKC problems with one degree of redundancy. Each disjoint self-motion manifold is distinguished using an identification number that is assigned based on geometric properties of the manipulator. The self-motion manifold identification numbers are later used to identify bb-roadmap edges. The strategy is first described in general, then the concepts are demonstrated for a 3R-VSA manipulator like that in Fig. 1.

3.1. General strategy

Because the compliance mapping depends on the kinematic joint configuration, but the kinematic mapping is independent of the joint compliance configuration, the kinematic self-motion manifolds are identified first. Although the kinematic joint configurations are connected on each kinematic self-motion manifold, the total joint configurations in the *combined* joint space are not continuously connected in the compliance subspace. The kinematic self-motion manifolds are divided into regions where the compliance configurations are connected.

Consider a total task configuration on the task manipulation path:

$$\mathbf{x} \in \boldsymbol{x}(t) = \begin{bmatrix} \mathbf{x}_p \in \boldsymbol{x}_p(t) \\ \mathbf{x}_c \in \boldsymbol{x}_c(t) \end{bmatrix}.$$

The preimage of the kinematic task configuration is

(5)

$$\boldsymbol{f}_p^{-1}(\mathbf{x}_p) = \bigcup_{i=1}^{n_s} \boldsymbol{\mathcal{M}}_{p_i}(\boldsymbol{\psi})$$

where n_s is the number of disjoint kinematic self-motion manifolds $\boldsymbol{\mathcal{M}}_{p_i}(\boldsymbol{\psi})$. If available, analytical descriptions of the kinematic self-motion manifolds are used. Otherwise, numerical descriptions of the kinematic self-motion manifolds are used (e.g., parameterized by arc length [5]).

The preimage of the compliance task configuration given the kinematic joint configuration,

(6)

$$\boldsymbol{f}_c^{-1}(\mathbf{x}_c) \mid \mathbf{q}_p \in \boldsymbol{\mathcal{M}}_{p_i}(\boldsymbol{\psi}),$$

is unique for many manipulation cases [8], [9], [11], [12]. The joint compliance values for these cases are readily identified by joint compliance synthesis formulas related to the compliance realization conditions. For manipulation cases with equality constraints in their compliance realization conditions, the equality constraints are used to identify a one-dimensional sub-manifold (path) of the kinematic self-motion manifold $\boldsymbol{q}_{p_i}(\psi)$.

The joint synthesis formulas provided in [8], [9], [11], [12], evaluated over ψ, have discontinuities when any two twists align. The twists of any two joints (say joint *i* and joint *j*) align when

(7)

$$\mathbf{t}_j = \alpha \mathbf{t}_i,$$

where α is a scalar.

Because of the discontinuity, the total joint configurations are not continuously connected over the kinematic self-motion manifold. Instead, each set of continuously connected joint configurations on either side of the discontinuity is an *open* self-motion manifold. The number of self-motion manifolds for the task configuration $\mathbf{x} = \left[\mathbf{x}_p^T, \mathbf{x}_c^T\right]^T$ depends on the number of kinematic self-motion manifolds of $\mathbf{x}_p$ and on the number of twist alignment cases over each kinematic self-motion manifold. The bounding twist alignment cases of each self-motion manifold and the associated kinematic self-motion manifold determine the identification number.

The number of feasible self-motion paths (the connected subsets of *feasible* joint configurations on the self-motion manifolds) may be different. Each self-motion manifold may have 0, 1, or multiple disjoint feasible self-motion paths. The feasible self-motion manifolds are identified using a one dimensional search over each self-motion manifold, checking if the joint configuration is feasible. Bounds (endpoints) of the feasible self-motion paths are joint configurations with a joint variable saturated at its limit. The feasible self-motion paths at the task endpoints are used as end-point nodes in the bb-roadmap.

This general strategy of identifying self-motion manifolds is described in detail below for a 3R-VSA manipulator capable of performing a task in which the end-effector follows a desired path in both position and compliance.

3.2. 3R-VSA manipulator

Consider any end-effector position inside the dexterous workspace for which the distal link of the manipulator can fully rotate without changing the end-effector position. The *kinematic* self-motion manifolds are characterized by the analytical formulas of a four-bar Grashof mechanism, parameterized by the end-effector orientation $\psi = q_{p_1} + q_{p_2} + q_{p_3}$. There are two disjoint kinematic self-motion manifolds corresponding to the “elbow-up” and “elbow-down” solutions. Each of these kinematic self-motion manifolds is a *closed* manifold, where a manipulator performing a self-motion returns to its original kinematic configuration after a full rotation of the end-effector.

Given the kinematic joint configuration (parameterized by ψ and the elbow pose, i.e., "up" or "down"), the joint compliance configuration is resolved by the joint compliance synthesis equations provided in [8]. Since the joint compliance synthesis equations yield a *unique* joint compliance configuration (for all regular task configurations in which two twists do not align), the kinematic self-motion parameter ψ is suitable as the self-motion parameter that resolves both kinematics and compliance.

Although the joint compliance configurations are unique for each kinematic configuration, the joint compliance configurations are not continuously connected over the *kinematic* self-motion manifolds. The joint compliance synthesis formulas [8] yield undefined joint compliance values at twist alignment cases.

There are four relevant twist alignment cases satisfying (7):

$$\begin{aligned} \mathrm{a}^{+}&: \quad \mathbf{t}_2 \parallel \mathbf{t}_3 \text{and} \alpha > 0, \\ \mathrm{b}^{+}&: \quad \mathbf{t}_1 \parallel \mathbf{t}_3 \text{and} \alpha > 0, \\ \mathrm{a}^{-}&: \quad \mathbf{t}_2 \parallel \mathbf{t}_3 \text{and} \alpha < 0, \\ \mathrm{b}^{-}&: \quad \mathbf{t}_1 \parallel \mathbf{t}_3 \text{and} \alpha < 0. \end{aligned}$$

Each kinematic self-motion manifold ("elbow-up" and "elbow-down") has kinematic configurations corresponding to these twist alignment cases. The four kinematic configurations for the "elbow-up" manifold are shown in Fig. 4a. The self-motion parameter value ψ^* for each twist alignment case ($\psi^* = \psi_{\mathrm{a}^+}, \psi_{\mathrm{b}^+}, \psi_{\mathrm{a}^-}$, or ψ_{b^-}) is shown in Fig. 4b.

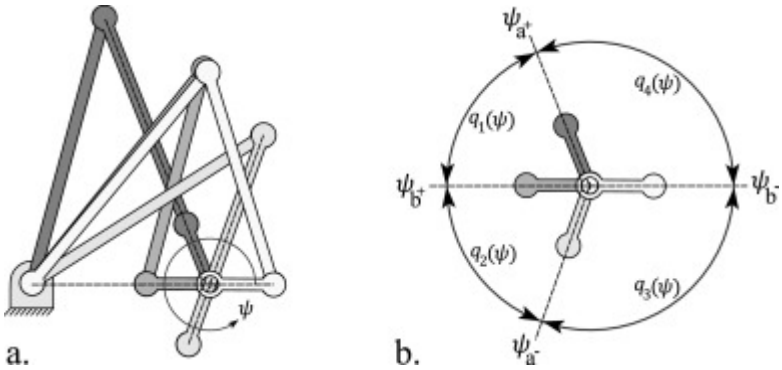

Fig. 4. Twist alignment cases of the elbow-up self-motion manifold. (a) Kinematic configurations of the twist alignment cases in order of darkest to lightest: a^+, b^+, a^-, b^-. (b) Elbow-up self-motion manifolds: $\boldsymbol{q}_i(\psi)$, $i = 1 - 4$ separated by kinematic self-motion parameter values: $\psi_{\mathrm{a}^+}, \psi_{\mathrm{b}^+}, \psi_{\mathrm{a}^-}, \psi_{\mathrm{b}^-}$.

