

Article

Extended Operational Space Kinematics, Dynamics, and Control of Redundant Serial Robots

Edward J. Haug ^{1,*}, Vincent De Sapio ² and Adrian Peidro ³ ¹ Mechanical Engineering and Applied Mathematics, The University of Iowa, Iowa City, IA 52242, USA² HRL Laboratories, Malibu, CA 90265, USA; vdesapio@hrl.com³ Systems Engineering and Automation Department, Miguel Hernandez University, 03202 Elche, Spain; apeidro@umh.es

* Correspondence: echaug@gmail.com

Abstract: A recently developed differential geometric representation of redundant serial robot kinematics is employed to create a new extended operational space dynamics and control formulation that explicitly accounts for redundant robot degrees of freedom. This formulation corrects deficiencies in kinematics and dynamics of redundant serial robots that have relied for over half a century on error-prone generalized inverse velocity-based kinematics for redundancy resolution. New ordinary differential equations of robot operational space dynamics are obtained, without the need for ad hoc derivation, in terms of task coordinates and self-motion coordinates that represent robot redundancy. A new extended operational space control approach is presented that exploits ordinary differential equations of motion in terms of task and self-motion coordinates, enabling enforcement of desired output trajectories, obstacle avoidance, and performance constraints. Four examples are presented with a one-degree-of-redundancy robot that demonstrate the validity and superior performance of the new formulation, relative to the traditional task space method used for redundant serial robot control. Finally, an example with eight degrees of redundancy is presented that further illustrates superior performance of the new operational space formulation.

Keywords: redundant serial robots; robot dynamics; operational space robot control



Citation: Haug, E.J.; De Sapio, V.; Peidro, A. Extended Operational Space Kinematics, Dynamics, and Control of Redundant Serial Robots. *Robotics* **2024**, *13*, 170. <https://doi.org/10.3390/robotics13120170>

Academic Editor: Raffaele Di Gregorio

Received: 22 October 2024

Revised: 24 November 2024

Accepted: 27 November 2024

Published: 30 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Redundant Manipulator Kinematics, Dynamics, and Control

The excess in number of control inputs over the number of functional outputs that characterize the performance of kinematically redundant serial robots provides an opportunity to achieve a desired output (or task) trajectory, while simultaneously optimizing selected measures of robot performance [1]. With this opportunity, however, come challenges in creating a mathematical representation of robot kinematics and dynamics that quantitatively represents robot redundant degrees of freedom, including an infinite number of inputs that yield a given output. The generalized inverse velocity space kinematics model of redundant robots originally presented by Whitney [2] is shown in Section 2.2 and references cited therein to suffer serious deficiencies that have limited its ability to exploit benefits of redundant robot control. A method has recently been presented that analytically and computationally characterizes nonlinear set-valued robot inverse kinematics at the position and orientation level [3], correcting deficiencies due to the generalized inverse velocity approach. This formulation yields an extended operational space and ordinary differential equations (ODEs) of dynamics that explicitly represent the versatility provided by robot redundancy and are ideally suited for robot control.

1.2. Traditional Task Space Formulation

A substantial redundant robotics literature has evolved based on a *task space formulation* introduced by Khatib [4], using robot models based on task (or output) coordinates. To date,

models of redundant serial robots in the traditional task space formulation have been based on mass-corrected generalized inverse velocity space kinematics that suffer kinematic and kinetic error. A theoretical, computational, and experimental analysis of eight controllers presented in the literature that are based on velocity, acceleration, and force redundancy resolution is presented in [5] and compared with the traditional task space formulation [4], showing that the latter suffers significant inaccuracies.

To compensate for errors associated with the traditional task space formulation, Park et al. [6] proposed an *extended operational velocity space* that introduced self-motion velocities to complement task space velocities that are employed in the traditional formulation. While these additional terms yield a linear approximation of redundant robot degrees of freedom, the associated velocity space kinematics fail to define an accurate model of redundant robot dynamics.

1.3. Organization of the Paper

A new redundant robot inverse kinematics formulation at the position and orientation level [3] is employed herein, using differential geometry for the definition of a redundant serial robot *extended operational space*. This space is parameterized by operational coordinates that include self-motion coordinates that accurately account for redundant robot degrees of freedom. Extended operational space ODEs of dynamics are derived that are equivalent to robot multibody dynamics ODEs in input coordinates and explicitly account for redundant degrees of freedom. A new control approach is presented, using the extended operational space ODEs for accurate tracking of a desired output trajectory, while enforcing obstacle avoidance and optimization of selected measures of robot performance. This approach corrects deficiencies identified in [5] that degrade performance of traditional task space control and approximation errors induced by the extended operational velocity space formulation [6].

The kinematic structure of redundant serial robots is defined in Section 2, and deficiencies of generalized inverse velocity space kinematics are identified. Equations of robot kinematics in task and self-motion coordinates are presented in Section 3 using differential geometry, summarizing derivations in [3] leading to a new extended operational space formulation that resolves deficiencies associated with generalized inverse velocity space kinematics. A new system of redundant robot operational space ODEs of dynamics introduced in [3] is presented in Section 4 that explicitly accounts for robot redundancy and is equivalent to the equations of robot multibody dynamics in input coordinates. A new control approach based on the extended operational space ODE is presented and illustrated with four applications of a one-degree-of-redundancy serial robot in Section 5 and an eight-degree-of-redundancy example in Section 6. Finally, conclusions and recommendations for future research are presented in Section 7.

1.4. Contributions of the Paper

The primary contributions of the paper are as follows:

- A. Based on concepts introduced in [3], a fundamentally new extended operational space is defined for serial robot kinematics and dynamics:
 - An explicit set-valued inverse kinematic mapping is derived for input coordinates as functions of task and self-motion coordinates.
 - Extended operational coordinates are defined and shown to be equivalent to input coordinates in parameterizing robot configuration space.
 - ODEs of robot dynamics are derived with extended operational coordinates as state variables
- B. A fundamentally new operational space control approach is introduced, including the following:
 - Robot control laws are defined and implemented using extended operational coordinates and operational space ODEs.

- The control structure explicitly allows for the tracking of self-motion coordinates, which is the only known variant of operational space control that allows this.
- Four control examples are treated using a redundant planar robot with one degree of redundancy, demonstrating superior performance of the extended operational space formulation relative to the traditional task space approach.
- A control example is treated for a robot with eight degrees of redundancy, further demonstrating the superiority of the extended operational space approach.

2. Redundant Serial Robot Kinematics

A *redundant serial robot* is comprised of a *chain of bodies* that are connected by single-degree-of-freedom joints. *Joint relative input coordinates* y_i between bodies in the chain define the position and orientation of outboard bodies, relative to their inboard counterparts. The terminal body in the chain is the *end effector*, whose *output coordinates* $z \in \mathbb{R}^m$ characterize its working capability, defined as twice continuously differentiable functions of input coordinates $y \in \mathbb{R}^n$, $n > m$, in the *forward kinematic mapping*

$$z = G(y) \quad (1)$$

where input coordinates are independent generalized coordinates that define the configuration of the underlying mechanism [7]. Here, $\mathbb{R}^k$ refers to k -dimensional Euclidean space with elements $x \in \mathbb{R}^k$ in the form of column vectors $x = [x_1 \ \cdots \ x_k]^T$, and superscript T denotes matrix transpose. Bold characters denote vectors and matrices.

2.1. Velocity Space Kinematics

For over half a century, kinematics of redundant serial robots has been modeled in velocity space, yielding fundamental deficiencies in representation of both kinematic and dynamic performance. In 1969, Whitney [2] made a significant contribution to resolve motion rate control of serial robots, using the Moore–Penrose generalized inverse to obtain an approximate velocity space inverse kinematic mapping. Since $G(y)$ of Equation (1) is twice continuously differentiable, an apparent velocity differential equation is obtained by differentiating Equation (1) with respect to time,

$$\dot{z} = G'(y)\dot{y} \quad (2)$$

where the *Jacobian matrix* $G'(y) \equiv [\partial G_i(y)/\partial y_j]_{m \times n}$ has full row rank for redundant serial robots with no inverse kinematic singularities [3]. For such a robot, the Moore–Penrose generalized inverse,

$$G'^{\dagger}(y) \equiv G'^T(y) \left(G'(y) G'^T(y) \right)^{-1} \quad (3)$$

that satisfies $G'(y)G'^{\dagger}(y) = I$ is applied to Equation (2) to obtain an inverse velocity mapping,

$$\dot{y} = G'^{\dagger}(y)\dot{z} + \left(I_n - G'^{\dagger}(y)G'(y) \right) \dot{y}^0 \quad (4)$$

where with $\dot{y}^0 \in \mathbb{R}^n$ arbitrary, the second term on the right yields velocities in the null space of $G'(y)$. This velocity space kinematics formulation, with minor variations, has been used to represent redundant serial robot kinematics for half a century [1].

