Multitask Constrained Motion Control Using a Mass-Weighted Orthogonal Decomposition

Vincent De Sapio[1]
Department of Scalable Modeling and Analysis,
Sandia National Laboratories,
Livermore, CA 94550
e-mail: vdesap@sandia.gov

Jaeheung Park
Department of Intelligent Convergence Systems,
Seoul National University,
Suwon 443-270, Korea
e-mail: park73@snu.ac.kr

This paper presents an approach to formulating task-level motion-control for holonomically constrained multibody systems based on a mass-weighted orthogonal decomposition. The basis for this approach involves the formation of a recursive null space for constraints and motion-control tasks onto which subsequent motion-control tasks are projected. The recursive null space arises out of the process of orthogonalizing individual task Jacobian matrices. This orthogonalization process is analogous to the Gram–Schmidt process used for orthogonalizing a vector basis. Based on this mass-weighted orthogonal decomposition, recursive algorithms are developed for formulating the overall motion-control equations. The natural symmetry between task-level dynamics and the dynamics of constrained systems is exploited in this approach. An example is presented to illustrate the practical application of this methodology. [DOI: 10.1115/1.4000907]

1 Introduction

The generalized motion-control of holonomically constrained mechanical systems is of significant importance to a variety of fields including dynamics, control theory, biomechanics, and robotics [1]. Holonomic motion-control involves the specification and execution of motion-control commands in the presence of system constraints [1–3]. These constraints can be expressed as functions solely of the generalized coordinates (holonomic), rather than the generalized velocities. Since the constraints restrict the motion of the system to a constrained motion manifold within configuration space, the specified motion must be consistent with this restricted subset of configuration space. Simple joint space control is problematic since the entire configuration space is assumed accessible and a particular set of arbitrarily chosen joint space coordinates will likely violate the system constraints. A task space control scheme [4,5] avoids this problem since, for redundant systems, a point in task space maps to a self-motion manifold [6] in configuration space. As long as the constrained motion and self-motion manifolds intersect, valid constraint consistent and task consistent solutions exist. In addition to effectively addressing constrained motion-control, task space control allows for a hierarchical specification of multiple motion-control tasks [7,8].

This paper presents a mass-weighted orthogonal decomposition approach to formulating task-level motion-control for holonomically constrained systems. The motion-control is specified as a set of task conditions, or holonomic and rheonomic (explicitly time-dependent) servo constraints [1,9], while the system constraints are imposed. Due to the symmetry between tasks and constraints, the system constraints can be incorporated naturally into the overall task-level dynamics. A recursive null space, associated with the constraints and motion-control tasks, is formulated. Subsequent motion-control tasks are projected onto this null space. The recursive null space arises out of the process of orthogonalizing individual task Jacobian matrices. This is analogous to the Gram–Schmidt process. Based on this mass-weighted orthogonal decomposition, recursive algorithms are presented for formulating the overall motion-control equations. An example is presented to illustrate the practical application of this methodology.

[1]Corresponding author.

Contributed by the Applied Mechanics Division of ASME for publication in the Journal of Applied Mechanics. Manuscript received November 19, 2008; final manuscript received December 10, 2009; published online April 9, 2010. Assoc. Editor: Igor Mezic.

2 Task Space Description

A task space description is used as the basis for the results to be developed throughout this paper. We first review the task space framework for a single task.

2.1 Single Task Case. The operational space framework addresses the dynamics and control of branching chain robots [4,5]. Given a branching chain system the initial step involves defining a set of m task, or operational space, coordinates $\mathbf{x} \in \mathbb{R}^m$. The function $\mathbf{x}(\mathbf{q})$ represents a kinematic mapping from the set of n_q generalized coordinates $\mathbf{q} \in \mathbb{R}^{n_q}$ to the set of task space coordinates. The task space coordinates can represent any function of the generalized coordinates but typically are chosen to describe the set of control coordinates (control output) associated with a motion-control task. Figure 1 (left) illustrates a simple kinematic chain where the task space coordinates are chosen to be the coordinates associated with positioning the terminal point of the chain. Furthermore, by taking the gradient of $\mathbf{x}$, we have the relationship

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

where $\mathbf{J}(\mathbf{q})$ is the Jacobian of $\mathbf{x}$

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} \in \mathbb{R}^{m \times n_q} \tag{2}$$

At this point we can address the dynamics of a system in task space. In the nonredundant case any generalized force can be produced by a task space force $\mathbf{f}$ acting at the task point along the task coordinates. Figure 1 (left) illustrates the action of the task space force for the intuitive case of a Cartesian positioning task. The generalized force (or control torque) $\tau \in \mathbb{R}^{n_q}$ is then composed as $\mathbf{J}^T\mathbf{f}$. In the redundant case an additional term needs to complement the task term in order to realize any arbitrary generalized force. We will refer to this term as the null space term and it can be composed as $\mathbf{N}^T\tau_*$, where τ_* is an arbitrary generalized force and $\mathbf{N}^T$ is the null space projection matrix.

We now express the configuration space equation of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{J}^T\mathbf{f} + \mathbf{N}^T\tau_* = \tau \tag{3}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ is the configuration space mass matrix, $\mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n_q}$ is the vector of centrifugal and Coriolis terms, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{n_q}$ is the vector of gravity terms. We can premultiply Eq. (3) by $\mathbf{J}\mathbf{M}^{-1}$ and rearrange to get

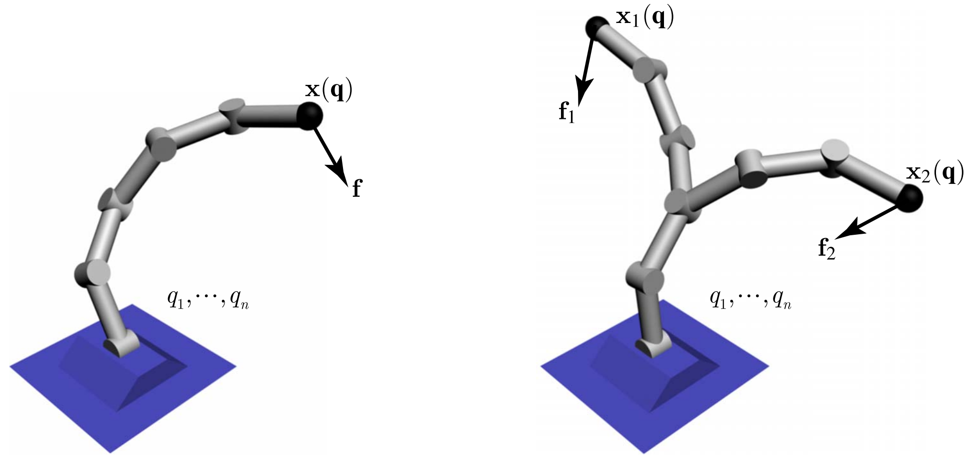

Fig. 1 (Left) A chain with a single task defined. The task space vector $\mathbf{x}=(x,y,z)$ describes the Cartesian position of the terminal point of the chain. The Jacobian J corresponds to this task. The task space force is denoted by f. (Right) A branching chain with two tasks defined. The task space vectors $\mathbf{x}_1=(x_1,y_1,z_1)$ and $\mathbf{x}_2=(x_2,y_2,z_2)$ describe the Cartesian positions of the two independent terminal points.

$$\ddot{\mathbf{x}} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{f} + \mathbf{J}\mathbf{M}^{-1}\mathbf{N}^T\tau_* - \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \mathbf{J}\mathbf{M}^{-1}\mathbf{g} + \dot{\mathbf{J}}\dot{\mathbf{q}} \quad (4)$$

where we note that $\ddot{\mathbf{x}}=\mathbf{J}\ddot{\mathbf{q}}+\dot{\mathbf{J}}\dot{\mathbf{q}}$. We can now impose the condition that the term associated with the null space $\mathbf{N}^T\tau_*$ must not contribute to the task space acceleration. This is referred to as *dynamic consistency* [5] and is expressed as

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{N}^T\tau_* = \mathbf{J}\mathbf{M}^{-1}(\mathbf{1} - \mathbf{J}^T\mathbf{J}^{T\#})\tau_* = \mathbf{0}, \quad \forall\ \tau_* \in \mathbb{R}^{n_q} \quad (5)$$

where $\mathbf{J}^{\#}$ is a generalized inverse of $\mathbf{J}$. We can solve for $\mathbf{J}^{\#}$ under this condition and denote this solution as $\bar{\mathbf{J}}$, the dynamically consistent inverse of $\mathbf{J}$ [5]

$$\bar{\mathbf{J}} = \mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1} \in \mathbb{R}^{n_q\times m} \quad (6)$$

This represents a unique right inverse of $\mathbf{J}$, where by construction, the null space projection matrix $\mathbf{N}^T=\mathbf{1}-\mathbf{J}^T\bar{\mathbf{J}}^T$ is guaranteed not to influence the task acceleration.