The joint compliances associated with the aligned twists, approach $\pm\infty$ as ψ approaches ψ^* from the left, and approach $\mp\infty$ as ψ approaches ψ^* from the right. Because the compliance values diverge, the self-motion manifold is *open*. Therefore, the preimage of a regular task configuration is composed of eight open self-motion manifolds:

(8)

$$\boldsymbol{f}^{-1}(\mathbf{x}) = \bigcup_{i=1}^{8} \boldsymbol{q}_i(\psi),$$

where $\boldsymbol{q}_i(\psi)$, i=1to4 is a self-motion manifold in the elbow-up pose (shown in Fig. 4b) and $\boldsymbol{q}_i(\psi)$, i=5to8 is a self-motion manifold in the elbow-down pose. Each subscript value is the self-motion manifold identification number corresponding to a unique set of bounding twist alignment cases and elbow pose. Having the same identification number is a necessary, but not sufficient, condition for two self-motion manifolds to be homotopic. Note, the number and type of twist alignment cases is different for task configurations outside the dexterous workspace.

Although each self-motion manifold is unbounded in joint space, the manifold is fully captured by the bounded kinematic self-motion parameter $\psi \in (\psi_{a^{\pm}}, \psi_{b^{\pm}})$. Each self-motion manifold is unbounded in that each joint compliance may range between $\pm\infty$. However, the compliance of each joint-*i* has practical limits $q_{c_i} \in \left[q_{c_{i_{\min}}}, q_{c_{i_{\max}}}\right], q_{c_{i_{\min}}} \geq 0$. Therefore, the feasible regions on self-motion manifolds are necessarily bounded by joint configurations with a joint compliance saturated at its limit.

4. Bifurcations from singularities

To characterize the connectivity of the RIKC solution space, bifurcation nodes associated with singularities must be identified. This involves 1) identifying the task instances at which they occur, i.e., coregular values, 2) identifying the coregular self-motion paths at the coregular values, 3) identifying the singularity and splitting each coregular self-motion path into separate "branches", on either side of the singularity.

This section first describes the bifurcation structure of the self-motion manifolds presented in Section 3.2 and describes its relationship to homotopy classes of joint paths. The details for identifying each coregular value and its corresponding coregular self-motion paths and singularity are then provided.

4.1. Bifurcation structure

For RIKC problems addressed here, the disjoint self-motion manifolds are open manifolds, each bounded by kinematic configurations associated with a twist alignment. Consider self-motion manifolds $\boldsymbol{q}_i(\psi)$ and $\boldsymbol{q}_j(\psi)$ that are bounded by different pairs of twist alignment cases, but with a shared case (e.g., $\boldsymbol{q}_1(\psi)$ and $\boldsymbol{q}_2(\psi)$ in Fig. 4b). As the task progresses, these disjoint self-motion manifolds converge/split at a coregular value. The bifurcation structure, associated with two independent open coregular self-motion paths, is illustrated by the RIKC solution space shown in Fig. 5a. The bifurcation branch roadmap for this structure is shown in Fig. 5b. In the RIKC solution space, all self-motion manifolds (solid lines) to the left of the dashed-dotted plane correspond to self-motion manifolds $\boldsymbol{q}_i(\psi)$, and those to the right correspond to $\boldsymbol{q}_j(\psi)$, where *i* and *j* denote the identification numbers. Two independent open coregular self-motion paths exist, one with kinematic self-motion (dashed line) and one without kinematic self-motion (dashed-dotted line). The intersection point of these coregular self-motion paths is the singularity (×). Each coregular self-motion path is separated into two paths at the singularity; these are bifurcation branches (black nodes) in the bb-roadmap. This bifurcation structure has four branches per singularity, whereas the structure associated with closed manifolds in Fig. 2a has two branches per singularity.

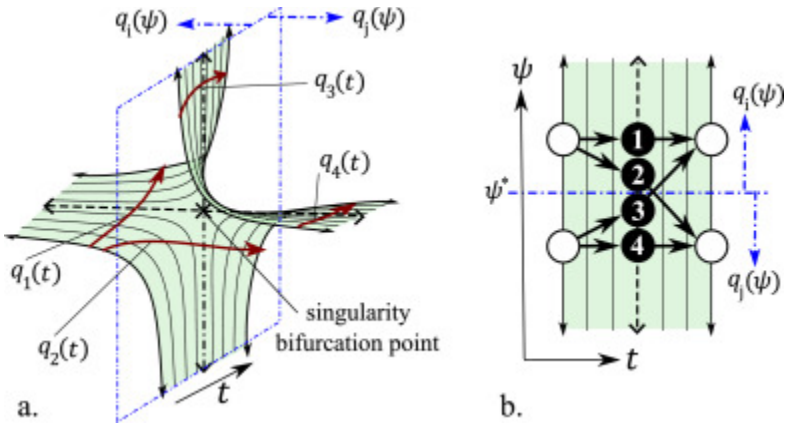

Fig. 5. Bifurcation structure. (a) RIKC solution space near a bifurcation point associated with a singularity. Solid lines are regular self-motion manifolds, dashed and dashed-dotted lines are coregular self-motion paths, and their intersection (×) is the bifurcation point. Joint paths (solid lines with arrow endpoints) are in different homotopy classes. The other self-motion manifolds have their own non-bifurcating RIKC solution regions (not shown). (b) Rotated top view of Fig. 5a such that kinematic self-motion is in the vertical direction and time *t* is in the horizontal direction. The corresponding bb-roadmap is superimposed on the surface. Each joint path $\boldsymbol{q}_k(t)$ crosses a different bifurcation branch associated with a numbered black node.

Fig. 5 a shows four joint paths (solid lines with arrow endpoints). Two paths start at one regular self-motion manifold $\boldsymbol{q}_i(\psi)$, and two paths start at a different regular self-motion manifold $\boldsymbol{q}_j(\psi)$. The manipulator's joint configuration can switch from being in a self-motion manifold $\boldsymbol{q}_i(\psi)$ to $\boldsymbol{q}_j(\psi)$ (or $\boldsymbol{q}_j(\psi)$ to $\boldsymbol{q}_i(\psi)$) only if the joint path crosses the vertical dashed-dotted line corresponding to a coregular self-motion path for which there is no kinematic motion. Each of the joint paths shown has a different combination of start and terminal self-motion manifold identification numbers and is in a different homotopy class. For each case, the homotopy class of the joint path is identified by the coregular self-motion branch that the path crosses.

4.2. Identifying coregular values

The strategy for identifying coregular values is based on investigating twist alignment cases and the compliance realization conditions for a simpler manipulator with fewer joints.

The task compliance of a serial manipulator with n_p compliant joints is given by

(9)

$$\mathbf{C}_x = q_{c_i}\mathbf{t}_i\mathbf{t}_i^T + q_{c_j}\mathbf{t}_j\mathbf{t}_j^T + q_{c_k}\mathbf{t}_k\mathbf{t}_k^T + \cdots + q_{c_z}\mathbf{t}_z\mathbf{t}_z^T,$$

where $i, j, k, \ldots, z$ are joints $1,2,3, \ldots, n_p$ in no particular order.

If twists *i* and *j* align, (7) is substituted into (9), yielding

(10)

$$\mathbf{C}_x = (q_{c_i} + \alpha^2 q_{c_j})\mathbf{t}_i\mathbf{t}_i^T + q_{c_k}\mathbf{t}_k\mathbf{t}_k^T + \cdots + q_{c_z}\mathbf{t}_z\mathbf{t}_z^T,$$

where $\mathbf{t}_k \ldots \mathbf{t}_z$ are the independent twists and

(11)

$$q_{c_c} = \left(q_{c_i} + \alpha^2 q_{c_j}\right)$$

is the combined compliance of joints *i* and *j* as seen from the twist associated with joint-*i*.