2.2. Deficiencies in Redundant Robot Velocity Space Kinematics

The first deficiency in velocity space kinematics is that Equation (2) is not an ODE. Written in the differential form $dz - G'(y)dy = 0$, it is seen to be a *Pfaffian differential equation* [8] that behaves more like a partial differential equation than an ODE and is extremely difficult to solve. The second deficiency is that $G'^{\dagger}(y)$ in Equation (3) is generally not a total differential, i.e., Equation (4) is generally nonholonomic and yields nonperiodic or *noncyclic* solutions for $y(t)$, even if $z(t)$ is periodic [3,9–11]. The third and most critical

deficiency is that the velocity space inverse kinematic mapping of Equation (4) does not lead to valid ODEs of robot dynamics. These deficiencies have been known for a quarter century [9,10,12,13], but they remain largely unresolved.

Early research on resolving deficiencies in the velocity space kinematics approach focused on physically based concepts such as introducing artificial damping, using impedance control, and carrying out torque optimization [14–16]. More recent research has focused on using numerical methods such as adaptive extended Jacobians and null-space projections [17,18]. These papers cite well over 100 additional papers that have sought to resolve the deficiencies noted above. Some progress has been reported, but fundamental deficiencies in the velocity space kinematics approach remain.

2.3. Traditional Task Space Dynamics of Redundant Serial Robots

The Moore–Penrose generalized inverse of Equation (3) represents a purely kinematic approach to robot analysis, since it contains no inertial information. The traditional task space approach [4] was introduced to address kinetics by including inertia effects into the definition of the generalized inverse matrix and projecting dynamics into the task space. In the *non-redundant* case $n = m$, when the Jacobian is nonsingular, any applied robot input space generalized force F^y can be produced by a task space force F^z acting at the task point, e.g., the end effector, along task coordinates. The applied generalized force is then obtained as $G^T(y)F^z$, using the *principle* of virtual work [7,19]. In the case of redundant robots, a null-space projection term $N^T(y)F^0$, $F^0 \in R^n$ is needed as a complement to the task term, in order to represent an arbitrary applied generalized force.

The *d'Alembert equation of motion* of a serial robot in input coordinates is [7,19]

$$F^y = G^T(y)F^z + N^T(y)F^0 = M(y)\ddot{y} - S(y, \dot{y}) - Q^y(y, \dot{y}) \quad (5)$$

where $M(y)$ is the symmetric positive definite mass matrix, $Q^y(t, y, \dot{y})$ is a vector of applied generalized forces that act on and between bodies in the robot chain, and $S(y, \dot{y})$ is a vector of velocity coupling terms (sometimes called Coriolis forces). The term $Q^y(t, y, \dot{y})$ is restricted here to gravitational force, i.e., $Q^y(y)$. The left side of Equation (5) represents the decomposition of input space generalized forces into task space and null-space terms, $F^y = \underbrace{G^T(y)F^z}_{\text{task space}} + \underbrace{N^T(y)F^0}_{\text{null space}}$. Multiplying Equation (5) by $G'(y)M^{-1}(y)$, noting that

$\ddot{z} = G'\ddot{y} + \dot{G}'\dot{y}$, rearranging terms, and suppressing arguments yields

$$\ddot{z} - G'M^{-1}S - GM^{-1}Q^y = G'M^{-1}G'^TF^z + G'M^{-1}N^TF^0 + \dot{G}'\dot{y} \quad (6)$$

A condition can now be imposed to ensure that the term $G'M^{-1}N^TF^0$ associated with the null space does not contribute to task space acceleration. This has been defined as *dynamic consistency* [13] and is expressed as

$$G'M^{-1}N^TF^0 = G'M^{-1}(I - G'^TG'^{T\#})F^0 = 0 \quad (7)$$

for all $F^0 \in R^n$, where $G'^{T\#}$ represents an arbitrary generalized inverse of G'^T that satisfies the identity $G'^TG'^{T\#}G'^T = G'^T$. The transpose of the *dynamically consistent generalized inverse* of the Jacobian is shown in [13] to be

$$G'^{T\#}(y) = [G'(y)M^{-1}(y)G'^T(y)]^{-1}G'(y)M^{-1}(y) \quad (8)$$

Defining the symmetric positive definite task space mass matrix $\Lambda(y) = [G'(y)M^{-1}(y)G'^T(y)]^{-1}$, the task space centrifugal and Coriolis force vector $\mu(y, \dot{y}) = -\Lambda(y)G'(y)M^{-1}(y)S(y, \dot{y}) - \Lambda(y)\dot{G}'(y)\dot{y}$, and the task space gravity force vector

$p(\mathbf{y}) = -\Lambda(\mathbf{y})G'(\mathbf{y})M^{-1}(\mathbf{y})Q^y(\mathbf{y})$ [13], Equation (6) can be formally expressed as the *task space equation of motion* [4],

$$F^z = \Lambda(\mathbf{y})\ddot{\mathbf{z}} + \mu(\mathbf{y}, \dot{\mathbf{y}}) + p(\mathbf{y}) \quad (9)$$

It is important to note that Equation (9) does not constitute a system of ODEs in $\mathbf{z}$, since its terms are dependent on robot input coordinates $\mathbf{y}$. It is shown in Section 3 that $\mathbf{y}$ cannot be written as an explicit function of $\mathbf{z}$. Therefore, the task coordinates $\mathbf{z} \in R^m$ and their derivatives do not constitute state variables in this system of equations. Since $\mathbf{y} \in R^n$ are state variables for the robot and $n > m$, task coordinates $\mathbf{z} \in R^m$ must be augmented by $n - m$ additional coordinates to create a valid system of *operational space generalized coordinates* that serve as *state variables*. The traditional task space is thus not a state space for the redundant robot.

3. Extended Operational Space

To extend the velocity space formulation of Section 2.1 to robot configuration (position and orientation) space, it is required that a set-valued inverse kinematic mapping be constructed, i.e., that a solution $\mathbf{y} = f(\mathbf{z}, \mathbf{v})$ of Equation (1) be found with an arbitrary vector $\mathbf{v} \in R^{n-m}$ of coordinates that define robot redundancy. Such a mapping has been presented in [3] for redundant serial robots, using differential geometry, and is summarized in this section. It is used in Section 4 to obtain a new system of extended operational space ODEs of redundant robot dynamics.

3.1. Inverse Configuration Kinematics

An input–output pair $\mathbf{x} = [\mathbf{y}^T \ \mathbf{z}^T]^T \in R^{n+m}$ defines a *robot configuration*. The *robot configuration space* is defined as $X = \{\mathbf{x} \in R^{n+m} : G(\mathbf{y}) - \mathbf{z} = 0\}$. It should be noted that *configuration space* in the robotics literature is often defined as the space of input (joint) coordinates. The robot configuration space defined herein is a subset of the product of input and output spaces [20,21] that satisfies Equation (1). It thus embodies the topology of the robot, not just its input space. More specifically, X is the graph of Equation (1) that is a *differentiable manifold* if $G(\mathbf{y})$ is smooth and $G'(\mathbf{y})$ has full rank, with a single chart $\phi(\mathbf{x}) = \mathbf{y}$ on X [21]. This is effectively a *forward kinematic differentiable manifold*, parameterized by $\mathbf{x} = \psi(\mathbf{y}) = [\mathbf{y}^T \ G^T(\mathbf{y})]^T$.

In order to make this paper complete, the critical topic of inverse configuration kinematics is summarized from original derivations in [3,22]. To avoid inverse kinematic singular behavior that occurs at input coordinates for which the Jacobian matrix $G'(\mathbf{y})$ is rank deficient [3], the *regular robot configuration space* is defined as $\tilde{X} = \{\mathbf{x} \in X : \text{rank}(G'(\mathbf{y})) = m\}$. The entire regular robot configuration space $\tilde{X}$ cannot, in general, be characterized by a single continuously differentiable inverse kinematic mapping. The only practical global inverse kinematic mapping is based on concepts of *differential geometry* [21] that employ *local kinematic representations* on open sets $N^j \subset \tilde{X}$, whose union is the regular configuration space, i.e., $\cup_j N^j = \tilde{X}$. In N^j , $\bar{\mathbf{x}}^j$ is a *base point* about which an inverse kinematic mapping is constructed over N^j .