We can manipulate Eq. (4) to arrive at

$$\mathbf{f} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\ddot{\mathbf{x}} + (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{g} \quad (7)$$

This expresses the task space (operational space) equation of motion

$$\mathbf{\Lambda}(\mathbf{q})\ddot{\mathbf{x}} + \mu(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{p}(\mathbf{q}) = \mathbf{f} \in \mathbb{R}^m \quad (8)$$

where $\mathbf{\Lambda}(\mathbf{q}) \in \mathbb{R}^{m\times m}$ is the operational space mass matrix, $\mu(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^m$ is the operational space centrifugal and Coriolis force vector, and $\mathbf{p}(\mathbf{q}) \in \mathbb{R}^m$ is the operational space gravity vector. These terms are given by

$$\mathbf{\Lambda}(\mathbf{q}) = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1} \quad (9)$$

$$\mu(\mathbf{q},\dot{\mathbf{q}}) = \bar{\mathbf{J}}^T\mathbf{b} - \mathbf{\Lambda}\dot{\mathbf{J}}\dot{\mathbf{q}} \quad (10)$$

$$\mathbf{p}(\mathbf{q}) = \bar{\mathbf{J}}^T\mathbf{g} \quad (11)$$

$$\bar{\mathbf{J}}^T = \mathbf{\Lambda}\mathbf{J}\mathbf{M}^{-1} \quad (12)$$

Thus, the overall dynamics of our multibody system can be mapped into task space by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \tau \xrightarrow{\bar{\mathbf{J}}^T} \mathbf{f} = \mathbf{\Lambda}\ddot{\mathbf{x}} + \mu + \mathbf{p} \quad (13)$$

In a complementary manner the overall dynamics can be mapped into the task consistent null space (or self-motion space) using $\mathbf{N}^T$.

We can design control for our system in task space coordinates using Eq. (8). Additionally, we can specify the null space behavior of our system with the term $\mathbf{N}^T\tau_*$. The null space control term is guaranteed not to interfere with the task dynamics of Eq. (8) due to the condition of dynamic consistency. This allows for decoupled control design. Finally, the overall control torque applied to the system is composed as in Eq. (3), which is

$$\tau = \mathbf{J}^T\mathbf{f} + \mathbf{N}^T\tau_* \quad (14)$$

A controller employing Eq. (8) would be assumed to have imperfect knowledge of the system. Therefore, Eq. (8) should reflect estimates for the inertial and gravitational terms. Additionally, a control law needs to be incorporated. To this end we replace $\ddot{\mathbf{x}}$ in Eq. (8) with the input of the decoupled system [5] $\mathbf{f}^\star$ to yield the dynamic compensation equation

$$\mathbf{f} = \hat{\mathbf{\Lambda}}\mathbf{f}^\star + \hat{\mu} + \hat{\mathbf{p}} \quad (15)$$

where the $\hat{.}$ represents estimates of the dynamic properties. Any suitable control law can be chosen to serve as input of the decoupled system. In particular, we can choose a linear proportional-derivative (PD) control law of the form

$$\mathbf{f}^\star = \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \ddot{\mathbf{x}}_d \quad (16)$$

2.2 Multitask Case. We now consider a branching chain system for which a set of n tasks $\{\mathbf{x}_1(\mathbf{q}),\cdots,\mathbf{x}_n(\mathbf{q})\}$ and n task forces $\{\mathbf{f}_1,\cdots,\mathbf{f}_n\}$ have been defined. Figure 1 (right) illustrates this. A given task $\mathbf{x}_i \in \mathbb{R}^{m_i}$ has m_i task coordinates. A given task force $\mathbf{f}_i \in \mathbb{R}^{m_i}$ acts at the corresponding task point $\mathbf{x}_i$. Associated with the tasks is a set of n Jacobian matrices $\{\mathbf{J}_1,\cdots,\mathbf{J}_n\}$ given by

$$\mathbf{J}_i = \frac{\partial \mathbf{x}_i}{\partial \mathbf{q}} \in \mathbb{R}^{m_i\times n_q} \quad (17)$$

The task velocities are then given by $\dot{\mathbf{x}}_i=\mathbf{J}_i\dot{\mathbf{q}}$. We will stipulate that the tasks are independently specified, that is, nonconflicting. This requires that the matrix $(\mathbf{J}_1^T\ldots\mathbf{J}_n^T)^T \in \mathbb{R}^{m\times n_q}$ be full rank, where $m=\sum_{i=1}^{n} m_i$.

We will now consider the formulation of task-level motion-control for this multitask case. Our goal is to control the system

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \tau \tag{18}$$

with respect to the set of tasks $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ (control outputs) given a control torque τ (control input). The methodology that we use to achieve this will also be able to seamlessly accommodate constraints, as we shall see later.

This objective of formulating multitask motion-control will be facilitated by an orthogonalization process applied to the task Jacobians. Specifically, given the transposes of the Jacobian matrices $\{\mathbf{J}_1^T, \cdots, \mathbf{J}_n^T\}$, we seek a projection process that can be used to generate a set of mutually orthogonal matrices $\{\mathbf{Q}_1^T, \cdots, \mathbf{Q}_n^T\}$, where $\mathbf{Q}_i \in \mathbb{R}^{m_i \times n_q}$. The specific notion of orthogonality that is relevant here will be defined later. First we will briefly review the concept of orthogonalization in inner product spaces.

2.3 Orthogonalization in Inner Product Spaces. The Gram–Schmidt process represents a standard procedure for orthogonalizing a set of vectors in an inner product space [10,11]. In particular, given a subspace V of the Euclidean space $\mathbb{R}^n$ with some basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$, we can construct an orthonormal basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ for V. First, we make the assignment

$$\mathbf{u}_1 = \frac{1}{\|\mathbf{v}_1\|}\mathbf{v}_1 \tag{19}$$

Subsequent terms are generated as

$$\mathbf{u}_i = \frac{1}{\|\mathbf{v}_i - \text{proj}_{V_{i-1}}\mathbf{v}_i\|}(\mathbf{v}_i - \text{proj}_{V_{i-1}}\mathbf{v}_i) \tag{20}$$

where

$$V_{i-1} = \text{span}(\mathbf{u}_1, \ldots, \mathbf{u}_{i-1}) \tag{21}$$

and the projection operator is

$$\text{proj}_{V_{i-1}}\mathbf{v}_i = \sum_{j=1}^{i-1} \langle \mathbf{u}_j, \mathbf{v}_i \rangle \mathbf{u}_j \tag{22}$$

In the specific case of the Euclidean inner product space, the inner product operator $\langle \cdot, \cdot \rangle$ represents the standard Euclidean dot product. That is, for vectors $\mathbf{a}$ and $\mathbf{b}$, we have

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b} \tag{23}$$

and the distance metric (norm) is

$$\|\mathbf{a}\| = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle} \tag{24}$$

Thus, the projection operator Eq. (22) can be expressed as

$$\text{proj}_{V_{i-1}}\mathbf{v}_i = \left(\sum_{j=1}^{i-1} \mathbf{u}_j \mathbf{u}_j^T \right) \mathbf{v}_i \tag{25}$$

and we have

$$\mathbf{v}_i - \text{proj}_{V_{i-1}}\mathbf{v}_i = \left(\mathbf{1} - \sum_{j=1}^{i-1} \mathbf{u}_j \mathbf{u}_j^T \right) \mathbf{v}_i \tag{26}$$

Other inner products, and corresponding inner product spaces, can be defined. A particularly useful one can be defined with respect to the inverse of the configuration space mass matrix $\mathbf{M}(\mathbf{q})$, where

$$\langle \mathbf{a}, \mathbf{b} \rangle_{M^{-1}} \triangleq \mathbf{a}^T \mathbf{M}^{-1} \mathbf{b} \tag{27}$$

and the distance metric is

$$\|\mathbf{a}\|_{M^{-1}} = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle_{M^{-1}}} \tag{28}$$

The vector space $\mathbb{R}^n$ endowed with the inner product of Eq. (27) forms an inner product space. The process of orthogonalization in this inner product space follows the same procedure as described by Eqs. (19)–(22). For notational brevity we will omit the subscript M^{-1} throughout the remainder of this paper. All occurrences of the inner product will refer to this operation performed with respect to $\mathbf{M}^{-1}$.