The task compliance expressed in (10) can be realized by a simpler manipulator with one less joint. The simpler manipulator has separate, more limiting, compliance realization conditions.

The twist alignment cases that can realize a task configuration on the task manipulation path are identified by searching (with root finding or optimization) for instances that satisfy the compliance realization conditions of the simpler manipulator over the task manipulation path ***x***(*t*). The compliance realization conditions are evaluated using the inverse kinematic solutions of each twist alignment case of the redundant manipulator. Each case may have 0, 1, or multiple task instances for which the compliance realization conditions are satisfied.

These steps used in identifying the coregular values of the 3R manipulator performing a particle planar task in the kinematic dexterous workspace are described below.

3R-VSA manipulator
The particle-planar task compliance of a 3R-VSA manipulator end-effector is given by (9) where *i, j*, and *k* are joints 1, 2, and 3 in no particular order.

For a singularity to occur, two twists must align and the third twist must satisfy an additional requirement. Each singularity case must satisfy (10) and must satisfy a 2R compliance realization condition. For example, the 3R manipulator with an elbow-up pose and twist alignment case b^- in Fig. 6a has the same compliant behavior as the 2R manipulator in Fig. 6b.

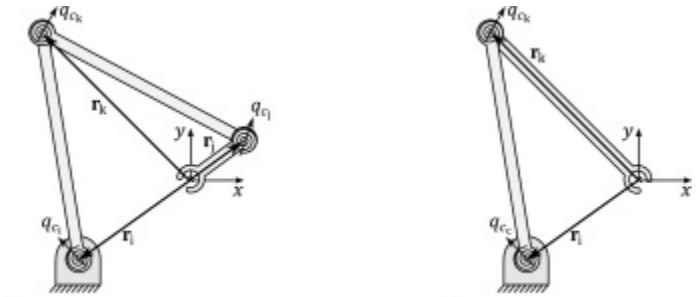

Fig. 6. Twist alignments. (a) 3R-VSA manipulator at twist alignment case b^-. (b) 2R-VSA manipulator capable of achieving the same particle elastic behavior.

According to the compliance realization condition in [8], the 2R-VSA manipulator can achieve a specified task compliance $\mathbf{C}_x$ if and only if

(12)

$$\mathbf{r}_i^T \mathbf{C}_x \mathbf{r}_k = 0.$$

The calculated [8] joint compliances q_{c_c} and q_{c_k} for the 2R realization are unique. The compliance values for the 3R manipulator at this kinematic configuration are not unique. At these configurations a self-motion path exists in the compliance subspace.

The coregular values on the task manipulation paths are readily obtained by finding the roots of (12) over $\boldsymbol{x}(t)$ where $\mathbf{r}_i$ and $\mathbf{r}_k$ are defined by the unique inverse kinematic solutions of the twist alignment cases.

4.3. Coregular self-motion without kinematic motion

At each coregular value, there exists a normal self-motion path $\boldsymbol{q}(\psi)$ (described in Section 3) with motion in the combined kinematic and compliance space and there exists another self-motion path $\boldsymbol{q}(\psi_c)$ for which there is no kinematic self-motion. These paths are coregular self-motion paths intersecting at the bifurcation singularity. Coregular self-motion paths exist when a twist alignment occurs (7) and the compliance realization conditions of the effective simpler manipulator (e.g., condition (12) for the 2R manipulator) are satisfied.

Joint compliances of the twist-aligned joints, q_{c_i} and q_{c_j} can linearly trade-off their value while keeping q_{c_c} constant. Let $\psi_c \in (-\infty, \infty)$ be the *compliance self-motion parameter* that resolves this trade-off:

(13)

$$q_{c_i} = \psi_c q_{c_c}$$

and

(14)

$$q_{c_j} = \frac{(1-\psi_c)}{\alpha^2} q_{c_c}.$$

The self-motion path $\boldsymbol{q}(\psi_c)$ is defined by the task configuration $\mathbf{x}$, the twist alignment case, and ψ_c. The compliances of the aligned twists q_{c_i}, q_{c_j} are resolved by ψ_c using (13) and (14). The self-motion path $\boldsymbol{q}(\psi_c)$ is linear in joint space.

The bounds of the self-motion parameter ψ_c are identified by substituting joint limits into (13) and (14). The self motion parameter bounds due to the compliance limits of joint-*i* and joint-*j* are

(15)

$$\left[\psi_{c_{i_{\min}}}, \psi_{c_{i_{\max}}}\right] = \left[\frac{q_{c_{i_{\min}}}}{q_{c_c}}, \frac{q_{c_{i_{\max}}}}{q_{c_c}}\right]$$

And

(16)

$$\left[\psi_{c_{j_{\min}}}, \psi_{c_{j_{\max}}}\right] = \left[1-\alpha^2\frac{q_{c_{j_{\max}}}}{q_{c_c}}, 1-\alpha^2\frac{q_{c_{j_{\min}}}}{q_{c_c}}\right],$$

respectively.

The bounds of the feasible self-motion path is given by

(17)

$$[\psi_{c_{\min}}, \psi_{c_{\max}}] = \left[\max\left(\psi_{c_{i_{\min}}}, \psi_{c_{j_{\min}}}\right), \min\left(\psi_{c_{i_{\max}}}, \psi_{c_{j_{\max}}}\right)\right].$$

A feasible self-motion path exists if $\psi_{c_{\min}} < \psi_{c_{\max}}$,, and if the other joint coordinate values $(\mathbf{q}_p, q_k, \ldots, q_z)$ are all feasible.

4.4. Identifying the singularity

The coregular self-motion $\boldsymbol{q}(\psi)$ intersects $\boldsymbol{q}(\psi_c)$ at ψ^* when (7) is satisfied. The compliance configuration, normally resolved over ψ by the compliance synthesis formulas are not valid at the twist alignment case. They are, however, valid at ψ values a small step to either side of ψ^*. The compliance configurations resolved on either side of ψ^* are used to approximate the intersecting joint configuration on $\boldsymbol{q}(\psi_c)$. The singular configuration is then identified by maximizing the condition number of the total Jacobian matrix over ψ_c. The singularity may or may not be a feasible joint configuration. If, the singularity is not a feasible joint configuration the bifurcation does not occur at the coregular value. Instead, the bifurcation results from a joint limit at a task time before (or after) the coregular value.

5. Bifurcations from joint limits

To characterize the connectivity of the RIKC solution space, bifurcation nodes associated with joint limits must be identified. This section describes a method to quickly identify bifurcation points (and premature termination points) resulting from joint limits in RIK and RIKC solution spaces with one degree of redundancy. The method involves using a *boundary edge path planner*, an instantaneous path planner that tracks the boundary edge of the feasible RIK(C) solution space.

The boundaries of the feasible RIK(C) solution space result from truncation by a joint limit, as illustrated in Fig. 7a, by the joint-limit hyperplane (dashed-dotted plane) truncating the self-motion manifolds (solid lines). Fig. 7b illustrates *conceptually*[3] a feasible RIKC solution space with several truncated regions, and the bifurcation × at t_1 corresponds to the bifurcation in Fig. 7a. Each continuous and smooth boundary edge (edge of the shaded region that is not a dotted line) of the feasible RIK(C) solution space corresponds to saturation of a single joint variable. Each path (thick solid line with arrows) tracking the boundary edge of the feasible solution space is generated from a starting configuration (circle dot) forward and backward in time (arrows indicate the direction of path generation). The starting configurations are boundary configurations of previously identified feasible self-motion paths (dotted lines). Each velocity-based generated path is generated forward in time until normalized time $t = 1$ (or backward in time until $t = 0$), or until the path can no longer be generated without leaving the boundary edge. If the path cannot be generated further, the final joint configuration of the path is either a bifurcation point (×) or a premature termination point (⊗). The boundary edge path planner and the selection of starting points are described below.