For a given j and base point $\bar{\mathbf{x}}^j \in N^j \subset \tilde{X}$, define the $n \times m$ matrix $\mathbf{U}^j$ and an $n \times (n - m)$ matrix $\mathbf{V}^j$ such that

$$\mathbf{U}^j = G'(\bar{\mathbf{y}}^j) \quad G'(\bar{\mathbf{y}}^j) \mathbf{V}^j = 0 \quad \mathbf{V}^{jT} \mathbf{V}^j = \mathbf{I} \quad (10)$$

where $\mathbf{V}^j$ is computed as a matrix whose columns form an orthonormal basis for the null space of $G'(\bar{\mathbf{y}}^j)$, e.g., in MATLAB, using *singular value decomposition* [23]. The matrices $\mathbf{U}^j$ and $\mathbf{V}^j$ are defined to be constant on N^j . Note from the second relationship of Equation (10) that $\mathbf{U}^{jT} \mathbf{V}^j = 0$ and $\mathbf{V}^{jT} \mathbf{U}^j = 0$. Since $G'(\bar{\mathbf{y}}^j)$ has full rank, so do $\mathbf{U}^j$ and $\mathbf{V}^j$. Further, since $\mathbf{V}^{jT} \mathbf{U}^j = 0$, the columns of $\mathbf{V}^j$ and $\mathbf{U}^j$ are mutually orthogonal. The n combined linearly independent columns of $\mathbf{U}^j$ (m columns) and $\mathbf{V}^j$ ($n - m$ columns) therefore span R^n .

Using V^j and U^j of Equation (10), any solution y of Equation (1) in a neighborhood of $\bar{y}^j$ can be written as

$$y = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(u - \bar{u}^j) \quad (11)$$

where $\bar{v}^j$ and $\bar{u}^j$ are values of v and u associated with $\bar{x}^j$ on a trajectory that first enters N^j . They are introduced to ensure the continuity of y as a function of v and z in Equation (11). In N^1 , $\bar{v}^1 = 0$ and $\bar{u}^1 = 0$. At $y = \bar{y}^j$, Equation (11) implies $v = \bar{v}^j$ and $u = \bar{u}^j$.

To see that there is a unique solution of Equation (1) with y of Equation (11), i.e., of

$$G(\bar{y}^j + V^j(v - \bar{v}^j) - U^j(u - \bar{u}^j)) - z = 0 \quad (12)$$

for u as a function of z and v in a neighborhood of $z = \bar{z}^j$ and $v = \bar{v}^j$, the derivative of the left side of Equation (12) with respect to u , evaluated at $\bar{x}^j$, is $-G'(\bar{y}^j)U^j = -U^{jT}U^j$, which is nonsingular. The implicit function theorem [24] implies the existence of a unique, twice continuously differentiable solution $u = h^j(z, v)$ of Equation (12) in a neighborhood of $\bar{x}^j$. From Equation (11),

$$y(z, v) = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{u}^j) \quad (13)$$

This is the desired *set-valued inverse kinematic mapping* on N^j , in which coordinates $v \in R^{n-m}$ quantitatively define robot redundancy. Note that in contrast with Equation (4) that is linear in the free velocity $\dot{y}^0$, Equation (13) is nonlinear in v and represents nonlinear characteristics of redundant serial robots that cannot be characterized by Equation (4).

If $z(t)$ and $v(t)$ are periodic of period t_p on N^j , i.e., $z(t + t_p) = z(t)$ and $v(t + t_p) = v(t)$, and if Equation (13) holds throughout N^j , then

$$y(t + t_p) = \bar{y}^j + V^j v(t + t_p) - U^j(h(z(t + t_p), v(t + t_p)) - \bar{u}^j) = y(t)$$

Thus, $y(t)$ of Equation (13) is periodic of period t_p in N^j , and the extended operational space representation of robot kinematics is *cyclic* on N^j , or *locally cyclic*. For more details regarding cyclicity, see [10,11]. There is no basis to expect that the traditional task space approach is cyclic.

A computationally efficient iterative method for the evaluation of $h^j(z, v)$ is presented in Section 3.6. In this computation, when the number of iterations required for convergence in the numerical solution of Equation (12) exceeds a specified tolerance, the associated configuration $x = [y^T \ z^T]^T$ is designated $\bar{x}^{j+1}$, y , v , and u are designated $\bar{y}^{j+1}$, $\bar{v}^{j+1}$, and $\bar{u}^{j+1}$, a new neighborhood N^{j+1} is entered, and the parameterization of Equation (13) is redefined. As shown in [19] for a dynamic system simulation, less than 0.1% of CPU time and no user interaction is required for this reparameterization. For more detail on the process of selecting configurations $\bar{x}^j$ and reparameterization calculations, see [3,19].

For a given output z , with $v \in R^{n-m}$ arbitrary in a neighborhood of $\bar{v}^j$, Equation (13) defines a set of input coordinates,

$$\text{SMM}(z) = \left\{ y = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{u}^j) : v \text{ in a neighborhood of } \bar{v}^j \right\}$$

called the *robot self-motion manifold* in input space associated with output z . Since $u = h^j(z, v)$ is the solution of Equation (12), $G(\bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{u}^j)) - z = 0$, for all v in a neighborhood of $\bar{v}^j$, $y(z, v)$ of Equation (13) maps into the same z , i.e., $z = G(y(z, v))$. Elements of the vector $v \in R^{n-m}$ are thus called *self-motion coordinates*. With arbitrary self-motion coordinates v in a neighborhood of $\bar{v}^j$, Equation (13) defines $n - m$ *redundant degrees of freedom* of the robot that enable it to meet requirements that could not be met with a nonredundant robot. Self-motion coordinates v thus explicitly represent robot redundancy at the configuration level. This fundamental new result is in stark contrast with the velocity space kinematic representation presented in Sections 2.1 and 2.2.

It is important to note that the self-motion manifold is defined at the robot configuration level and is nonlinear in v . Since robot configuration information is not defined in the generalized inverse velocity formulation of Section 2.1, the self-motion manifold and self-motion coordinates are not available for obstacle avoidance and other performance optimization functions in the velocity space formulation. In fact, the second term of Equation (4) that defines a projection onto the null space of $G'(\mathbf{y})$ leads to a linear analysis in the velocity setting that only approximates the nonlinear self-motion manifold.

3.2. The Robot Extended Operational Space

Defining *extended operational coordinates* $\mathbf{w} = [\mathbf{z}^T \ \mathbf{v}^T]^T \in R^n$, the *robot extended operational space* W is the subset of R^n such that $\mathbf{x} = [\mathbf{z}^T \ \mathbf{y}^T]^T \in \tilde{X}$ for all v in a neighborhood of $\bar{v}^j$, where $\mathbf{y} \in SMM(\mathbf{z})$. Thus, the set-valued inverse kinematic mapping of Equation (13) is a twice continuously differentiable mapping from subsets of W into the robot input space. The fundamentally new space of extended operational coordinates and Equation (13) define inverse kinematic relations at the robot configuration level, i.e., between task space position and orientation coordinates of bodies that make up the robot mechanism, self-motion coordinates, and input coordinates between bodies in the chain that defines the serial robot. This is in stark contrast with robot velocity relations of Section 2.1 that suffer severe deficiencies as presented in Section 2.2. As is shown in Section 4, the extended operational space formulation enables the derivation of ODEs of robot dynamics that are equivalent to input space ODEs and correct deficiencies of the traditional task space dynamics formulation discussed in Section 2.3.

3.3. The Robot Functional Configuration Space

Defining robot functional coordinates $\mathbf{s} = [\mathbf{y}^T \ \mathbf{w}^T]^T = [\mathbf{y}^T \ \mathbf{z}^T \ \mathbf{v}^T]^T \in R^{2n}$, the robot functional configuration space is defined as

$$\tilde{S} = \bigcup_{N^j} \left\{ \mathbf{s} \in R^{2n} : \bar{\mathbf{x}}^j = [\bar{\mathbf{y}}^j \ \bar{\mathbf{z}}^j]^T \in \tilde{X}, \mathbf{y} = \bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v} - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{h}^j(\mathbf{z}, \mathbf{v}) - \bar{\mathbf{w}}^j) \right\}$$

Similar to the definition of the robot configuration space as the subset of the product of input and output spaces that satisfy Equation (1), $\tilde{S}$ is defined as the subset of the product of input and extended operational spaces that satisfy Equation (13).