2.4 Orthogonalization of Jacobian Matrices. To generate a mutually orthogonal set of matrices $\{\mathbf{Q}_1^T, \cdots, \mathbf{Q}_n^T\}$ from the set of Jacobian transposes $\{\mathbf{J}_1^T, \cdots, \mathbf{J}_n^T\}$, we will make use of the inner product operator of Eq. (27). We note that when applied to matrices this operator does not qualify as an inner product. That is, for matrices $\mathbf{A}$ and $\mathbf{B}$, the operation

$$\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{A}^T \mathbf{M}^{-1} \mathbf{B} \tag{29}$$

is not a scalar. Consequently, we do not have an inner product space as before when we were considering vectors. Nevertheless, it will be conceptually useful to consider a process analogous to vector orthogonalization, which utilizes the operator of Eq. (29) on matrices. Specifically, two matrices, $\mathbf{A}$ and $\mathbf{B}$, will be considered orthogonal to each other if

$$\langle \mathbf{A}, \mathbf{B} \rangle = \langle \mathbf{B}, \mathbf{A} \rangle^T = \mathbf{A}^T \mathbf{M}^{-1} \mathbf{B} = \mathbf{0} \tag{30}$$

We begin the orthogonalization process with the assignment

$$\mathbf{Q}_1^T = \mathbf{J}_1^T \tag{31}$$

Subsequent terms are generated in a manner analogous to the Gram–Schmidt process

$$\mathbf{Q}_i^T = \mathbf{J}_i^T - \text{proj}_{V_{i-1}}\mathbf{J}_i^T \tag{32}$$

where the projection operator is defined as

$$\begin{aligned} \text{proj}_{V_{i-1}}\mathbf{J}_i^T &\triangleq \sum_{j=1}^{i-1} \mathbf{Q}_j^T \langle \mathbf{Q}_j^T, \mathbf{Q}_j^T \rangle^{-1} \langle \mathbf{Q}_j^T, \mathbf{J}_i^T \rangle \\ &= \left[\sum_{j=1}^{i-1} \mathbf{Q}_j^T (\mathbf{Q}_j \mathbf{M}^{-1} \mathbf{Q}_j^T)^{-1} \mathbf{Q}_j \mathbf{M}^{-1} \right] \mathbf{J}_i^T \end{aligned} \tag{33}$$

Thus

$$\begin{aligned} \mathbf{Q}_i^T &= \mathbf{J}_i^T - \sum_{j=1}^{i-1} \mathbf{Q}_j^T \langle \mathbf{Q}_j^T, \mathbf{Q}_j^T \rangle^{-1} \langle \mathbf{Q}_j^T, \mathbf{J}_i^T \rangle \\ &= \left[\mathbf{1} - \sum_{j=1}^{i-1} \mathbf{Q}_j^T (\mathbf{Q}_j \mathbf{M}^{-1} \mathbf{Q}_j^T)^{-1} \mathbf{Q}_j \mathbf{M}^{-1} \right] \mathbf{J}_i^T \end{aligned} \tag{34}$$

or more concisely

$$\mathbf{Q}_i^T = \left(\mathbf{1} - \sum_{j=1}^{i-1} \mathbf{Q}_j^T \overline{\mathbf{Q}}_j^T \right) \mathbf{J}_i^T \tag{35}$$

where $\overline{\mathbf{Q}}_j$ denotes the dynamically consistent mass-weighted inverse (defined in Sec. 2.1) of $\mathbf{Q}_j$. That is

$$\overline{\mathbf{Q}}_j^T \triangleq (\mathbf{Q}_j \mathbf{M}^{-1} \mathbf{Q}_j^T)^{-1} \mathbf{Q}_j \mathbf{M}^{-1} \tag{36}$$

We note that the process of generating an orthogonalized set of Jacobians from an arbitrary set using the matrix

$$\mathbf{1} - \sum_{j=1}^{i-1} \mathbf{Q}_j^T \overline{\mathbf{Q}}_j^T \tag{37}$$

is analogous to the process of generating an orthogonalized set of vectors from an arbitrary vector basis using the matrix

$$\mathbf{1} - \sum_{j=1}^{i-1} \mathbf{u}_j \mathbf{u}_j^T \tag{38}$$

We will now prove two orthogonality properties related to the matrices $\{\mathbf{Q}_1^T, \cdots, \mathbf{Q}_n^T\}$ and $\{\mathbf{J}_1^T, \cdots, \mathbf{J}_n^T\}$. These properties will be critical to the derivations presented in the following section. They are also analogous to properties associated with an orthogonalized set of vectors.

PROPERTY 1. *The set of matrices* $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_n^T\}$ *are mutually orthogonal.*

Proof. Let us assume that the set of matrices $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_{i-1}^T\}$ are mutually orthogonal. That is

$$\langle \mathbf{Q}_l^T,\mathbf{Q}_m^T\rangle = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{Q}_m^T = \mathbf{0} \tag{39}$$

for any $l,m\in\{1,\cdots,i-1\}$ where $l\neq m$. Now, from Eq. (35) we have the following relationship for $\mathbf{Q}_i^T$:

$$\mathbf{Q}_i^T = \mathbf{J}_i^T - \sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T \tag{40}$$

We can use this relationship to form the expression $\langle \mathbf{Q}_l^T,\mathbf{Q}_i^T\rangle$. Expanding, we have

$$\langle \mathbf{Q}_l^T,\mathbf{Q}_i^T\rangle = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T - \mathbf{Q}_l\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T = \mathbf{0} \tag{41}$$

Based on the orthogonality property given by Eq. (39) we note that the summation in Eq. (41) reduces to

$$\mathbf{Q}_l\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{Q}_l^T\overline{\mathbf{Q}}_l^T\mathbf{J}_i^T = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T \tag{42}$$

where $l\in\{1,\cdots,i-1\}$. Substituting this into Eq. (41) we have

$$\langle \mathbf{Q}_l^T,\mathbf{Q}_i^T\rangle = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T - \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T = \mathbf{0} \tag{43}$$

for any $l\in\{1,\cdots,i-1\}$. Thus, $\mathbf{Q}_i^T$ is orthogonal to $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_{i-1}^T\}$ and the set $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_i^T\}$ is mutually orthogonal. We further note that

$$\langle \mathbf{Q}_1^T,\mathbf{Q}_2^T\rangle = \mathbf{J}_1\mathbf{M}^{-1}[\mathbf{1}-\mathbf{J}_1^T(\mathbf{J}_1\mathbf{M}^{-1}\mathbf{J}_1^T)^{-1}\mathbf{J}_1\mathbf{M}^{-1}]\mathbf{J}_2^T = \mathbf{0} \tag{44}$$

We have shown that if $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_{i-1}^T\}$ are mutually orthogonal then $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_i^T\}$ are also mutually orthogonal. Since $\mathbf{Q}_1^T$ and $\mathbf{Q}_2^T$ are orthogonal then $\{\mathbf{Q}_1^T,\cdots,\mathbf{Q}_n^T\}$ must be mutually orthogonal by induction.

PROPERTY 2. *The matrices* $\mathbf{J}_l^T$ *and* $\mathbf{Q}_i^T$ *are orthogonal, where* $l < i$.

Proof. Given Eq. (35) we can express the following relationships for $\mathbf{J}_l^T$ and $\mathbf{Q}_i^T$:

$$\mathbf{J}_l^T = \mathbf{Q}_l^T + \sum_{j=1}^{l-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_l^T \tag{45}$$

$$\mathbf{Q}_i^T = \mathbf{J}_i^T - \sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T \tag{46}$$

We can use these relationships to form the expression $\langle \mathbf{J}_l^T,\mathbf{Q}_i^T\rangle$. Expanding, we have

$$\langle \mathbf{J}_l^T,\mathbf{Q}_i^T\rangle = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T - \mathbf{Q}_l\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T + \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\mathbf{J}_i^T - \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T \tag{47}$$

Based on the orthogonality property given by Eq. (39), we note that the first summation in Eq. (47) reduces to

$$\mathbf{Q}_l\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{Q}_l^T\overline{\mathbf{Q}}_l^T\mathbf{J}_i^T = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i^T \tag{48}$$

and the third summation in Eq. (47) reduces to

$$\mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\sum_{j=1}^{i-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T = \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T\mathbf{J}_i^T = \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\mathbf{J}_i^T \tag{49}$$

where $l < i$. Substituting Eqs. (48) and (49) into Eq. (47) yields

$$\langle \mathbf{J}_l^T,\mathbf{Q}_i^T\rangle = \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i - \mathbf{Q}_l\mathbf{M}^{-1}\mathbf{J}_i + \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\mathbf{J}_i^T - \mathbf{J}_l\sum_{j=1}^{l-1}\overline{\mathbf{Q}}_j\mathbf{Q}_j\mathbf{M}^{-1}\mathbf{J}_i^T = \mathbf{0} \tag{50}$$

Therefore, the matrices $\mathbf{J}_l^T$ and $\mathbf{Q}_i^T$ have been shown to be orthogonal, where $l < i$.