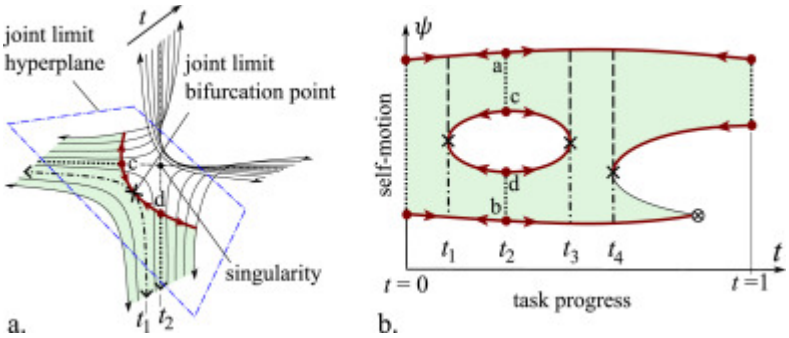

Fig. 7. Boundary edges from joint limits. (a) Joint limit hyperplane truncating the RIK(C) solution space (shaded surface is the feasible region). (b) An RIK(C) solution space with several truncated regions. Thick solid lines are instantaneously generated paths starting from boundary configurations (circles) of previously identified self-motion paths (dotted lines).

5.1. Boundary edge path planner

The boundary edge path planner is based on the SNS algorithm and involves the integration of (3) with inequality constraints (4) for joint limits. The SNS algorithm is typically formulated for the weighted pseudoinverse with an identify weighting matrix: $\mathbf{W} = \mathbf{I}$ (i.e., the Moore-Penrose pseudoinverse). However, passive compliance control uses a non-identity, configuration dependent weighting matrix to identify the optimal actuator velocity (primary and VSA) [3]. The SNS algorithm is modified to account for any positive definite weighting matrix, using

(18)

$$\mathbf{J}_{\text{SNS}}^{\dagger} = \mathbf{S}\mathbf{W}^{-\frac{1}{2}}\left(\mathbf{J}\mathbf{W}^{-\frac{1}{2}}\mathbf{S}\right)^{\#}.$$

A joint path along the boundary edge is generated using the SNS algorithm with *pre-saturation*. Normal implementation of the SNS algorithm initializes with all joints non-saturated ($\mathbf{S} = \mathbf{I}$). For boundary tracking, the joint path starts at a boundary configuration where joint variable *i* is saturated. Joint variable *i* is locked at its boundary value by initializing $\mathbf{S}$ with $S_{ii} = 0$ and setting the i^{th} value of $\dot{\mathbf{q}}_N$ to zero. The boundary path is then generated by integrating (3) with $\mathbf{J}^{\dagger}$ replaced by (18).

When the task path $\boldsymbol{x}(t)$ is continuous and differentiable, the boundary edge path planner terminates at the task endpoint, or terminates prematurely. The condition for termination is that the rank of the saturated Jacobian matrix is smaller than the dimension of the task, i.e., $\text{rank}(\mathbf{JS}) < m$. This occurs in the following two cases:

1. If $\text{rank}(\mathbf{S}) < m$,, the termination point is the intersection point of two RIKC boundary edges.
2. If $\text{rank}(\mathbf{S}) = m$,, the termination point is a joint limit tangent point where a self-motion manifold tangentially encounters a boundary edge path.

Case 1 always corresponds to a premature termination point. Case 2 corresponds to either a bifurcation point or a premature termination point.

If the boundary edge path planner terminates at a time other than the task start or task end, the terminal point $\mathbf{q}_f$ is either a premature termination point or a joint limit bifurcation point. If $\mathbf{q}_f$ has

two saturated joint variables, then rank($\mathbf{S}$) $< m$ and $\mathbf{q}_f$ is a premature termination point. If $\mathbf{q}_f$ has only one saturated joint variable, and a joint path can be generated beyond t_f (starting from the joint configuration generated just before $\mathbf{q}_f$) by unlocking the saturated joint, then $\mathbf{q}_f$ is a bifurcation point.

5.2. Boundary edge start points

The initial joint configuration for each generated boundary edge path is selected from the previously identified feasible self-motion paths of task endpoints (which were identified when generating endpoint nodes, Section 3), and of coregular values (which were identified when searching for singularities, Section 4).

Let $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ be sets of feasible self-motion paths of the task start point and task terminal point, respectively. For each boundary configuration of each feasible self-motion path in $\boldsymbol{\mathcal{S}}$, a boundary tracking joint path is generated forward in time (until $t = 1$). For each boundary configuration of each feasible self-motion path in $\boldsymbol{\mathcal{T}}$, a boundary tracking joint path is generated backward in time (until $t = 0$). In Fig. 7b, $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ each have a single feasible self-motion path at $t = 0$ and $t = 1$, respectively. The boundary edge joint path generated from the lower boundary configuration in $\boldsymbol{\mathcal{S}}$ terminates at $\otimes$. The boundary edge joint path generated from the lower boundary configuration in $\boldsymbol{\mathcal{T}}$ terminates at × at t_4. A boundary tracking joint path generated from the upper configuration in $\boldsymbol{\mathcal{S}}$ terminates at the upper configuration in $\boldsymbol{\mathcal{T}}$.

Consider the bifurcation points × at t_1 and t_3 associated with the "hole" (closed path associated with the saturation of a single joint) in Fig. 7b. To find them with the boundary edge path planer, a starting configuration on a feasible self-motion at some time t_2 between the bifurcation points of the "hole" must be identified. For the RIKC problems, "holes" span coregular values for which the feasible coregular self-motion paths (identified in Section 4) are between joint limit bifurcation points. Let $\boldsymbol{\mathcal{C}}$ be the set of feasible coregular self-motion paths. For each boundary configuration of each feasible self-motion path in $\boldsymbol{\mathcal{C}}$, a boundary edge path is generated forward in time and backward in time. Boundary tracking paths starting from the top and bottom configuration at t_2 in Fig. 7b (points a and b) terminate at boundary configurations in $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$. Boundary tracking paths starting from the remaining boundary configurations at t_2 (points c and d) terminate at × at t_1 for backward generated paths and terminate at × at t_3 for forward generated paths (not seen in Fig. 7a).

For all boundary tracking paths generated from the boundary configurations in $\boldsymbol{\mathcal{S}}$, $\boldsymbol{\mathcal{T}}$, and $\boldsymbol{\mathcal{C}}$, the path's termination case is checked, using the methods described in Section 5.1, to determine if the endpoint corresponds to a task endpoint, premature termination point, or joint limit bifurcation point. Multiple boundary edge paths may have the same terminal configuration. The set of unique premature termination points is identified and the set of unique joint limit bifurcation points is identified. For each joint limit bifurcation point, the feasible self-motion path touching the bifurcation point is generated. These self-motion paths are split into two branches (e.g., dashed and dashed-dotted lines in Fig. 7) and are used as nodes in the bb-roadmap.

6. Bifurcation branch roadmap construction

The final step in characterizing the connectivity of the RIKC solution space is constructing the bifurcation branch roadmap (bb-roadmap). Means of identifying the nodes (feasible self-motion paths)

were described in Sections 3–5. Additional node characteristics are described below. The remaining challenge in constructing the bb-roadmap is identifying edges that correspond to homotopy classes of joint paths connecting feasible self-motion paths nodes. This section describes a fast method to identify the bb-roadmap edges using the self-motion manifold identification numbers. The self-motion manifold identification numbers are assigned in a consistent manner for the entire RIKC solution space (described in Section 3).

6.1. Roadmap node characteristics

The nodes of the bifurcation branch roadmap are either feasible self-motion paths of task endpoints or bifurcation branches, i.e., each a feasible self-motion path on one side of a bifurcation point. There are subtle differences in the characteristics of the self-motion paths introduced in this paper. These differences are reviewed below using examples illustrated in Figs. 5a and 7a.