From Equation (11), $\mathbf{v} = \bar{\mathbf{v}}^j + \mathbf{V}^{jT}(\mathbf{y} - \bar{\mathbf{y}}^j)$. This and the kinematic mapping of Equation (1) define a $\mathbf{y}$ -parameterization of $\tilde{S}$, $\mathbf{s} = \bar{\boldsymbol{\psi}}^j(\mathbf{y}) = [\mathbf{y}^T \ \mathbf{G}^T(\mathbf{y}) \ \bar{\mathbf{v}}^j + (\mathbf{y} - \bar{\mathbf{y}}^j)^T \mathbf{V}^j]^T \in \tilde{S}$ for $\mathbf{y}$ in a neighborhood of $\bar{\mathbf{y}}^j$. Conversely, the kinematic mapping of Equation (13) enables the $\mathbf{w}$ -parameterization $\mathbf{s} = \boldsymbol{\psi}^j(\mathbf{w}) = \left[\left(\bar{\mathbf{y}}^j + \mathbf{V}^j(\mathbf{v} - \bar{\mathbf{v}}^j) - \mathbf{U}^j(\mathbf{h}^j(\mathbf{z}, \mathbf{v}) - \bar{\mathbf{w}}^j) \right)^T \ \mathbf{z}^T \ \mathbf{v}^T \right]^T$ of $\tilde{S}$ on N^j . This duality of input and extended operational coordinate parameterizations of $\tilde{S}$ provides the foundation for a fundamentally new analytical and numerical representation of redundant serial robot kinematics and dynamics. The robot functional configuration space can be parameterized by either $\mathbf{y}$ or $\mathbf{w}$. It cannot, however, be parameterized by only output $\mathbf{z}$. It is not possible to obtain ODEs of robot dynamics in only $\mathbf{z}$, as is attempted in the traditional task space setting of Section 2.3 [4].

While $\tilde{S}$ has many of the characteristics of a differentiable manifold, the foregoing has not shown it is a differentiable manifold [21]. This leaves open questions regarding its singularity-free domains of robot functionality for future research.

Kinematics and dynamics on $\tilde{S}$ must be carried out on individual sets N^j , called *charts*, which cover $\tilde{S}$, and transitioned to adjacent charts as robot configurations progress along a trajectory in $\tilde{S}$, as shown schematically in Figure 1. A piecewise analysis on charts is unavoidable since, in general, there is no globally valid extended operational coordinate parameterization $\boldsymbol{\psi}(\mathbf{w})$ of $\tilde{S}$. This attribute of differential geometry that transforms local to global properties of sets and mappings is one of its greatest contributions. The unavoidable

reality, however, is that one must adopt local operational space parameterizations, since no global parameterization generally exists.

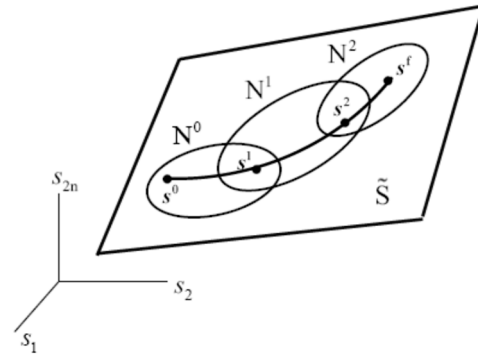


Figure 1. Trajectory along charts in $\tilde{S}$.

3.4. Velocity Kinematics on the Extended Operational Space

Differentiating the identity $z - G(\bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{w}^j)) = 0$ on N^j that is obtained by substituting y of Equation (13) into Equation (1) with respect to t , suppressing index j for notational convenience, and recalling that U^j and V^j are constant on N^j , yields

$$\dot{z} - G'(y)V\dot{v} + G'(y)U\dot{h} = 0 \quad (14)$$

The matrix $G'(\bar{y})U = U^T U$ is nonsingular, and $G'(y)$ is a continuously differentiable matrix function of y , so $G'(y)U$ is nonsingular in a neighborhood $K \subset \mathbb{R}^n$ of $\bar{y}$. The matrix

$$B(y) = (G'(y)U)^{-1} \quad (15)$$

is therefore well defined and continuously differentiable in K . A computationally efficient iterative method for the evaluation of $B(y)$ is presented in Section 3.6. For readers who may be concerned with the cost of evaluating the inverse matrix in Equation (15), note that its cost is no greater than that of evaluating $(G'(y)G'^T(y))^{-1}$ in the generalized inverse of Equation (3).

Substituting Equation (15) into Equation (14), $\dot{h} = -B(y)\dot{z} + B(y)G'(y)V\dot{v}$. Differentiating Equation (13) with respect to t , using this result and the chain rule of differentiation,

$$\begin{aligned} \dot{y} &= UB(y(z, v))\dot{z} + (V - UB(y(z, v))G'(y(z, v))V)\dot{v} \\ &= UB(y(z, v))\dot{z} + D(y(z, v))\dot{v} \end{aligned} \quad (16)$$

where $D(y(z, v)) \equiv [I - UB(y(z, v))G'(y(z, v))]V$. At $y = \bar{y}$, from Equation (10), $G'(\bar{y})V = 0$, so $D(\bar{y}) = V$ has full rank. Since $D(y)$ is a continuous matrix function of y , $D(y)$ has full rank in a neighborhood of $\bar{y}$. Matrix multiplication and the use of Equation (15) show that $G'(y)D(y) = [G'(y) - G'(y)UB(y)G'(y)]V = [G'(y) - G'(y)]V = 0$, so the columns of $D(y)$ span the null space of $G'(y)$ in a neighborhood of $\bar{y}$.

Note that $G'(y)(UB(y)) = (G'(y)U)B(y) = I$, so $UB(y)$ in Equation (16) is a generalized inverse of $G'(y)$, and Equation (16) is of the form of the ill-fated velocity space relation of Equation (4), except that $\dot{v} \in \mathbb{R}^{n-m}$ is comprised of independent self-motion velocities, and v appears in nonlinear form. The fundamental difference between Equations (4) and (16) is that the latter is the time derivative of Equation (13), i.e., it is holonomic, and Equation (4) is generally nonholonomic [10], i.e., it is not the time derivative of any algebraic (holonomic) equation. This result is obtained without having to resort to the use of the Frobenius theorem of differential geometry [9,10,21]. The inverse kinematic mapping of Equation (13) thus resolves deficiencies in redundant robot velocity space kinematics summarized in

Section 2.2 that have plagued redundant serial robot kinematics, dynamics, and control for half a century.

With *extended operational coordinates* $\mathbf{w} = [\mathbf{z}^T \quad \mathbf{v}^T]^T$ that quantitatively represent robot output and redundancy, Equation (16) may be written as

$$\dot{\mathbf{y}}(\mathbf{w}, \dot{\mathbf{w}}) = [\mathbf{UB}(\mathbf{y}(\mathbf{w})) \quad \mathbf{D}(\mathbf{y}(\mathbf{w}))] \dot{\mathbf{w}} = \mathbf{H}(\mathbf{y}(\mathbf{w})) \dot{\mathbf{w}} \quad (17)$$

where $\mathbf{H}(\mathbf{y}) = [\mathbf{UB}(\mathbf{y}) \quad \mathbf{D}(\mathbf{y})]$. To see that the matrix $\mathbf{H}(\mathbf{w})$ is nonsingular, expand

$$\begin{bmatrix} \mathbf{G}'(\mathbf{y}) \\ \mathbf{V}^T \end{bmatrix} \mathbf{H}(\mathbf{y}) = \begin{bmatrix} \mathbf{G}'(\mathbf{y}) \\ \mathbf{V}^T \end{bmatrix} [\mathbf{UB}(\mathbf{y}) \quad \mathbf{D}(\mathbf{y})] = \begin{bmatrix} \mathbf{G}'(\mathbf{y})\mathbf{UB}(\mathbf{y}) & \mathbf{G}'(\mathbf{y})\mathbf{D}(\mathbf{y}) \\ \mathbf{V}^T\mathbf{UB}(\mathbf{y}) & \mathbf{V}^T(\mathbf{I} - \mathbf{UB}(\mathbf{y})\mathbf{G}'(\mathbf{y}))\mathbf{V} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{I}$$