3 Task-Level Control Decomposition

PROPERTY 1 states that the matrices $\mathbf{Q}_l^T$ and $\mathbf{Q}_m^T$ are orthogonal, where $l\neq m$. Therefore

$$\mathbf{Q}_l^T\overline{\mathbf{Q}}_l^T\mathbf{Q}_m^T\overline{\mathbf{Q}}_m^T = \mathbf{Q}_l^T(\mathbf{Q}_l\mathbf{M}^{-1}\mathbf{Q}_l^T)^{-1}\mathbf{Q}_l\mathbf{M}^{-1}\mathbf{Q}_m^T\overline{\mathbf{Q}}_m^T = \mathbf{0} \tag{51}$$

where $l\neq m$. We can thus express the cumulative summation

$$\mathbf{1} - \sum_{j=1}^{i}\mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T \tag{52}$$

as the cumulative product

$$\prod_{j=1}^{i}(\mathbf{1} - \mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T) \tag{53}$$

We will define Eq. (53) as the cumulative null space projection matrix

$$\mathbf{N}_i^T \triangleq \prod_{j=1}^{i}(\mathbf{1} - \mathbf{Q}_j^T\overline{\mathbf{Q}}_j^T) = \mathbf{N}_{i-1}^T(\mathbf{1} - \mathbf{Q}_i^T\overline{\mathbf{Q}}_i^T) \tag{54}$$

Thus

$$\mathbf{Q}_i^T = \mathbf{N}_{i-1}^T\mathbf{J}_i^T \tag{55}$$

We would now like to express the recursive nature of the operational space equations. First, we will examine the nonorthogonal system. We recall the equation of motion (3)

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \tau \tag{56}$$

The control torque is composed of task and null space terms

$$\tau = \sum_{i=1}^{n}\mathbf{J}_i^T\mathbf{f}_i + \mathbf{N}_n^T\tau_* = \mathbf{J}^T\mathbf{f} + \mathbf{N}^T\tau_* \tag{57}$$

where $\mathbf{J}$ is the vertical concatenation of the n task Jacobians $\{\mathbf{J}_1,\cdots,\mathbf{J}_n\}$, the overall task force vector $\mathbf{f}$ is the vertical concatenation of the n task forces $\{\mathbf{f}_1,\cdots,\mathbf{f}_n\}$, and $\mathbf{N}^T=\mathbf{N}_n^T$. Premultiplying Eq. (56) by $\mathbf{J}\mathbf{M}^{-1}$ yields

$$\mathbf{J}\ddot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{b} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{f} \tag{58}$$

We note that the overall task acceleration vector $\ddot{\mathbf{x}}$ is the vertical concatenation of the n task accelerations $\{\ddot{\mathbf{x}}_1,\cdots,\ddot{\mathbf{x}}_n\}$. The kinematic relation $\ddot{\mathbf{x}}=\mathbf{J}\ddot{\mathbf{q}}+\dot{\mathbf{J}}\dot{\mathbf{q}}$ can be substituted into Eq. (58), yielding

$$\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{f} \tag{59}$$

Expanding yields

$$\begin{pmatrix}\ddot{\mathbf{x}}_1\\ \vdots\\ \ddot{\mathbf{x}}_n\end{pmatrix}+\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_1\dot{\mathbf{q}}\\ \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_n\dot{\mathbf{q}}\end{pmatrix}+\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{g}\\ \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{g}\end{pmatrix}=\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{J}_1^T & \cdots & \mathbf{J}_1\mathbf{M}^{-1}\mathbf{J}_n^T\\ \vdots & \ddots & \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{J}_1^T & \cdots & \mathbf{J}_n\mathbf{M}^{-1}\mathbf{J}_n^T\end{pmatrix}\begin{pmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_n\end{pmatrix} \quad (60)$$

or

$$\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_i\dot{\mathbf{q}}+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{g}=\mathbf{J}_i\mathbf{M}^{-1}\tau=\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{n}\mathbf{J}_j^T\mathbf{f}_j \quad (61)$$

We note that Eq. (60) is a fully dense system since, in general, no elements of $\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$ are *zero*. An alternate way of generating the task space dynamics is to use a set of orthogonalized task Jacobians to represent the control torque. That is

$$\tau=\sum_{i=1}^{n}\mathbf{Q}_i^T\mathbf{z}_i+\mathbf{N}^T\tau_* \quad (62)$$

where $\{\mathbf{z}_1,\cdots,\mathbf{z}_n\}$ are the task forces associated with the orthogonalized task Jacobians. We then express the task dynamics as

$$\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_i\dot{\mathbf{q}}+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{g}=\mathbf{J}_i\mathbf{M}^{-1}\tau=\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{n}\mathbf{Q}_j^T\mathbf{z}_j \quad (63)$$

It will be useful to split the summation on the right hand side into two parts. This will consist of the summation from 1 to i, and the summation from $i+1$ to n. So we have

$$\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{n}\mathbf{Q}_j^T\mathbf{z}_j=\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{i}\mathbf{Q}_j^T\mathbf{z}_j+\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=i+1}^{n}\mathbf{Q}_j^T\mathbf{z}_j \quad (64)$$

PROPERTY 2 states that the matrices $\mathbf{J}_l^T$ and $\mathbf{Q}_i^T$ are orthogonal, where $l<i$. Consequently, the summation from $i+1$ to n vanishes. So, Eq. (63) can be written as

$$\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_i\dot{\mathbf{q}}+\mathbf{J}_i\mathbf{M}^{-1}\mathbf{g}=\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{i}\mathbf{Q}_j^T\mathbf{z}_j \quad (65)$$

In matrix form we have

$$\ddot{\mathbf{x}}+\mathbf{J}\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}\dot{\mathbf{q}}+\mathbf{J}\mathbf{M}^{-1}\mathbf{g}=\mathbf{J}\mathbf{M}^{-1}\mathbf{Q}^T\mathbf{z} \quad (66)$$

or

$$\begin{pmatrix}\ddot{\mathbf{x}}_1\\ \vdots\\ \ddot{\mathbf{x}}_n\end{pmatrix}+\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_1\dot{\mathbf{q}}\\ \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{b}-\dot{\mathbf{J}}_n\dot{\mathbf{q}}\end{pmatrix}+\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{g}\\ \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{g}\end{pmatrix}=\begin{pmatrix}\mathbf{J}_1\mathbf{M}^{-1}\mathbf{Q}_1^T & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{J}_n\mathbf{M}^{-1}\mathbf{Q}_1^T & \cdots & \mathbf{J}_n\mathbf{M}^{-1}\mathbf{Q}_n^T\end{pmatrix}\begin{pmatrix}\mathbf{z}_1\\ \vdots\\ \mathbf{z}_n\end{pmatrix} \quad (67)$$

Given the lower triangular form of $\mathbf{J}\mathbf{M}^{-1}\mathbf{Q}^T$ we can solve this system much more efficiently than the system of Eq. (60). Specifically, we can solve for $\mathbf{z}$ by recursively solving for the subvectors $\mathbf{z}_i$. Thus

$$\mathbf{z}_i=(\mathbf{J}_i\mathbf{M}^{-1}\mathbf{Q}_i)^{-1}\left[\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}\left(\mathbf{b}+\mathbf{g}-\sum_{j=1}^{i-1}\mathbf{Q}_j^T\mathbf{z}_j\right)-\dot{\mathbf{J}}_i\dot{\mathbf{q}}\right] \quad (68)$$

Defining the cumulative control torque as

$$\tau_i=\sum_{j=1}^{i}\mathbf{Q}_j^T\mathbf{z}_j=\tau_{i-1}+\mathbf{Q}_i^T\mathbf{z}_i \quad (69)$$

we have

$$\mathbf{z}_i=(\mathbf{J}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}[\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}(\mathbf{b}+\mathbf{g}-\tau_{i-1})-\dot{\mathbf{J}}_i\dot{\mathbf{q}}] \quad (70)$$

The total control torque is

$$\tau=\tau_n+\mathbf{N}^T\tau_* \quad (71)$$

We can incorporate this recursive structure into a formal algorithm. Algorithm 1 describes the task-level orthogonalization procedure.