- A *self-motion path* is a path in the RIKC solution space for which the end-effector configuration does not change. Feasibility is not considered.
- A *coregular self-motion path* is a self-motion path at a coregular value (e.g., dashed and dashed-dotted lines in Fig. 5a and thin and thick dotted lines at $t = t_2$ in Fig. 7a).
- A *regular self-motion path* is a self-motion path at a regular value (e.g., thin solid lines).
- A *feasible self-motion path* is a (regular or coregular) self-motion path in the feasible RIKC solution space (paths completely on the shaded (feasible) surface. For example, the feasible coregular self-motion paths in Figs. 5a and 7 a are the dashed and dashed-dotted lines in Fig. 5a and thick dotted lines in Fig. 7a).
- A *bifurcation branch* is a portion of a feasible self-motion path adjacent to a bifurcation point. A singularity bifurcation point has coregular bifurcation branches (e.g., the portions of the dashed and dashed-dotted lines left, right, above, and below the × in Fig. 5a). A joint limit bifurcation point has regular bifurcation branches (e.g., the dashed and dashed-dotted lines to either side the × in Fig. 7a).

The feasible self-motion paths in $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ correspond to endpoint nodes (e.g., white nodes in Fig. 3b) of the bb-roadmap. Let $\boldsymbol{\mathcal{B}}_{cr}$ be the set of coregular bifurcation branches identified in Section 4; these correspond to singularity bifurcation nodes (black nodes) of the bb-roadmap. Let $\boldsymbol{\mathcal{B}}_{r}$ be the set of regular bifurcation branches identified in Section 5; these correspond to joint limit bifurcation nodes (gray nodes) of the bb-roadmap. Let $\boldsymbol{\mathcal{P}}$ be the set of premature termination points identified in Section 5; these correspond to premature termination nodes (⊗ nodes) in the bb-roadmap.

The feasible self-motion paths in $\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{B}}_{cr}, \boldsymbol{\mathcal{B}}_{r}, \boldsymbol{\mathcal{P}}$ are sorted by time into the set of all nodes of the bifurcation branch roadmap $\boldsymbol{\mathcal{N}}$. There are multiple nodes with the same time value (e.g., two at each joint limit bifurcation point, four at each singularity bifurcation point).

Additional nodes are used in edge generation. When a joint limit bifurcation occurs, the bifurcation branches are not the nearby feasible coregular self-motion paths (e.g., the thick dotted lines in Fig. 7a). These coregular self-motion paths are needed, however, for edge identification. Let $\boldsymbol{\mathcal{C}}_{f}$ be the set of feasible coregular self-motion paths (that are not bifurcation branches) that are needed for identifying roadmap edges.

6.2. Roadmap edges

Because the RIKC solution space consists of open self-motion manifolds, there is exactly one homotopy class of joint paths connecting two nodes (not the case for closed self-motion manifolds [5]). A brute-force approach for identifying the edges is to attempt to generate a joint path between every two node combination. This would involve using the bi-directional instantaneous path planner [5] and checking if the generated joint path crosses the self-motion paths of any other nodes.

The edges of the bb-roadmap can be determined much faster by investigating the connectivity of the self-motion manifolds using the self-motion manifold identification numbers and the bifurcation structure. Consider first, an RIKC problem without joint limits. An edge between nodes A and B exists if a self-motion path $\boldsymbol{q}^{+}(\psi)$ immediately following a node A $\in \boldsymbol{\mathcal{N}}$ deforms continuously along the feasible RIKC solution space until it reaches a self-motion path $\boldsymbol{q}^{-}(\psi)$ immediately preceding a later node B $\in \boldsymbol{\mathcal{N}}$ (a later bifurcation node or task endpoint node).

Given node A at time t_A, node B is selected from a subset of $\boldsymbol{\mathcal{N}}$ based on the following: 1) $t_B > t_A$ and 2) $\mathrm{Id}_{\mathrm{num}}\left(\boldsymbol{q}_B^{-}(\psi)\right) = \mathrm{Id}_{\mathrm{num}}\left(\boldsymbol{q}_A^{+}(\psi)\right)$, where $\mathrm{Id}_{\mathrm{num}}(\cdot)$ is the identification number. Edges are generated to the nearest nodes (in time) in this subset.

This roadmap construction process is illustrated in Fig. 8 for a simple case without joint limits. Fig. 8a shows a simplified representation of an RIKC solution space with 4 self-motion manifolds (identification numbers 1–4) at each time instance. The four manifolds are separated by 4 twist alignment cases (dashed lines) on a closed kinematic self-motion manifold that are "cut and unrolled" into a flat surface in Fig. 8a parameterized by ψ. The space contains 3 bifurcations from singularities at coregular values, each with a structure like that shown in Fig. 5. The bb-roadmap of the RIKC solution space near each singularity is overlayed on the surface to indicate the connectivity. The bifurcation branches (black nodes) of each singularity are in sets of four. Each of the four branches has self-motion manifolds $[\boldsymbol{q}^{-}(\psi), \boldsymbol{q}^{+}(\psi)$ with a unique combination of self-motion manifold identification numbers (indicated by the white numbered nodes with dashed edges). The bb-roadmap of this RIKC solution space is given in Fig. 8b.

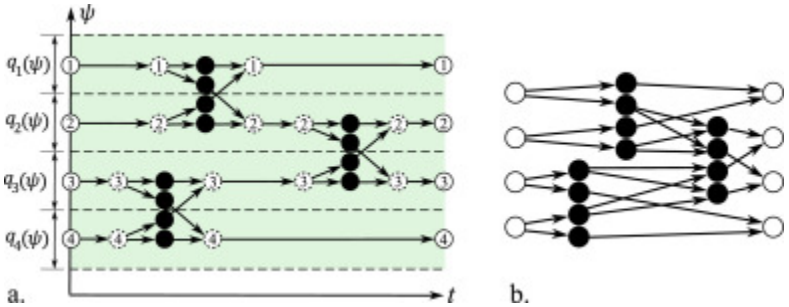

Fig. 8. RIKC bifurcation branch roadmap construction. (a) Simplified representation of the RIKC solution space parameterized by *t* on the horizontal axis and the kinematic self-motion parameter ψ on the vertical axis. (b) Bifurcation branch roadmap.

The roadmap construction process is slightly modified when joint limits exist. Joint limit bifurcations cause multiple disjoint feasible self-motion paths to exist in the same self-motion manifold. Although the feasible self-motion paths are no longer complete manifolds, they are assigned the same

identification number as the manifolds they are in. An identification number reassignment procedure is used to distinguish disjoint feasible self-motion paths in the same manifold. Also, because of joint limit bifurcations, some feasible coregular self-motion paths are not in $\boldsymbol{\mathcal{N}}$, but are in $\boldsymbol{\mathcal{C}_f}$. These coregular self-motion paths must be considered because a continuously deforming feasible self-motion path that crosses a feasible coregular self-motion path without kinematic self-motion (vertical dotted line in Fig. 7a) changes from one self-motion identification number to another by crossing the twist alignment case.

Reassignment of the identification number is only required for joint limit bifurcation branches (nodes in $\boldsymbol{\mathcal{B}_r}$) and for the feasible self-motion paths they deform into. If the bifurcation (splitting) occurs as time progresses forward, the $\boldsymbol{q}^{+}(\psi)$ self-motion identification number is reassigned a unique number *k*. Each of these feasible self-motion paths (say of Node A ∈ $\boldsymbol{\mathcal{N}}$) continuously deforms into a feasible self motion path of a node B ∈ $\{\boldsymbol{\mathcal{N}}, \boldsymbol{\mathcal{C}_f}\}$ Node B is identified using the boundary edge path planner. The self-motion identification number of $\boldsymbol{q}^{-}(t)$ of B is reassigned as *k*. A similar process is used when the bifurcation occurs as time progresses backward.