This shows that $\mathbf{H}(\mathbf{y})$ is nonsingular and that

$$\mathbf{H}^{-1}(\mathbf{y}) = \begin{bmatrix} \mathbf{G}'(\mathbf{y}) \\ \mathbf{V}^T \end{bmatrix} \quad (18)$$

The explicit solution of Equation (17) for $\dot{\mathbf{w}}$ is thus

$$\dot{\mathbf{w}} = \mathbf{H}^{-1}(\mathbf{y}) \dot{\mathbf{y}} = \begin{bmatrix} \mathbf{G}'(\mathbf{y}) \\ \mathbf{V}^T \end{bmatrix} \dot{\mathbf{y}} \quad (19)$$

3.5. Acceleration Kinematics on the Extended Operational Space

Differentiating Equation (19) with respect to t , $\ddot{\mathbf{w}} = \mathbf{H}^{-1}(\mathbf{y})\ddot{\mathbf{y}} + \begin{bmatrix} \mathbf{G}'(\mathbf{y})\hat{\mathbf{y}} \\ \mathbf{0} \end{bmatrix} \dot{\mathbf{y}}$, where over hat denotes a variable that is held constant for the indicated partial differentiation. Thus,

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{H}(\mathbf{y})\ddot{\mathbf{w}} - [\mathbf{UB}(\mathbf{y}) \quad \mathbf{D}(\mathbf{y})] \begin{bmatrix} \mathbf{G}'(\mathbf{y})\hat{\mathbf{y}} \\ \mathbf{0} \end{bmatrix} \dot{\mathbf{y}} \\ &= \mathbf{H}(\mathbf{y})\ddot{\mathbf{w}} + \mathbf{E}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{UB}(\mathbf{y})\ddot{\mathbf{z}} + \mathbf{D}(\mathbf{y})\ddot{\mathbf{v}} + \mathbf{E}(\mathbf{y}, \dot{\mathbf{y}}) \end{aligned} \quad (20)$$

where with $\mathbf{y}(\mathbf{w}) = \mathbf{y}(\mathbf{z}, \mathbf{v})$ of Equation (13) and $\dot{\mathbf{y}}(\mathbf{w}, \dot{\mathbf{w}})$ of Equation (17), $\mathbf{E}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{E}(\mathbf{y}(\mathbf{w}), \dot{\mathbf{y}}(\mathbf{w}, \dot{\mathbf{w}})) \equiv -\mathbf{UB}(\mathbf{y}(\mathbf{w})) \left(\mathbf{G}'(\mathbf{y})\hat{\mathbf{y}} \right)_y \dot{\mathbf{y}}(\mathbf{w}, \dot{\mathbf{w}})$.

3.6. Computation of $\mathbf{h}(\mathbf{z}, \mathbf{v})$ and $\mathbf{B}(\mathbf{y})$

While vector and matrix functions $\mathbf{h}(\mathbf{z}, \mathbf{v})$ and $\mathbf{B}(\mathbf{y})$ for serial redundant robots are shown to exist and be differentiable functions of $\mathbf{z}$, $\mathbf{v}$, and $\mathbf{y}$, their derivations do not show how to evaluate them. Since they are central to implementing kinematic position, velocity, and acceleration analyses, numerical methods for their evaluation presented in [3] are summarized here.

At $\bar{\mathbf{x}} \in \tilde{\mathbf{X}}$, $\mathbf{B}(\bar{\mathbf{y}}) = (\mathbf{G}'(\bar{\mathbf{y}})\mathbf{U})^{-1} = (\mathbf{U}^T\mathbf{U})^{-1}$ is numerically evaluated. For $\mathbf{y}^i$ at time t^i on a time grid, $\mathbf{B}(\mathbf{y}^i)$ must satisfy Equation (15), in the form $\bar{\mathbf{R}} \equiv (\mathbf{G}'(\mathbf{y}^i)\mathbf{U})\mathbf{B}(\mathbf{y}^i) - \mathbf{I} = \mathbf{0}$. With an approximation $\mathbf{B}^{(1)} \approx \mathbf{B}(\mathbf{y}^{i-1})$ of the solution and suppressing arguments $\mathbf{y}^i$, since they do not change in the iterative process for $\mathbf{B}(\mathbf{y}^i)$, a matrix version of *Newton–Raphson iteration* is $(\mathbf{G}'\mathbf{U})\Delta\mathbf{B}^{(j)} = -\bar{\mathbf{R}}^{(j)} = -\mathbf{G}'\mathbf{UB}^{(j)} + \mathbf{I}$, where (j) denotes the iteration number. Since $\mathbf{G}'\mathbf{U}$ need not be inverted with great precision in the Newton–Raphson process [23] and $\mathbf{B}^{(j)} \approx (\mathbf{G}'\mathbf{U})^{-1}$, $\Delta\mathbf{B}^{(j)} = -\mathbf{B}^{(j)}\mathbf{G}'\mathbf{UB}^{(j)} + \mathbf{B}^{(j)}$ and $\mathbf{B}^{(j+1)} = \mathbf{B}^{(j)} + \Delta\mathbf{B}^{(j)}$. This yields the computationally efficient iterative algorithm that requires only matrix multiplication,

$$\mathbf{B}^{(j+1)} = 2\mathbf{B}^{(j)} - \mathbf{B}^{(j)}\mathbf{G}'\mathbf{UB}^{(j)}, \quad j = 1, 2, \dots, \text{until } \|\mathbf{G}'\mathbf{UB}^{(j+1)} - \mathbf{I}\| \leq \text{Btol} \quad (21)$$

where Btol is the error tolerance.

While $\mathbf{h}(\mathbf{z}^i, \mathbf{v}^i)$ cannot be analytically determined, it can be evaluated as accurately as desired using Newton–Raphson iteration to solve Equation (12) for $\mathbf{u} = \mathbf{h}(\mathbf{z}^i, \mathbf{v}^i)$, with

$G_u \Delta u^{(j)} = -G' U \Delta u^{(j)} = -B^{-1} \Delta u^{(j)} = -G(\bar{y} + V(v - \bar{v}) - U(u^{(j)} - \bar{u}))$. The solution is $\Delta u^{(j)} = BG(\bar{y} + V(v - \bar{v}) - U(u^{(j)} - \bar{u}))$ and $u^{(j+1)} = u^{(j)} + \Delta u^{(j)}$, i.e., the iterative algorithm

$$u^{(j+1)} = u^{(j)} + BG(\bar{y} + V(v - \bar{v}) - U(u^{(j)} - \bar{u})), j = 1, 2, \dots \quad (22)$$

until $\|G(\bar{y} + V(v - \bar{v}) - U(u^{(j+1)} - \bar{u}))\| \leq \text{utol}$

where utol is the error tolerance. Since the Newton–Raphson method does not require an exact Jacobian [23], the matrix B is held constant throughout the process. This is an efficient computation, requiring only matrix multiplication.

4. ODEs of Input and Extended Operational Space Dynamics

4.1. Input Space ODE of Dynamics

Since input y is a vector of mechanism generalized coordinates, the *d'Alembert variational equation of motion* of a serial robot is [7,19]

$$\begin{aligned} \delta y^T [M(y)\ddot{y} - S(y, \dot{y}) - Q^y(t, y, \dot{y}) - F^y(y)] - \delta z^T F^z(z) \\ = \delta y^T [M(y)\ddot{y} - R(t, y, \dot{y}) - F^y(y)] - \delta z^T F^z(z) = 0 \end{aligned} \quad (23)$$

which holds for all virtual displacements δy and δz that satisfy the differential form of Equation (1), i.e., $\delta z = G'(y)\delta y$. In Equation (23), $M(y)$ is a symmetric positive definite mass matrix, $Q^y(t, y, \dot{y})$ is a vector of applied generalized forces that act on and between bodies in the robot chain, $S(y, \dot{y})$ is a vector of velocity coupling terms (sometimes called Coriolis forces), $F^y(y)$ is a vector of input generalized forces, $F^z(z)$ is a vector of generalized forces that act on the output frame due to its interaction with the environment and in shorthand notation, $R(t, y, \dot{y}) \equiv S(y, \dot{y}) + Q^y(t, y, \dot{y})$. Constraint reaction forces do not appear in this equation [7,19]. Substituting Equation (1) and $\delta z = G'(y)\delta y$ into Equation (23), $\delta y^T [M(y)\ddot{y} - R(t, y, \dot{y}) - F^y(y) - G'^T(y)F^z(G(y))] = 0$, which holds for arbitrary δy . This yields the *input space ODE of dynamics*,

$$M(y)\ddot{y} - R(t, y, \dot{y}) - F^y(y) - G'^T(y)F^z(G(y)) = 0 \quad (24)$$

4.2. Extended Operational Space ODE of Dynamics

Substituting $\ddot{y}$ of Equation (20), $\dot{y}(w, \dot{w})$ of Equation (16), and $y(w)$ of Equation (13) into Equation (24),

$$\begin{aligned} M(y(w))H(y(w))\ddot{w} + M(y(w))E(y(w), \dot{y}(w, \dot{w})) \\ - R(t, y(w), \dot{y}(w, \dot{w})) - F^y(y) - G'^T(y(w))F^z(G(y(w))) = 0 \end{aligned} \quad (25)$$