Algorithm 1 Task-level orthogonalization

$\mathbf{N}_0^T=\mathbf{1}$ {initialization}
$\tau_0=\mathbf{0}$ {initialization}
for $i=1$ to n **do**
    $\mathbf{Q}_i^T=\mathbf{N}_{i-1}^T\mathbf{J}_i^T$
    $\mathbf{z}_i=(\mathbf{J}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}[\ddot{\mathbf{x}}_i+\mathbf{J}_i\mathbf{M}^{-1}(\mathbf{b}+\mathbf{g}-\tau_{i-1})-\dot{\mathbf{J}}_i\dot{\mathbf{q}}]$
    $\overline{\mathbf{Q}}_i^T=(\mathbf{Q}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}\mathbf{Q}_i\mathbf{M}^{-1}$
    $\mathbf{N}_i^T=\mathbf{N}_{i-1}^T(\mathbf{1}-\mathbf{Q}_i^T\overline{\mathbf{Q}}_i^T)$
    $\tau_i=\tau_{i-1}+\mathbf{Q}_i^T\mathbf{z}_i$
end for
$\mathbf{N}^T=\mathbf{N}_n^T$
$\tau=\tau_n+\mathbf{N}^T\tau_*$ {total control torque}

The dynamic compensation equation and control law, analogous to Eqs. (15) and (16), are

$$\mathbf{z}_i=(\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{Q}_i^T)^{-1}[\mathbf{f}_i^\star+\mathbf{J}_i\hat{\mathbf{M}}^{-1}(\hat{\mathbf{b}}+\hat{\mathbf{g}}-\tau_{i-1})-\dot{\mathbf{J}}_i\dot{\mathbf{q}}] \quad (72)$$

$$\mathbf{f}_i^\star=\mathbf{K}_{p_i}(\mathbf{x}_{d_i}-\mathbf{x}_i)+\mathbf{K}_{v_i}(\dot{\mathbf{x}}_{d_i}-\dot{\mathbf{x}}_i)+\ddot{\mathbf{x}}_{d_i} \quad (73)$$

3.1 Single Task Case. We can apply Algorithm 1 to a single task $\mathbf{x}$. We have

$$\mathbf{z}_1=\mathbf{f}=(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}}+\mathbf{J}\mathbf{M}^{-1}(\mathbf{b}+\mathbf{g})-\dot{\mathbf{J}}\dot{\mathbf{q}}] \quad (74)$$

This is equivalent to the operational space equation of motion Eq. (8). Additionally, we have

$$\mathbf{N}_1^T=\mathbf{1}-\mathbf{J}^T\overline{\mathbf{J}}^T \quad (75)$$

$$\tau_1=\mathbf{J}^T\mathbf{f} \quad (76)$$

$$\mathbf{N}^T=\mathbf{N}_1^T=\mathbf{1}-\mathbf{J}^T\overline{\mathbf{J}}^T \quad (77)$$

$$\tau=\tau_1+\mathbf{N}^T\tau_*=\mathbf{J}^T\mathbf{f}+\mathbf{N}^T\tau_* \quad (78)$$

and the dynamic compensation equation and control law

$$\mathbf{f}=(\mathbf{J}\hat{\mathbf{M}}^{-1}\mathbf{J}^T)^{-1}[\mathbf{f}^\star+\mathbf{J}\hat{\mathbf{M}}^{-1}(\hat{\mathbf{b}}+\hat{\mathbf{g}})-\dot{\mathbf{J}}\dot{\mathbf{q}}] \quad (79)$$

$$\mathbf{f}^\star=\mathbf{K}_p(\mathbf{x}_d-\mathbf{x})+\mathbf{K}_v(\dot{\mathbf{x}}_d-\dot{\mathbf{x}})+\ddot{\mathbf{x}}_d \quad (80)$$

Of course, the efficacy, both analytical and computational, of Algorithm 1 is most apparent for multiple tasks, where the lower triangular form of $\mathbf{J}\mathbf{M}^{-1}\mathbf{Q}^T$ allows for efficient recursive solutions. This nevertheless demonstrates how simply it reproduces the single task solution.

4 Task-Level Control Decomposition With Constraints

We can introduce constraints into the recursive task structure developed in the previous section. This will provide a means of applying recursive task-level control structures to constrained systems.

We introduce a set of m_C holonomic and scleronomic (not explicitly time-dependent) [12] constraint equations $\phi(\mathbf{q})=\mathbf{0}$. The mechanism of Fig. 2 is an example of a system where the holonomic constraints describe a kinematic loop closure. A task, $\mathbf{x}$, is to be controlled subject to the constraints. In general, the con-

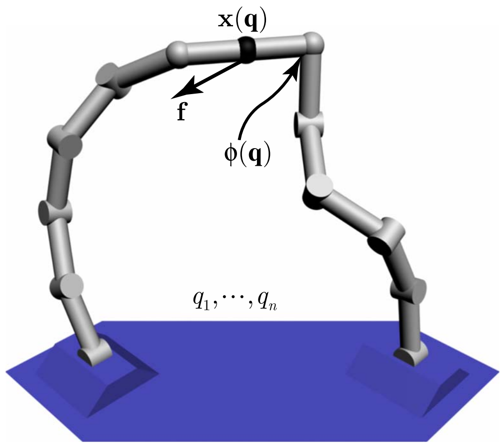

Fig. 2 A multibody system with holonomic constraints in the form of loop constraints. The task space vector $\mathbf{x}=(x,y,z)$ describes the Cartesian position of a point on one of the links. The objective is to control the system using task-level commands in the presence of the mechanism constraints.

straint equations are satisfied on a $p=n-m_C$ dimensional manifold Q^p in configuration space $Q=\mathbb{R}^n$ (see Fig. 3). The gradient of ϕ yields the constraint Jacobian matrix $\mathbf{\Phi}$

$$\mathbf{\Phi} = \frac{\partial \phi}{\partial \mathbf{q}} \in \mathbb{R}^{m_C \times n_q} \tag{81}$$

There exists a natural symmetry between the structure of task-level dynamics and the dynamics of constrained systems. This symmetry is derived from the common mathematical description used for tasks and constraints. Both utilize a Jacobian representation (task Jacobian $\mathbf{J}$ or constraint matrix $\mathbf{\Phi}$). Despite the common form used in specifying tasks and constraints, the mechanism by which tasks and constraints are satisfied differs. Tasks are *achieved* by means of a control input, whereas constraints are *imposed* by the physical structure of the multibody system. Nevertheless, due to their common mathematical form there are similarities between the structure of task dynamics and constrained dynamics. A more detailed discussion of these similarities can be found in Ref. [1].

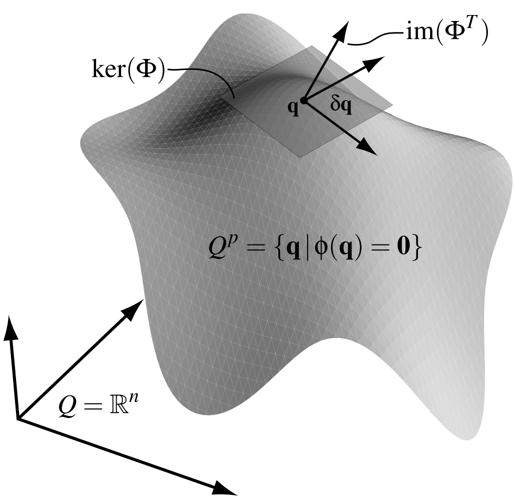

Fig. 3 The configuration space constrained motion manifold Q^p defined by the constraint equations $\phi(\mathbf{q})=0$. All constraint consistent virtual variations $\delta\mathbf{q}$ lie in the tangent space of Q^p and are orthogonal to the constraint forces.

Drawing on this symmetry between task-level dynamics and constrained dynamics we will represent the constraint matrix $\mathbf{\Phi}$ as the initial Jacobian in the recursive structure of Algorithm 1. In order to preserve the same indexing $(i=1,\cdots,n)$ on the tasks we will use a 0 index for the constraint terms. Thus, $\mathbf{J}_0^T=\mathbf{\Phi}^T$. Furthermore, the constraints imply $\ddot{\mathbf{x}}_0=\mathbf{0}$. We will also denote the cumulative total torque, which includes the constraint torque, as η_i rather than τ_i, which represents the cumulative control torque. Keeping these facts in mind, from Algorithm 1 we obtain

$$\mathbf{Q}_0^T = \mathbf{\Phi}^T \tag{82}$$

$$\mathbf{z}_0 = \alpha + \rho \tag{83}$$

$$\overline{\mathbf{Q}}_0^T = \mathbf{H\Phi M}^{-1} \tag{84}$$

$$\mathbf{N}_0^T = \mathbf{\Theta}^T \tag{85}$$

$$\eta_0 = \mathbf{\Phi}^T(\alpha + \rho) \tag{86}$$

where $\mathbf{H} \in \mathbb{R}^{m_C \times m_C}$ is the constraint space mass matrix, which reflects the system inertia projected at the constraint

$$\mathbf{H} \triangleq (\mathbf{\Phi M}^{-1}\mathbf{\Phi}^T)^{-1} \tag{87}$$

the term $\alpha \in \mathbb{R}^{m_C}$ is the vector of centrifugal and Coriolis forces projected at the constraint

$$\alpha \triangleq \mathbf{H\Phi M}^{-1}\mathbf{b} - \mathbf{H}\dot{\mathbf{\Phi}}\dot{\mathbf{q}} \tag{88}$$

the term $\rho \in \mathbb{R}^{m_C}$ is the vector of gravity forces projected at the constraint

$$\rho \triangleq \mathbf{H\Phi M}^{-1}\mathbf{g} \tag{89}$$

and $\mathbf{\Theta} \in \mathbb{R}^{n_q \times n_q}$ is the constraint null space matrix

$$\mathbf{\Theta} \triangleq \mathbf{1} - \overline{\mathbf{\Phi}}\mathbf{\Phi} \tag{90}$$