The edge generation methodology is summarized in the following algorithmic procedure. For each node in $\boldsymbol{\mathcal{N}}$ and not in $\boldsymbol{\mathcal{T}}$, roadmap edges are identified using the following steps:

1. Denote the selected node as I, its task time t_i, and its forward self-motion $\boldsymbol{q}_i^{+}(\psi)$, where *i* is its self-motion identification number.
2. Identify all the nodes in $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{C}_f}$ with time $t > t_i$ each having a backward self-motion $\boldsymbol{q}_{\mathrm{j}}^{-}(\psi)$ | $j = i$.
3. From the set in Step 2, identify those with the smallest time $t > t_i$ (there can be multiple). Let $t_{\mathcal{C}_f} > t_i$ be the smallest time of the relevant nodes in $\boldsymbol{\mathcal{C}_f}$; and let $t_{\mathcal{N}} > t_i$ be the smallest time of the relevant nodes in $\boldsymbol{\mathcal{N}}$.
4. If $t_{\mathcal{C}_f} < t_{\mathcal{N}}$,, a coregular self-motion exists between the edge nodes. Because a twist alignment case can be crossed at $t_{\mathcal{C}_f}$, the forward self-motion $\mathbf{q}_i^{+}(\psi)$ identification number *i* is changed to that of the node associated with $t_{\mathcal{C}_f}$, t_i is updated to $t_{\mathcal{C}_f}$, and Steps 2–3 are repeated.
5. If $t_{\mathcal{N}} < t_{\mathcal{C}_f}$ (or if there are no relevant nodes in $\boldsymbol{\mathcal{C}_f}$ from Step 2), there is an edge from I to each of the relevant nodes in $\boldsymbol{\mathcal{N}}$ with the time value $t_{\mathcal{N}}$.

The resulting directed graph fully captures the connectivity of the RIK solution space. This combined kinematic and compliance solution space bb-roadmap is used together with the bifurcation branch algorithm [5] to identify the globally optimal solution to the RIKC path planning problem.

7. Case study

This section demonstrates the algorithm for generating the bifurcation branch roadmap and obtaining the globally optimal joint path for a 3R-VSA planar manipulator. Two different task manipulation paths are considered. The motion path is the same in each; only the task compliance is different. Although the tasks are relatively simple, the connection structure of the two RIKC solution spaces are relatively complex and quite different from each other.

7.1. Manipulator and task description

Consider the 3R-VSA manipulator and particle planar compliance task illustrated in Fig. 9. The link lengths, normalized by the reach of the manipulator *L*, are: $l_1 = 0.46$, $l_2 = 0.43$, and $l_3 = 0.11$ (anthropomorphic ratios [27]). The kinematic joint positions q_{p_1}, q_{p_2}, and q_{p_3} are expressed in radians; the joint compliances q_{c_1}, q_{c_2}, and q_{c_3} are normalized by $(f_g L)^{-1}$ where f_g is the weight of the manipulator.

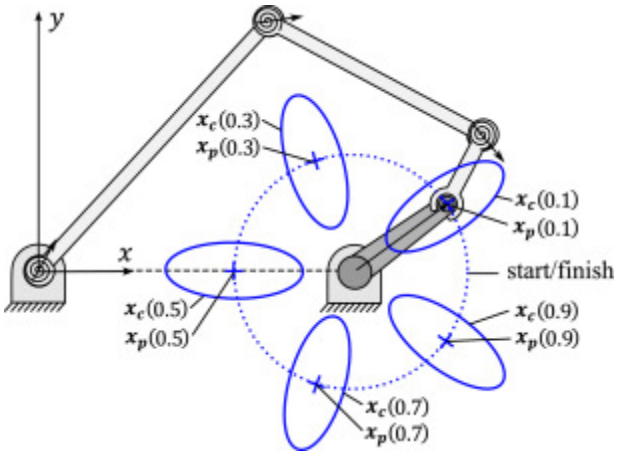

Fig. 9. 3R-VSA manipulator performing a crank-turning task with an unknown work load. Five equally spaced instants in time illustrate the continuous end-effector path $\boldsymbol{x}_p(t)$ and the continuous compliance path $\mathbf{x}_c(t)$.

For 3R-VSA manipulators performing particle-planar compliance tasks, the total joint configuration is

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

and the total task configuration is

$$\mathbf{x} = \left[x, y, C_{x_{11}}, C_{x_{12}}, C_{x_{22}}\right]^T.$$

The constrained manipulation task is to turn a crank. The end-effector motion path is given by

(19)

$$\boldsymbol{x}_p(t) = \begin{bmatrix} 0.5 + 0.2\cos(2\pi t) \\ 0.2\sin(2\pi t) \end{bmatrix},$$

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

Since the crank handle has a fixed radial distance from the crank center, the task compliance in the direction along the crank arm is high to limit the constraint force that may result from error in the commanded relative position of the end-effector. The task compliance in the direction of crank motion is low to limit the end-effector deflection from the planned path due to the crank work-load. In Fig. 9, the desired compliance ellipse[4] is shown at five instants in time, where the major and minor radii of the ellipse are eigenvalues of the task compliance matrix.

The normalized task compliance path (task compliance normalized by Lf_g^{-1}) in matrix form is

(20)

$$\mathbf{C}_x(t) = \mathbf{U}\begin{bmatrix} \frac{\lambda_2}{\gamma(t)} & 0 \\ 0 & \lambda_2 \end{bmatrix}\mathbf{U}^T$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2]$ are the normalized eigenvectors, $\lambda_2 = 0.1$ is the larger eigenvalue and $\gamma = \frac{\lambda_2}{\lambda_1}$ is the ellipse aspect ratio. The eigenvalues are constant in each task, but the eigenvectors change so that the orientation of the larger compliance eigenvector is always along the crank arm:

(21)

$$\mathbf{U} = \begin{bmatrix} -\sin(2\pi t) & \cos(2\pi t) \\ \cos(2\pi t) & \sin(2\pi t) \end{bmatrix}.$$

Each joint compliance is controlled using a VSA with an exponential compliance versus actuation profile. The feasible compliance range is $q_{c_i} \in [0.001, 10]$ corresponding to the actuator limits $\phi_i \in [0, \pi/2]$. The VSA actuator position is related to the joint compliance by: $\phi_{c_i} = \frac{1}{\xi}\ln\left(\frac{q_{c_i}}{c_0}\right)$, where $\xi = 5.86$ and $c_0 = 0.001$. The total actuator configuration is $\boldsymbol{\phi} = \left[\boldsymbol{\phi}_p^T, \phi_{c_1}, \phi_{c_2}, \phi_{c_3}\right]^T$, where $\boldsymbol{\varphi}_p$ is a 3-vector of primary motion positions. The manipulator is oriented in the horizontal plane where gravity does not affect the equilibrium joint position such that $\boldsymbol{\phi}_p = \mathbf{q}_p$.

The globally optimal path is the one that minimizes the actuator velocities, i.e., minimizes

(22)

$$G_{\text{global}} = \int \frac{1}{2}\dot{\boldsymbol{q}}(t)^T \mathbf{W}\big(\boldsymbol{q}(t)\big)\dot{\boldsymbol{q}}(t)dt,$$

where $\mathbf{W}(\cdot)$ is a joint configuration dependent weighing matrix, selected such that $\| \dot{\boldsymbol{\phi}} \|^2 = \dot{\mathbf{q}}^T\mathbf{W}\dot{\mathbf{q}}$:

(23)

$$\mathbf{W}(\mathbf{q}) = \text{diag}\left(\left[1,1,1,\left(\frac{1}{\xi q_{c_1}}\right)^2,\left(\frac{1}{\xi q_{c_2}}\right)^2,\left(\frac{1}{\xi q_{c_3}}\right)^2\right]\right),$$

where diag(·) sorts the vector elements into the diagonal elements of a diagonal matrix and $\frac{1}{\xi q_{c_i}}$ is the sensitivity of the VSA actuator position to variation in the compliance value of joint *i*.