This is the *extended operational space ODE of dynamics*, which holds on each N^j . Since $M(y(w))$ and $H(y(w))$ are nonsingular, with initial conditions on w and $\dot{w}$, the extended operational space initial-value problem for the second-order ODE of Equation (25) is well posed, i.e., it has a unique solution that depends continuously on problem data [19].

For use in control system design, it is helpful to write Equation (25) as an explicit system of differential equations in z and v , where $w = [z^T \ v^T]^T$, and v represents robot redundancy. Substituting δy of Equation (16) and $\ddot{y}$ of Equation (20) into Equation (23),

$$\begin{aligned} \delta z^T \left[B^T(y)U^T M(y)UB(y)\ddot{z} + B^T(y)U^T M(y)D(y)\ddot{v} \right. \\ \left. + B^T(y)U^T (M(y)E(y, \dot{y}) - R(t, y, \dot{y}) - F^y(y)) - F^z(G(y)) \right] \\ + \delta v^T \left[D^T(y)M(y)UB(y)\ddot{z} + D^T(y)M(y)D(y)\ddot{v} \right. \\ \left. + D^T(y)(M(y)E(y, \dot{y}) - R(t, y, \dot{y}) - F^y(y)) \right] = 0 \end{aligned}$$

Since δz and δv are arbitrary, this yields a coupled system of second-order nonlinear *extended operational space ODEs* in z and v that is ideally suited for control system design,

$$\begin{aligned} & B^T(y)U^T M(y)UB(y)\ddot{z} + B^T(y)U^T M(y)D(y)\ddot{v} \\ & + B^T(y)U^T (M(y)E(y, \dot{y}) - R(t, y, \dot{y}) - F^y(y)) - F^z(G(y)) = 0 \\ & D^T(y)M(y)UB(y)\ddot{z} + D^T(y)M(y)D(y)\ddot{v} \\ & + D^T(y)(M(y)E(y, \dot{y}) - R(t, y, \dot{y}) - F^y(y)) = 0 \end{aligned} \quad (26)$$

The nonlinear dependence of terms in Equation (26) on z and v follows from the fact that $y(z, v) = \bar{y}^j + V^j(v - \bar{v}^j) - U^j(h^j(z, v) - \bar{u}^j)$ of Equation (13) and $\dot{y} = UB(y)\dot{z} + D(y)\dot{v}$ of Equation (16) are nonlinear functions of z and v .

As motion of the system progresses over $\tilde{S}$, *reparameterizations* in the extended operational space may be required in making the transition between charts, as depicted in Figure 1 and outlined in Section 3.3. The requirement for reparameterization is an inconvenience that cannot be avoided, since global parameterizations of $\tilde{S}$ with operational coordinates do not generally exist. The computational cost of reparameterization is, however, minimal [19]. In the process of reparameterization, the continuity of y , $\dot{y}$, z , and $\dot{z}$ is enforced, leading to *restart values*

$$v = \bar{v} \quad \dot{v} = V^T \dot{y} \quad (27)$$

of v and $\dot{v}$, the latter obtained by multiplying Equation (16) on the left by V^T . As a result, $v(t)$ is continuous, but $\dot{v}(t)$ may be discontinuous at reparameterization times.

Since the functional operational space $\tilde{S}$ can be parameterized by either y or w and steps in the transformations from Equation (24) to Equations (26) are reversible, the ODEs of Equations (24) through (26) are equivalent, i.e., they have the same solutions. Without coordinates v , Equation (26) in only output coordinates z could not be equivalent to robot ODEs of multibody dynamics of Equation (24). This shows that the Equation (9) in the traditional task space formulation [4] cannot be equivalent to robot ODE of multibody dynamics.

4.3. A Single-Degree-of-Redundancy Serial Robot Example

The redundant robot of Figure 2 has three input coordinates $y \in R^3$ and two output coordinates $z \in R^2$. Its kinematic equation is

$$z = \begin{bmatrix} y_1 + \cos y_3 \\ y_2 + \sin y_3 \end{bmatrix} \equiv G(y) \quad (28)$$

with Jacobian $G'(y) = \begin{bmatrix} 1 & 0 & -\sin y_3 \\ 0 & 1 & \cos y_3 \end{bmatrix}$. In addition to its analytical simplicity, this is an attractive example for analysis, since the Jacobian matrix is of full rank throughout the robot configuration space. Therefore, any pathological behavior of the robot cannot be attributed to a rank-deficient Jacobian.

To obtain the input space ODE of motion for the robot of Figure 2, vectors that locate its three bodies, modeled as point masses in the plane, are $r_1 = y_1 u_x$, $r_2 = y_1 u_x + y_2 u_y$, and $r_3 = (y_1 + \cos y_3)u_x + (y_2 + \sin y_3)u_y$, where $u_x = [1 \ 0]^T$ and $u_y = [0 \ 1]^T$. Variations, or virtual displacements, of the mass locations are obtained by taking the differentials of the r_i , $\delta r_1 = \delta y_1 u_x$, $\delta r_2 = \delta y_1 u_x + \delta y_2 u_y$, and $\delta r_3 = (\delta y_1 - \delta y_3 \sin y_3)u_x + (\delta y_2 + \delta y_3 \cos y_3)u_y$. Accelerations of the masses are obtained by taking two derivatives of the r_i , $\ddot{r}_1 = \ddot{y}_1 u_x$, $\ddot{r}_2 = \ddot{y}_1 u_x + \ddot{y}_2 u_y$, and $\ddot{r}_3 = (\ddot{y}_1 - \ddot{y}_3 \sin y_3 - \dot{y}_3^2 \cos y_3)u_x + (\ddot{y}_2 + \ddot{y}_3 \cos y_3 - \dot{y}_3^2 \sin y_3)u_y$. With these results, the d'Alembert variational equation of motion is [7,19]

$$\sum_i^3 m_i \delta r_i^T \ddot{r}_i - \delta y^T F^y - \delta z^T F^z + \delta y_2 m_2 g + (\delta y_2 + \delta y_3 \cos y_3) m_3 g = 0 \quad (29)$$

where F^y are input forces and torques, and F^z is the vector of external forces and torques that act on the end effector. Expanding products in Equation (29), using $u_x^T u_x = u_y^T u_y = 1$, $u_x^T u_y = 0$, and $\delta z = G'(y)\delta y$, yields the input ODE of motion of Equation (24) with $M(y) = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & -m_3 \sin y_3 \\ 0 & m_2 + m_3 & m_3 \cos y_3 \\ -m_3 \sin y_3 & m_3 \cos y_3 & m_3 \end{bmatrix}$ and $R(y, \dot{y}) = \begin{bmatrix} m_3 y_3^2 \cos y_3 \\ m_3 \dot{y}_3^2 \sin y_3 - (m_2 + m_3)g \\ -m_3 g \cos y_3 \end{bmatrix}$. For the extended operational space ODE of Equation (25), $B(y) = (G'(y)U)^{-1}$, $D(y) = [I_3 \quad -UB(y)G'(y)]V$, and $E(y, \dot{y}) = -UB(y) \begin{bmatrix} -\dot{y}_3^2 \cos y_3 \\ -\dot{y}_3^2 \sin y_3 \end{bmatrix}$. Since $M(y)$ is nonsingular and $H^{-1}(y)$ is given by Equation (18), suppressing arguments of functions for notational convenience, Equation (25) may be written in the explicit ODE form

$$\ddot{w} = -H^{-1}E + M^{-1}H^{-1}(R + F^y + G'^T F^z) \quad (30)$$

With initial conditions $y(0) = [0 \ 0 \ 0]^T = \bar{y}$, $z(0) = [1 \ 0]^T = \bar{z}$, and $\dot{y}(0) = [-1 \ 1 \ 0]^T$; associated $U = G'^T(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^T$ and $V = (1/\sqrt{2})[0 \ -1 \ 1]^T$; and output force $F^z = [0 \ 9]^T$, suppressing arguments, the input space ODE of Equation (24) is

$$\ddot{y} = M^{-1}(R + F^y + G'^T F^z) \quad (31)$$

In the extended operational space ODE of Equation (30), initial conditions on w and $\dot{w}$ are, using initial conditions on y and $\dot{y}$ of Equation (27), $w(0) = [z^T(0) \ v(0)]^T = [1 \ 0 \ 0]^T$ and $\dot{w}(0) = \begin{bmatrix} G'(y(0)) \\ V^T \end{bmatrix} \dot{y}(0)$.