For tasks $i=1,\cdots,n$, we have

$$\mathbf{Q}_i^T = \mathbf{N}_{i-1}^T\mathbf{J}_i^T \tag{91}$$

$$\mathbf{z}_i = (\mathbf{J}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}[\ddot{\mathbf{x}}_i + \mathbf{J}_i\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \eta_{i-1}) - \dot{\mathbf{J}}_i\dot{\mathbf{q}}] \tag{92}$$

$$\overline{\mathbf{Q}}_i^T = (\mathbf{Q}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}\mathbf{Q}_i\mathbf{M}^{-1} \tag{93}$$

$$\mathbf{N}_i^T = \mathbf{N}_{i-1}^T(\mathbf{1} - \mathbf{Q}_i^T\overline{\mathbf{Q}}_i^T) \tag{94}$$

$$\eta_i = \eta_{i-1} + \mathbf{Q}_i^T\mathbf{z}_i \tag{95}$$

which is similar to the unconstrained case except that the cumulative total torque η_i replaces the cumulative control torque τ_i. The total torque η is given by

$$\eta = \eta_n + \mathbf{N}^T\tau_* = \tau + \mathbf{\Phi}^T\lambda = \mathbf{Q}_0^T\mathbf{z}_0 + \sum_{j=1}^{n}\mathbf{Q}_j^T\mathbf{z}_j + \mathbf{N}^T\tau_* \tag{96}$$

where λ is the vector of Lagrange multipliers representing the constraint forces [12]. The control torque τ is given by

$$\tau = \eta - \mathbf{\Phi}^T\lambda \tag{97}$$

To address the fact that some joints may be passive or unactuated, we introduce a selection matrix $\widetilde{\mathbf{S}} \in \mathbb{R}^{(n_q-k)\times n_q}$, where k is the number of actuated joints. The condition on the unactuated joints is then

$$\widetilde{\mathbf{S}}\tau = \mathbf{0} \tag{98}$$

Algorithm 2 summarizes the process of task-level control orthogonalization under constraints described by Eqs. (82)–(98). As in Algorithm 1, a controller employing the equations in Algorithm 2 should reflect estimates for the inertial and gravitational terms. Modifying Eq. (92), we have the following dynamic compensation equation:

$$\mathbf{z}_i = (\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{Q}_i^T)^{-1}[\mathbf{f}_i^\star + \mathbf{J}_i\hat{\mathbf{M}}^{-1}(\hat{\mathbf{b}} + \hat{\mathbf{g}} - \eta_{i-1}) - \dot{\mathbf{J}}_i\dot{\mathbf{q}}] \tag{99}$$

Algorithm 2 Task-level orthogonalization with constraints

1: $\mathbf{N}_0^T = \mathbf{\Theta}^T$ {initialization}
2: $\eta_0 = \mathbf{\Phi}^T(\alpha + \rho)$ {initialization}
3: for $i = 1$ to n **do**
4: $\quad \mathbf{Q}_i^T = \mathbf{N}_{i-1}^T\mathbf{J}_i^T$
5: $\quad \mathbf{z}_i = (\mathbf{J}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}[\ddot{\mathbf{x}}_i + \mathbf{J}_i\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \eta_{i-1}) - \dot{\mathbf{J}}_i\dot{\mathbf{q}}]$
6: $\quad \overline{\mathbf{Q}}_i^T = (\mathbf{Q}_i\mathbf{M}^{-1}\mathbf{Q}_i^T)^{-1}\mathbf{Q}_i\mathbf{M}^{-1}$
7: $\quad \mathbf{N}_i^T = \mathbf{N}_{i-1}^T(\mathbf{1} - \mathbf{Q}_i^T\overline{\mathbf{Q}}_i^T)$
8: $\quad \eta_i = \eta_{i-1} + \mathbf{Q}_i^T\mathbf{z}_i$
9: end for
10: $\mathbf{N}^T = \mathbf{N}_n^T$
11: $\eta = \eta_n + \mathbf{N}^T\tau_*$ {total torque}
12: $\tau = \eta - \mathbf{\Phi}^T\lambda$ {total control torque}
13: $\widetilde{\mathbf{S}}\tau = \mathbf{0}$ {condition on unactuated joints}

4.1 Single Task Case. We can apply Algorithm 2 to a single task $\mathbf{x}$. First, it is instructive to examine the nonorthogonal case. We express the constrained equation of motion

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \tau + \mathbf{\Phi}^T\lambda \tag{100}$$

Our control torque can be represented as [1]

$$\tau = \mathbf{\Phi}^T\mathbf{f}_C + \mathbf{J}^T\mathbf{f} = (\mathbf{\Phi}^T \quad \mathbf{J}^T)\begin{pmatrix}\mathbf{f}_C \\ \mathbf{f}\end{pmatrix} \tag{101}$$

where $\mathbf{f}$ is the component of the applied control force acting in the task direction and $\mathbf{f}_C$ is the component of the applied control force acting along the constraint direction. Premultiplying Eq. (100) by $\mathbf{\Phi}\mathbf{M}^{-1}$ yields

$$\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g} = \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{f} + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T(\mathbf{f}_C + \lambda) \tag{102}$$

and premultiplying Eq. (100) by $\mathbf{J}\mathbf{M}^{-1}$ yields

$$\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\mathbf{f} + \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T(\mathbf{f}_C + \lambda) \tag{103}$$

We wish to solve for the control forces $\mathbf{f}$ and $\mathbf{f}_C$ in Eqs. (102) and (103). These equations can be equivalently represented in matrix form as

$$\begin{pmatrix}\mathbf{0} \\ \ddot{\mathbf{x}}\end{pmatrix} + \begin{pmatrix}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}}\end{pmatrix} + \begin{pmatrix}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g} \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{g}\end{pmatrix} - \begin{pmatrix}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\end{pmatrix} \times \begin{pmatrix}\lambda \\ \mathbf{0}\end{pmatrix} = \begin{pmatrix}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\end{pmatrix}\begin{pmatrix}\mathbf{f}_C \\ \mathbf{f}\end{pmatrix} \tag{104}$$

This expression of the constrained task-level equation of motion was presented in Ref. [1]. In order to compute the control torque, we must solve for $\mathbf{f}_C$ and $\mathbf{f}$ in Eq. (104) and apply Eq. (101). This is a rather tedious process, which is detailed in the Appendix. Using Algorithm 2, we can arrive at this result much less tediously due to the recursive nature of the formulation. We have

$$\mathbf{z}_0 = \alpha + \rho = (\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T)^{-1}[\mathbf{\Phi}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g}) - \dot{\mathbf{\Phi}}\dot{\mathbf{q}}] \tag{105}$$

and

$$\mathbf{z}_1 = (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho\}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] \tag{106}$$

Our total torque is then

$$\eta = \mathbf{\Phi}^T(\alpha + \rho) + \mathbf{\Theta}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho)\} - \dot{\mathbf{J}}\dot{\mathbf{q}})] \tag{107}$$

and our total control torque is

$$\tau = \mathbf{\Phi}^T(\alpha + \rho - \lambda) + \mathbf{\Theta}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho\}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] \tag{108}$$

which is identical to the result in the Appendix. Noting that

$$\alpha + \rho = \overline{\mathbf{\Phi}}^T(\mathbf{b} + \mathbf{g}) - \mathbf{H}\dot{\mathbf{\Phi}}\dot{\mathbf{q}} \tag{109}$$

and defining

$$\mathbf{\Lambda}_c(\mathbf{q}) \triangleq (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1} \tag{110}$$

$$\mu_c(\mathbf{q}, \dot{\mathbf{q}}) \triangleq \mathbf{\Lambda}_c\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{b} - \mathbf{\Lambda}_c(\dot{\mathbf{J}} - \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T\mathbf{H}\dot{\mathbf{\Phi}})\dot{\mathbf{q}} \tag{111}$$

$$\mathbf{p}_c(\mathbf{q}) \triangleq \mathbf{\Lambda}_c\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{g} \tag{112}$$

we can express Eq. (108) as

$$\tau + \mathbf{\Phi}^T\lambda = \mathbf{\Theta}^T\mathbf{J}^T(\mathbf{\Lambda}_c\ddot{\mathbf{x}} + \mu_c + \mathbf{p}_c) + \mathbf{\Phi}^T(\alpha + \rho) \tag{113}$$

Equation (113) expresses the control torque as a function of the task accelerations $\ddot{\mathbf{x}}$, the kinematic and dynamic properties, and the constraint forces λ. Employing a linear control law the control equation can be expressed as

$$\tau + \mathbf{\Phi}^T\lambda = \hat{\mathbf{\Theta}}^T\mathbf{J}^T(\hat{\mathbf{\Lambda}}_c\mathbf{f}^\star + \hat{\mu}_c + \hat{\mathbf{p}}_c) + \mathbf{\Phi}^T(\hat{\alpha} + \hat{\rho}) \tag{114}$$

where

$$\mathbf{f}_i^\star = \mathbf{K}_{p_i}(\mathbf{x}_{d_i} - \mathbf{x}_i) + \mathbf{K}_{v_i}(\dot{\mathbf{x}}_{d_i} - \dot{\mathbf{x}}_i) + \ddot{\mathbf{x}}_{d_i} \tag{115}$$

These equations need to be complemented by the condition on the unactuated joints

$$\widetilde{\mathbf{S}}\tau = \mathbf{0} \tag{116}$$

As with Algorithm 1, the efficacy of Algorithm 2 is most apparent for multiple tasks due to the recursive nature of the algorithm. This nevertheless demonstrates how simply Algorithm 2 reproduces the single task solution compared with the tedious process detailed in the Appendix.