7.2. Results

Results of the "elbow-up" poses are shown for two aspect ratios: $\gamma = 2$ and $\gamma = 10$. The bb-roadmaps for these tasks are shown in Figs. 10a and 11a, respectively. The routes through the bb-roadmaps identify the different joint path homotopy classes. For Task $\gamma = 2$, there are 9 routes that complete the task, of which, 5 are cyclic (listed in Table 1). For Task $\gamma = 10$, there are 120 routes that complete the task, of which, 30 are cyclic.

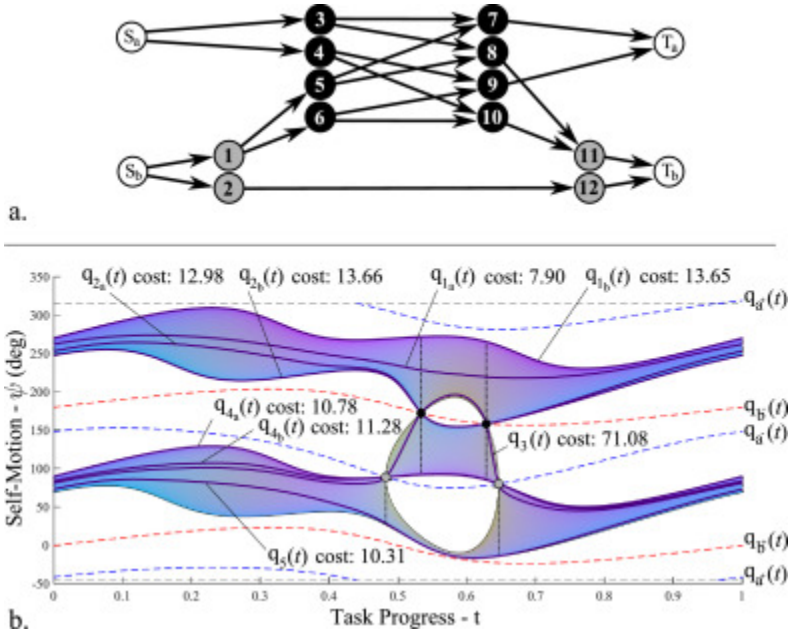

Fig. 10. RIKC solution space connection structure for Task $\gamma = 2$. (a) Bifurcation branch roadmap. (b) Sampled surface parameterized by *t* on the horizontal axis and the kinematic self-motion parameter ψ on the vertical axis.

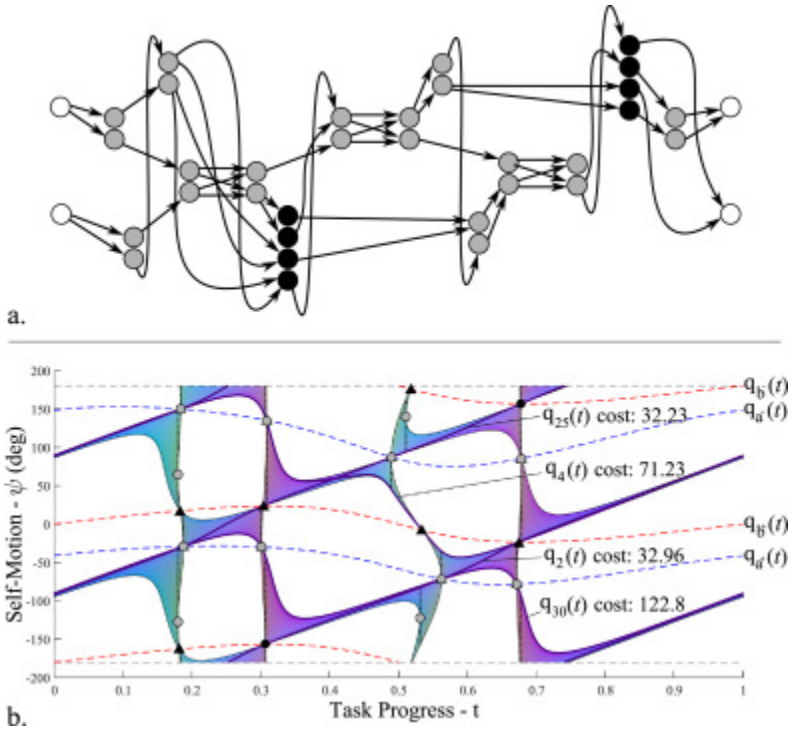

Fig. 11. RIKC solution space connection structure for Task $\gamma = 10$. (a) Bifurcation branch roadmap. (b) Sampled surface parameterized by *t* on the horizontal axis and the kinematic self-motion parameter ψ on the vertical axis.

Table 1. Cyclic homotopy classes for crank task $\gamma = 2$.

Cyclic homotopy class	**Node sequence in Fig. 10a**
1	S_a, 3, 7, T_a
2	S_a, 4, 9, T_a
3	S_b, 1, 5, 8, 11, T_b

4	S_b, 1, 6, 10, 11, T_b
5	S_b, 2, 12, T_b

If the crank needs to be turned a number of times, periodic boundary conditions are used. For Task $\gamma = 2$, the upper/lower bound culling method of the bb-algorithm [5] identified that only Homotopy Class 1 could contain the globally optimal cyclic joint path. However, for comparison purposes, all cyclic homotopy classes are presented for this task. Some homotopy classes had two locally optimal cyclic joint paths.

For Task $\gamma = 10$, the upper/lower bound culling method reduced the set of promising homotopy classes to four. Each homotopy class had a single locally optimal cyclic path.

The locally optimal cyclic paths found using the bb-algorithm are shown on the RIKC solution surfaces in Figs. 10b and 11b as thick solid lines. The cost value of each path, evaluated with (22), is shown in the figures. The globally optimal path cost values were 7.90 for Task $\gamma = 2$ and 32.23 for Task $\gamma = 10$. The actual RIKC solution space is a hyper surface in the 6-dimensional joint space. The surfaces in Figs. 10b and 11b are close representations of the hyper surface unrolled onto a 2-D plane, where normalized time (task progress) is on the horizontal axis and the kinematic self-motion parameter is on the vertical axis. The closed kinematic self-motion manifolds are "cut and unrolled" at the horizontal dashed lines. The curved dashed lines are the kinematic solutions of the twist alignment cases, $\boldsymbol{q}_{a^+}(t)$, $\boldsymbol{q}_{b^+}(t)$, $\boldsymbol{q}_{a^-}(t)$, and $\boldsymbol{q}_{b^-}(t)$, that separate the four (total configuration) self-motion manifolds.

The rasterized colored surface corresponds to feasible joint configurations, where the color of each point on the surface is given by RGB coordinate values corresponding to the compliance actuator configuration: $\boldsymbol{\phi}_c = [\phi_{c_1}, \phi_{c_2}, \phi_{c_3}]^T$ normalized by the actuation limits. Both surfaces have an overall blue hue indicating that the third joint is generally the most compliant.

Black circle markers on the simplified solution space correspond to bifurcation points from singularities. Gray circle markers on the surface correspond to bifurcation points from joint limits. Black triangle markers indicate the locations of feasible coregular self-motion paths $\boldsymbol{q}(\psi_c)$ in the compliance subspace that do not contain the singularity. Coregular paths in the compliance subspace are not visible on the plotted surface (which is parameterized by the *kinematic* self-motion parameter), but they can be imagined as lines going into or coming out of the page at the black circle and triangle marker locations. In some instances, the triangle markers are very close to gray circle markers. Vertical dotted lines are self-motion paths $\boldsymbol{q}(\psi)$ associated with bifurcation branches of the bb-roadmap. Each joint limit bifurcation point has two bifurcation branches. Each singularity bifurcation point has four bifurcation branches. Again, the two branches of $\boldsymbol{q}(\psi_c)$ are not visible on the plotted surface.