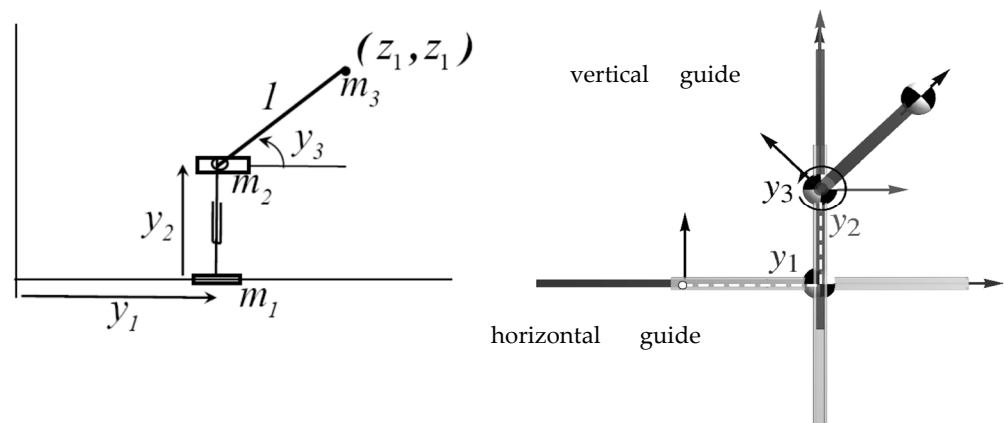


Figure 2. (Left) Redundant serial robot schematic. (Right) Robot mechanism structural rendering annotated with link axes, centers of mass, and joint displacements. All link masses are 1 kg.

In [3], an input force $F^y = [0 \ 9 \ \sin \pi t]^T$ is imposed, and Equations (30) and (31) are numerically integrated. The input coordinate trajectory $y_w(t)$ that is obtained from the computed value of $w(t)$, using the inverse kinematic mapping of Equation (13), deviates in norm from the value of $y(t)$ computed by integrating the input space ODE of Equation (31) by less than 10^{-11} over the entire time interval. This confirms the result reported in Section 4.2 that the input and extended operational space ODE of dynamics are equivalent. In this example, a single parameterization was adequate for the simulation. In more realistic applications, the reparameterization process outlined in Section 3.3 may be required.

Readers interested in larger scale applications are referred to [22,25]. Kinematics of a general seven-DOF spatial serial robot with specified task trajectory and obstacle avoidance constraint is treated in [25]. In [22], kinematics of a 23-DOF planar robot with 20 degrees of

redundancy is treated. A task trajectory is specified with constraints enforced on collision avoidance of each of 23 bodies with three obstacles, one moving. This large-scale kinematic control example is successfully treated using the formulation of Section 3. These examples are based on kinematic control, rather than dynamic system control presented in Section 5.

5. Extended Operational Space Control

Four examples are presented in this section to illustrate the use of the extended operational space ODE and traditional task space dynamics for the control of the one-degree-of-redundancy robot of Figure 2. The block diagram of the controlled plant dynamics is shown in Figure 3. The feedforward input to the plant is the vector F^y of generalized forces provided by the actuators, as prescribed by the controller. The feedback output of the plant is the input space state vector $\begin{bmatrix} y^T & \dot{y}^T \end{bmatrix}^T$, i.e., state measurements provided to the controller.

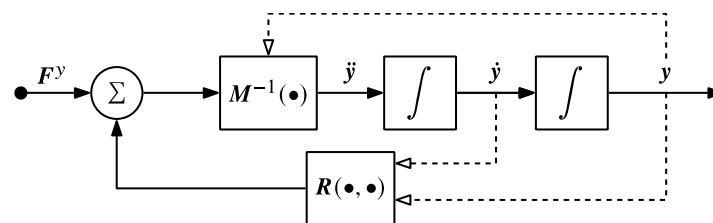


Figure 3. Block diagram for plant dynamics based on the input space ODE of Equation (24). Solid lines into a block indicate linear arguments and dashed lines represent nonlinear arguments.

The first example employs the traditional task space approach to track a figure 8 shaped output trajectory for the robot of Figure 2 with unit values of masses, providing a baseline with which to compare results obtained with the extended operational space formulation. The second example uses the extended operational space formulation with the same output trajectory and zero self-motion, to verify the functionality and cyclicity of that formulation. The third example uses both formulations to track the figure 8 shaped output trajectory, while minimizing the kinetic energy of the robot mechanism. The fourth example uses the extended operational space formulation to track the figure 8 shaped output trajectory, while avoiding collision with an obstacle.

5.1. Traditional Task Space Example

A closed-loop task space controller is investigated for controlling the redundant robot of Figure 2. The control is applied through generalized force F^y ,

$$F^y = G^T F^z \quad (32)$$

where F^z is the applied task space force in the task space dynamic equation of Equation (9) [13]. The input $F^\star$ of the decoupled system replaces the task acceleration $\ddot{z}$ in Equation (9) and is used to specify a simple proportional–derivative feedback control law,

$$F^\star = k_{z1}(z_d - z) + k_{z2}(\dot{z}_d - \dot{z}) + \ddot{z}_d \quad (33)$$

where $z_d(t)$ is the desired output, and k_{z_i} are proportional and derivative controller gains.

In this and subsequent control examples, it is assumed that the controller has access to perfect estimates of task space dynamic parameters. The task-level feedback controller described by Equations (32) and (33) is represented in block diagram form in Figure 4. Inputs to the controller are the input space state vectors that are used to compute task space coordinates. From these task space coordinates and desired reference signals, an input to the decoupled system is determined and used to estimate the required task space force vector. The output of the controller is the corresponding vector of generalized forces.

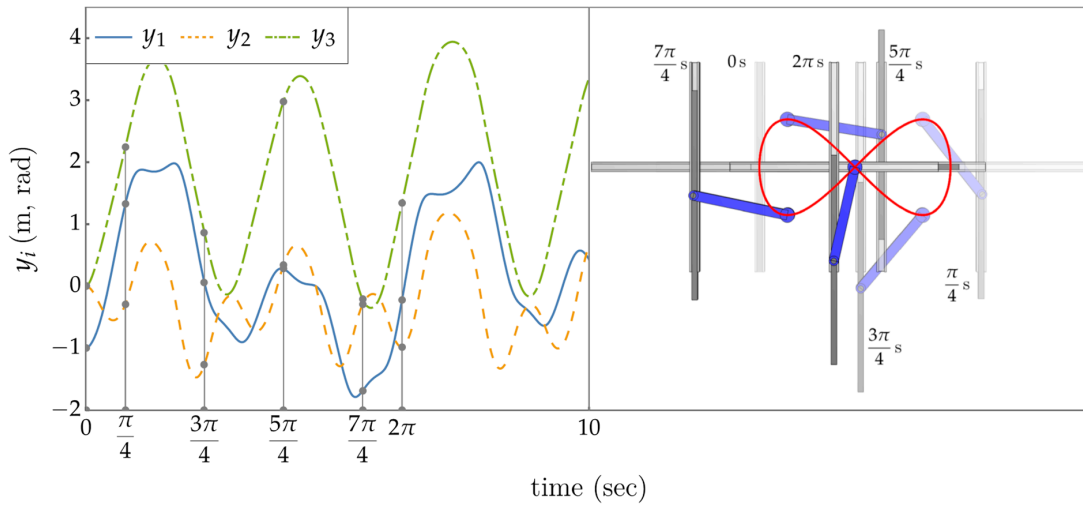


Figure 6. (Left) Plot of the input space coordinates versus time. The gray vertical lines and dots indicate key frames at regular intervals relative to the period of the task trajectory. (Right) Key frames composited into a single image, with earlier key frames fading away. Times are shown.