4.1.1 Example: Mechanism With Loop Closures. A parallel mechanism is depicted in Fig. 4. The constraint equations describe the loop closures and are given by

$$\phi(\mathbf{q}) = \begin{pmatrix}\mathbf{r}_{p_1} - \mathbf{r}_{l_1} \\ \mathbf{r}_{p_2} - \mathbf{r}_{l_2} \\ \mathbf{r}_{p_3} - \mathbf{r}_{l_3}\end{pmatrix} \tag{117}$$

We will define the task to control the position of the platform (see Fig. 4) while its orientation is uncontrolled; that is, $\mathbf{x} \triangleq (q_7 \ \ q_8)^T$. Considering only the elbow joints, q_2, q_4, and q_6, to be actuated we have

$$\widetilde{\mathbf{S}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{118}$$

where the condition on the unactuated joints is given by Eq. (116). The linear (PD) control law of Eq. (115) is used as the input of the decoupled system. The gains are chosen so as to achieve critically damped behavior of the task motion. Equation (114) is used to compute the control torque. Figure 5 shows simulation plots for the system under a goal position command. The time response of the platform orientation shows undamped oscillation due to the uncontrolled null space. Figure 6 shows simulation plots of the control torques generated for this motion. It is noted that the zero control torque is produced at the passive joints τ_1, τ_3, and τ_5 due to the condition of Eq. (116).

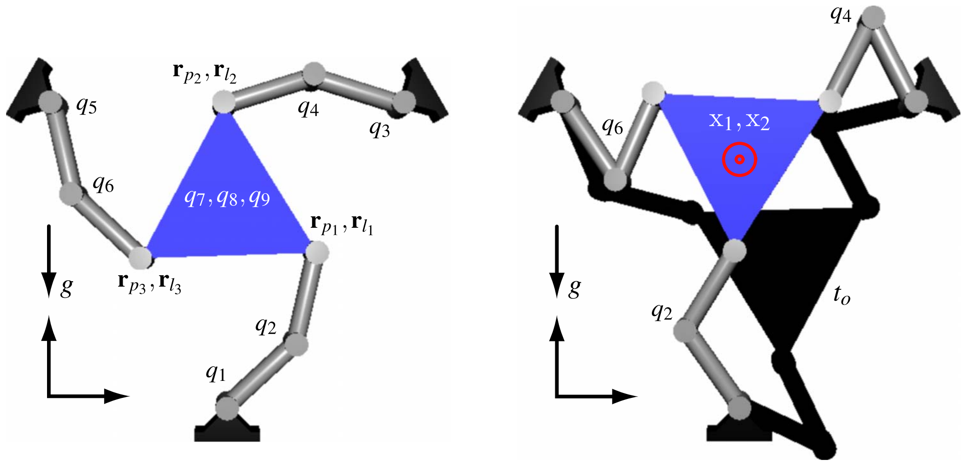

Fig. 4 (Left) Parallel mechanism consisting of serial chains with loop closures. The three elbow joints are actively controlled while the remaining joints are passive. (Right) The position of the platform is commanded to move to a target while its orientation is uncontrolled (uncontrolled null space motion). In this case n_q=9, m_T=2, m_C=6, and k=3.

5 Conclusion

We have presented an approach to formulating task-level motion-control for holonomically constrained multibody systems based on a mass-weighted orthogonal decomposition. This orthogonal decomposition is analogous to the Gram–Schmidt process used for orthogonalizing a vector basis. Recursive algorithms, which emerge from the mass-weighted orthogonal decomposition, have been presented to encapsulate this approach.

In addition to the use of task-level orthogonal decomposition, one of the other key elements of this work is that the natural symmetry between constrained dynamics and task space dynamics is exploited in order to synthesize dynamic compensation, which properly accounts for the system constraints while performing multiple control tasks. The presence of passive joints in the constrained system has also been accommodated. Collectively, the approaches presented here represent an effective set of techniques for applying task-level control to constrained multibody systems.

An example was presented to demonstrate the efficacy of this approach in simulation. As a practical matter it is assumed that the controller has access to the system state (via a forward dynamics solver in the simulated case or via sensors in the physical case) and estimates of the dynamic properties of the physical system. The results indicate that the analytical framework presented can be implemented in practical constrained multibody control problems.

Appendix: A Computing Control Torque Under Constraints

Given the expression for the control torque

$$\tau = \mathbf{\Phi}^T \mathbf{f}_C + \mathbf{J}^T \mathbf{f} = (\mathbf{\Phi}^T \quad \mathbf{J}^T) \begin{pmatrix} \mathbf{f}_C \\ \mathbf{f} \end{pmatrix} \tag{A1}$$

and the equation of motion

$$\begin{pmatrix} \mathbf{0} \\ \ddot{\mathbf{x}} \end{pmatrix} + \begin{pmatrix} \mathbf{\Phi M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} \\ \mathbf{JM}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} \end{pmatrix} + \begin{pmatrix} \mathbf{\Phi M}^{-1}\mathbf{g} \\ \mathbf{JM}^{-1}\mathbf{g} \end{pmatrix} - \begin{pmatrix} \mathbf{\Phi M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi M}^{-1}\mathbf{J}^T \\ \mathbf{JM}^{-1}\mathbf{\Phi}^T & \mathbf{JM}^{-1}\mathbf{J}^T \end{pmatrix} \times \begin{pmatrix} \lambda \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi M}^{-1}\mathbf{J}^T \\ \mathbf{JM}^{-1}\mathbf{\Phi}^T & \mathbf{JM}^{-1}\mathbf{J}^T \end{pmatrix} \begin{pmatrix} \mathbf{f}_C \\ \mathbf{f} \end{pmatrix} \tag{A2}$$

we can first solve for $\mathbf{f}_C$ and $\mathbf{f}$ by inverting the matrix

$$\begin{pmatrix} \mathbf{\Phi M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi M}^{-1}\mathbf{J}^T \\ \mathbf{JM}^{-1}\mathbf{\Phi}^T & \mathbf{JM}^{-1}\mathbf{J}^T \end{pmatrix} \tag{A3}$$

For a general 2×2 block matrix the inverse is given as

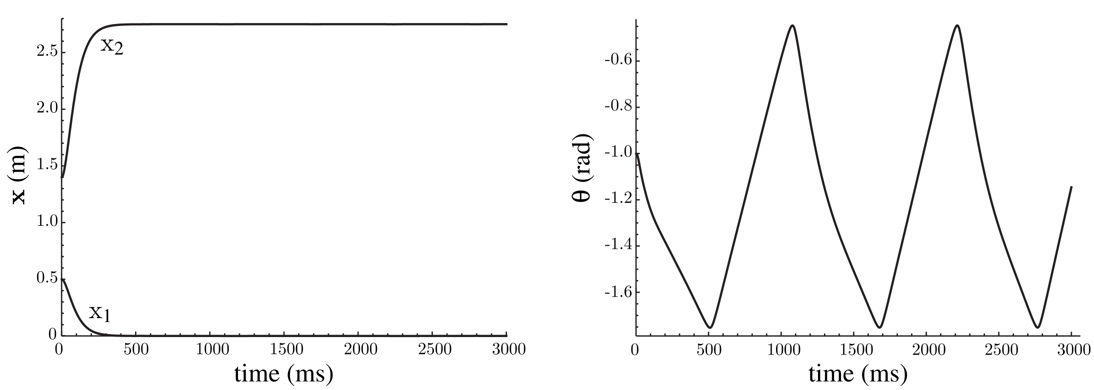

Fig. 5 The position of the platform is commanded to move to a target while its orientation is uncontrolled. (Left) Time response of the platform position showing linear critically damped behavior to the target. The control gains are K_p=100 and K_v=20. (Right) Time response of the platform orientation showing undamped null space oscillation due to the uncontrolled null space.