Although the two tasks are the same except for the compliance in the direction of crank's motion, the bb-roadmaps and the globally optimal cyclic joint paths are significantly different.

8. Conclusion

This paper provides a method for identifying the RIKC solution space connection structure (recorded in the bifurcation branch roadmap), for manipulation with one degree of redundancy. Procedures for

identifying the number and the structure of each self-motion manifold in the combined kinematic and compliance joint space were provided as well as procedures to identify bifurcations from singularities and from joint limits. The procedures for identifying bifurcations (and premature termination points) from joint limits can be applied to traditional RIK problems without compliance variables. The bifurcation branch roadmap is used in the bifurcation branch algorithm to identify the globally optimal path in the combined kinematic and compliance joint space to achieve the optimal sequence of joint configurations for passive compliance control. The structure of the compliance-extended redundant inverse kinematic (RIKC) solution space is determined by the manipulator structure and the task. Procedures were demonstrated and results were presented for a 3R-VSA manipulator performing particle-planar compliance tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Science Foundation under grant IIS-1427329.

References

[1] B. Vanderborght, A. Albu-Schäffer, A. Bicchi, E. Burdet, D.G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, *et al.* **Variable impedance actuators: a review.** Robot. Auton. Syst., 61 (12) (2013), pp. 1601-1614, 10.1016/j.robot.2013.06.009

[2] A. Albu-Schäffer, M. Fischer, G. Schreiber, F. Schoeppe, G. Hirzinger. **Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints?** 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4 (2004), pp. 3295-3301, 10.1109/IROS.2004.1389925

[3] J.J. Rice, J.M. Schimmels. **Passive compliance control of redundant serial manipulators.** J. Mech. Robot., 10 (4) (2018), p. 041009, 10.1115/1.4039591

[4] J. Burdick. **On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds.** Proceedings of the International Conference on Robotics and Automation, IEEE (1989), pp. 264-270, 10.1109/ROBOT.1989.99999

[5] J.J. Rice, J.M. Schimmels. **Multi-homotopy class optimal path planning for manipulation with one degree of redundancy.** (2020), 10.1016/j.mechmachtheory.2020.103834

[6] H. Höppner, W. Wiedmeyer, P. van der Smagt. **A new biarticular joint mechanism to extend stiffness ranges.** 2014 IEEE International Conference on Robotics and Automation (ICRA) (2014), pp. 3403-3410, 10.1109/ICRA.2014.6907349

[7] F.P. Petit. Analysis and control of variable stiffness robots, Eidgenössische Technische Hochschule ETH Zürich, Nr. 22134 (2014) Ph.D. thesis, Diss.

[8] S. Huang, J.M. Schimmels. **Realization of point planar elastic behaviors using revolute joint serial mechanisms having specified link lengths.** Mech. Mach. Theory, 103 (2016), pp. 1-20

[9] S. Huang, J.M. Schimmels. **Synthesis of point planar elastic behaviors using three-joint serial mechanisms of specified construction.** J. Mech. Robot., 9 (1) (2017), p. 011005, 10.1115/1.4035189

[10] S. Huang, J.M. Schimmels. **Geometric construction-based realization of planar elastic behaviors with parallel and serial manipulators.** J. Mech. Robot., 9 (5) (2017), p. 051006, 10.1115/1.4037019

[11] S. Huang, J.M. Schimmels. **Geometric approach to the realization of planar elastic behaviors with mechanisms having four elastic components.** J. Mech. Robot., 10 (4) (2018), p. 041004, 10.1115/1.4039399

[12] S. Huang, J.M. Schimmels, Geometry based synthesis of planar compliances with redundant mechanisms having five compliant components, Mech. Mach. Theory(Submitted 2018). 10.1016/j.mechmachtheory.2018.12.021

[13] S. Huang, J.M. Schimmels. **Synthesis of planar compliances with mechanisms having six compliant components: geometric approach.** J. Mech. Robot., 12 (3) (2020), 10.1115/1.4045648

[14] S. Huang, J.M. Schimmels. **Geometric construction-based realization of spatial elastic behaviors in parallel and serial manipulators.** IEEE Trans. Robot., 34 (3) (2018), pp. 764-780, 10.1109/TRO.2018.2805315

[15] F. Flacco, A. De Luca, O. Khatib. **Control of redundant robots under hard joint constraints: saturation in the null space.** IEEE Trans. Robot., 31 (3) (2015), pp. 637-654

[16] Y. Nakamura, H. Hanafusa. **Optimal redundancy control of robot manipulators.** Int. J. Robot. Res., 6 (1) (1987), pp. 32-42, 10.1177/027836498700600103

[17] K. Kazerounian, Z. Wang. **Global versus local optimization in redundancy resolution of robotic manipulators.** Int. J. Robot. Res., 7 (5) (1988), pp. 3-12, 10.1177/027836498800700501

[18] D.P. Martin, J. Baillieul, J.M. Hollerbach. **Resolution of kinematic redundancy using optimization techniques.** IEEE Trans. Robot. Autom., 5 (4) (1989), pp. 529-533, 10.1109/70.88067

[19] M. Galicki. **The planning of robotic optimal motions in the presence of obstacles.** Int. J. Robot. Res., 17 (3) (1998), pp. 248-259, 10.1177/027836499801700303

[20] B.J. Martin, J.E. Bobrow. **Minimum-effort motions for open-chain manipulators with task-dependent end-effector constraints.** Int. J. Robot. Res., 18 (2) (1999), pp. 213-224, 10.1177/027836499901800206

[21] A. Guigue, M. Ahmadi, M. Hayes, R. Langlois, F. Tang. **A dynamic programming approach to redundancy resolution with multiple criteria.** Proceedings of the International Conference on Robotics and Automation, IEEE (2007), pp. 375-1380, 10.1109/ROBOT.2007.363176

[22] E. Ferrentino, P. Chiacchio. **On the optimal resolution of inverse kinematics for redundant manipulators using a topological analysis.** J. Mech. Robot., 12 (3) (2020), 10.1115/1.4045178

[23] N.S. Bedrossian. **Classification of singular configurations for redundant manipulators.** Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on, IEEE (1990), pp. 818-823, 10.1109/ROBOT.1990.126089

[24] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, K. Kosuge. **Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution.** IEEE Trans. Robot., 24 (5) (2008), pp. 1131-1142, 10.1109/TRO.2008.2003266

[25] C.L. Luck, S. Lee. **Self-motion topology for redundant manipulators with joint limits.** [1993] Proceedings IEEE International Conference on Robotics and Automation, 3 (1993), pp. 626-631, 10.1109/ROBOT.1993.291835

[26] A. Liégois. **Automatic supervisory control of the configuration and behavior of multibody mechanisms.** IEEE Trans. Syst. Man. Cybern., 7 (12) (1977), pp. 868-871, 10.1109/TSMC.1977.4309644

[27] C.C. Gordon, T. Churchill, C.E. Clauser, B. Bradtmiller, J.T. McConville, I. Tebbetts, R.A. Walker. **Anthropometric survey of us army personnel: Summary statistics, interim report for 1988.** Tech. rep., Anthropology Research Project Inc Yellow Springs OH (1989)

[1]The "structure" of a 1-dimensional self-motion manifold is either open or closed.

[2]Three versions of the SNS algorithm are presented in Flacco et al. [15], a *basic, optimal*, and *fast* version. For RIK(C) problems with one degree of redundancy, the basic and optimal versions yield the same result.

[3]The illustrated surface is an abstract representation of the RIKC solution space, a 2-dimensional subspace of a much higher dimensional space (e.g., the 6 dimensional space of the 3R-VSA manipulator; 3 kinematic and 3 compliant). The 2-dimensional subspace is minimally parameterized using self-motion and time.

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