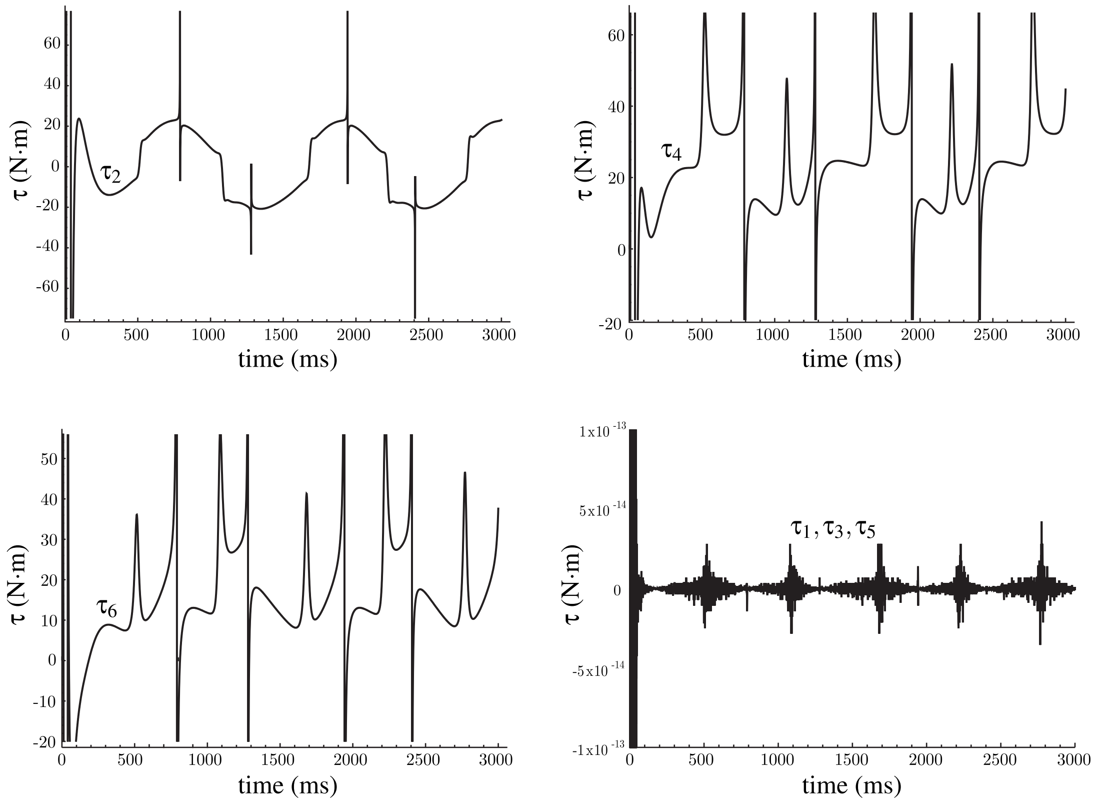

Fig. 6 The position of the platform is commanded to move to a target while its orientation is uncontrolled. Time response of the control torques during goal movement. Zero control torque (numerical error at the order of 10^{-14}) is produced at the passive joints τ_1, τ_3, and τ_5 due to the imposition of the passivity requirement in the controller. The control gains are K_p=100 and K_v=20.

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \tag{A4}$$

We can set

$$\mathbf{A} = \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T \tag{A5}$$

$$\mathbf{B} = \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T \tag{A6}$$

$$\mathbf{C} = \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T \tag{A7}$$

$$\mathbf{D} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \tag{A8}$$

Then

$$\mathbf{A}^{-1} = (\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T)^{-1} = \mathbf{H} \tag{A9}$$

$$\mathbf{A}^{-1}\mathbf{B} = \mathbf{H}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T = \overline{\mathbf{\Phi}}^T\mathbf{J}^T \tag{A10}$$

$$\mathbf{C}\mathbf{A}^{-1} = \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T\mathbf{H} = \mathbf{J}\overline{\mathbf{\Phi}} \tag{A11}$$

and

$$\begin{aligned}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T - \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T\mathbf{H}\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T)^{-1} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \\ &\quad - \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T\overline{\mathbf{\Phi}}^T\mathbf{J}^T)^{-1} = [\mathbf{J}\mathbf{M}^{-1}(\mathbf{1} - \mathbf{\Phi}^T\overline{\mathbf{\Phi}}^T)\mathbf{J}^T]^{-1} \\ &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\end{aligned} \tag{A12}$$

So we have

$$\begin{pmatrix} \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{J}^T \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{\Phi}^T & \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{H} + \overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\mathbf{J}\overline{\mathbf{\Phi}} & -\overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1} \\ -(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\mathbf{J}\overline{\mathbf{\Phi}} & (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1} \end{pmatrix} \tag{A13}$$

We can then solve for $\mathbf{f}_C$. This yields

$$\begin{aligned}\mathbf{f}_C &= -\overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\ddot{\mathbf{x}} + \alpha + \rho \\ &\quad + \overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\mathbf{J}\overline{\mathbf{\Phi}}(\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g}) \\ &\quad - \overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g}) - \lambda\end{aligned} \tag{A14}$$

or

$$\begin{aligned}\mathbf{f}_C &= \alpha + \rho - \lambda - \overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} - \mathbf{J}\overline{\mathbf{\Phi}}(\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} \\ &\quad + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g}) + \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g}]\end{aligned} \tag{A15}$$

Simplifying further yields

$$\begin{aligned}\mathbf{f}_C &= \alpha + \rho - \lambda - \overline{\mathbf{\Phi}}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1} \\ &\quad \times[\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho\}) - \dot{\mathbf{J}}\dot{\mathbf{q}}]\end{aligned} \tag{A16}$$

Solving for $\mathbf{f}$ we have

$$\begin{aligned}\mathbf{f} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\ddot{\mathbf{x}} - (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}\mathbf{J}\overline{\mathbf{\Phi}}(\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g}) \\ &\quad + (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g})\end{aligned} \tag{A17}$$

or

$$\mathbf{f} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} - \mathbf{J}\overline{\mathbf{\Phi}}(\mathbf{\Phi}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{\Phi}}\dot{\mathbf{q}} + \mathbf{\Phi}\mathbf{M}^{-1}\mathbf{g}) + \mathbf{J}\mathbf{M}^{-1}\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{g}] \tag{A18}$$

Simplifying yields

$$\mathbf{f} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho\}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] \tag{A19}$$

So

$$\tau = \mathbf{\Phi}^T\mathbf{f}_C + \mathbf{J}^T\mathbf{f} = \mathbf{\Phi}^T(\alpha + \rho - \lambda) + \mathbf{\Theta}^T\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{\Theta}^T\mathbf{J}^T)^{-1} \times [\ddot{\mathbf{x}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{b} + \mathbf{g} - \mathbf{\Phi}^T\{\alpha + \rho\}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] \tag{A20}$$

References

[1] De Sapio, V., Khatib, O., and Delp, S., 2006, "Task-Level Approaches for the Control of Constrained Multibody Systems," Multibody Syst. Dyn., **16**(1), pp. 73–102.

[2] DeSapio, V., Holzbaur, K., and Khatib, O., 2006, "The Control of Kinematically Constrained Shoulder Complexes: Physiological and Humanoid Examples," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, IEEE, New York, Vol. 1, pp. 470–475.

[3] Bruyninckx, H., and Khatib, O., 2000., "Gauss' Principle and the Dynamics of Redundant and Constrained Manipulators," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, IEEE, New York, Vol. 3, pp. 2563–2568.

[4] Khatib, O., 1987, "A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation," IEEE J. Rob. Autom., **3**(1), pp. 43–53.

[5] Khatib, O., 1995, "Inertial Properties in Robotic Manipulation: An Object Level Framework," Int. J. Robot. Res., **14**(1), pp. 19–36.

[6] Burdick, J., 1989, "On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, IEEE, New York, Vols. 1–3, pp. 264–270.

[7] Khatib, O., 2004, "Whole-Body Dynamic Behavior and Control of Human-Like Robots," Int. J. Humanoid Robot., **1**(1), pp. 29–43.

[8] Khatib, O., Sentis, L., and Park, J., 2008, "A Unified Framework for Whole-Body Humanoid Robot Control With Multiple Constraints and Contacts," *European Robotics Symposium*, H. Bruyninckx, L. Preucil, and M. Kulich, eds., Springer-Verlag, New York, pp. 303–312.

[9] Papastavridis, J. G., 2002, *Analytical Mechanics: A Comprehensive Treatise on the Dynamics of Constrained Systems for Engineers, Physicists, and Mathematicians*, Oxford University Press, Oxford.

[10] Axler, S., 2001, *Linear Algebra Done Right*, 2nd ed., Springer, New York.

[11] Bretscher, O., 2001, *Linear Algebra With Applications*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

[12] Lanczos, C., 1986, *The Variational Principles of Mechanics*, Dover, New York.