5.2. Extended Operational Space Example with Objective $v_d(t) = 0$

A closed-loop extended operational space controller is specified, in which control is applied through an input space force,

$$F^y = M(y)H(y)F^\star + M(y)E(y, \dot{y}) - R(y, \dot{y}) \quad (35)$$

Denoting the estimates of the dynamic parameters using an over hat, the input space acceleration generated from this force is $\ddot{y} = M^{-1}(F^y + R) = M^{-1}(\hat{M}\hat{H}F^\star + \hat{M}\hat{E} - \hat{R} + R)$ and the operational space acceleration is $\ddot{w} = H^{-1}(\ddot{y} - E) = H^{-1}(M^{-1}(\hat{M}\hat{H}F^\star + \hat{M}\hat{E} - \hat{R} + R) - E)$. With perfect estimates, $\ddot{w} = F^\star$.

As in the previous example, the input $F^\star$ of the decoupled system is used to specify a simple proportional–derivative feedback control law. In the extended operational space controller, this term is augmented to include self-motion coordinates, with the objective of achieving the self-motion trajectory $v_d(t) = 0$,

$$\begin{aligned} F^\star &= \overbrace{\begin{bmatrix} k_{z1}I_m & 0 \\ 0 & k_{v1}I_{(n-m)} \end{bmatrix}}^{K_1} (w_d - w) + \overbrace{\begin{bmatrix} k_{z2}I_m & 0 \\ 0 & k_{v2}I_{(n-m)} \end{bmatrix}}^{K_2} (\dot{w}_d - \dot{w}) + \ddot{w}_d \\ &= \begin{bmatrix} k_{z1}(z_d - z) + k_{z2}(\dot{z}_d - \dot{z}) + \ddot{z}_d \\ k_{v1}(v_d - v) + k_{v2}(\dot{v}_d - \dot{v}) + \ddot{v}_d \end{bmatrix} \end{aligned} \quad (36)$$

With perfect estimates, $F^\star - \ddot{w} = K_1(w_d - w) + K_2(\dot{w}_d - \dot{w}) + (\ddot{w}_d - \ddot{w}) = 0$, where k_{v_i} are control gains associated with self-motion coordinates. This yields the linearized error dynamics

$$K_1\varepsilon + K_2\dot{\varepsilon} + \ddot{\varepsilon} = 0 \quad (37)$$

where $\varepsilon = w_d - w$ is the extended operational space error.

The extended operational space feedback controller of Equations (35) and (36) can be applied to a task space trajectory tracking problem. It can also be applied to simultaneously track a task space trajectory and a self-motion trajectory. This is demonstrated by specifying a desired self-motion coordinate trajectory $v_d(t) = 0$ in this application, so self-motion coordinates that define robot redundancy appear explicitly in the control implementation of Equation (36). This contrasts with the traditional task space control implementation in Section 5.1 that does not depend explicitly on robot redundancy. The extended operational space feedback controller defined by Equations (35) and (36) is represented in block diagram form in Figure 7,

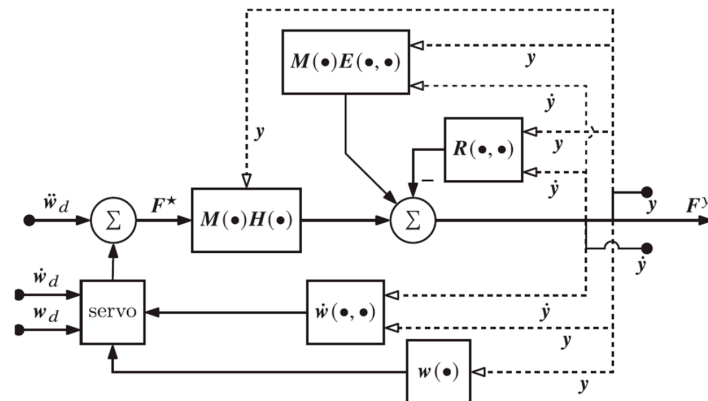


Figure 7. Block diagram for extended operational space feedback controller.

The operational space trajectory achieved through the application of the extended operational space controller is shown in Figure 8 (left). Controller gains of $k_{z_1} = k_{v_1} = 100$ (proportional) and $k_{z_2} = k_{v_2} = 20$ (derivative) were used. The tracked task trajectory matches the output trajectory generated in the previous example and the v coordinate of w is close to zero. The resulting periodic input coordinate trajectory obtained is shown in Figure 8 (Right). As predicted with periodic output and self-motion, the input space time histories of Figure 9 (left) are also periodic. Note that $y_2 = y_3$, since $v = (y_3 - y_2) / \sqrt{2} = 0$. Figure 9 (right) is the image generated as a composite of time frames at regular intervals relative to the period of the task trajectory. See Supplementary Materials for the animation.

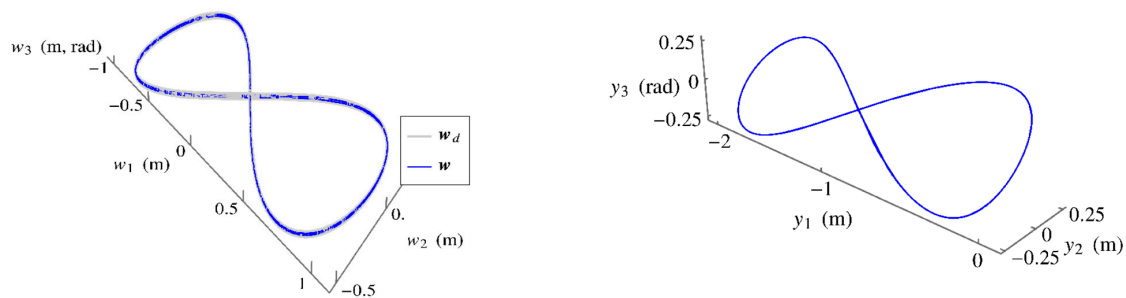


Figure 8. (Left) Plot of the operational space trajectory produced by the extended operational space controller compared to the reference trajectory. (Right) Input space trajectory showing cyclic behavior.

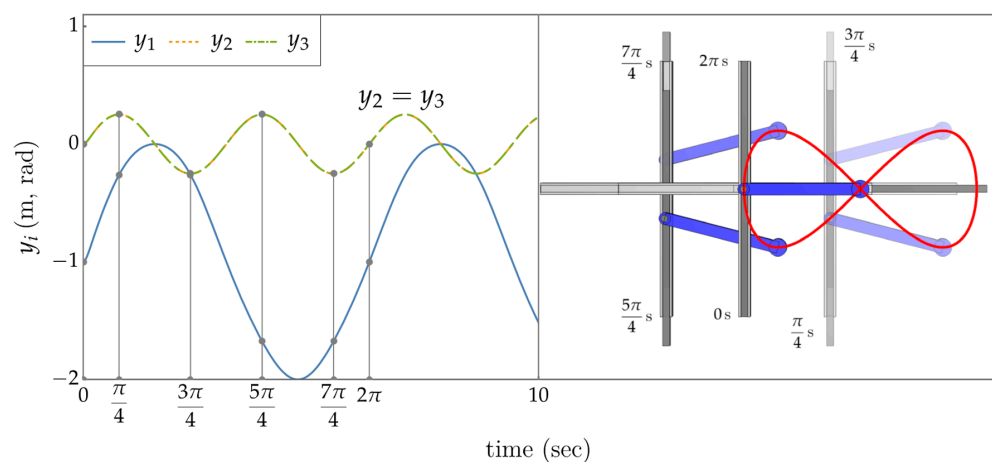


Figure 9. (Left) Plot of the input space coordinates versus time. The gray vertical lines and dots indicate key frames at regular intervals relative to the period of the task trajectory. (Right) Key frames composited into a single image, with earlier key frames fading away.

5.3. Example with Minimum Kinetic Energy Objective

The intention of this example is to show the effectiveness of minimizing a scalar function, while executing the tracking task, by projecting the gradient of a scalar function along the self-motion coordinates. Minimizing a scalar function by moving along its gradient, within the motion null space, is a common objective in traditional task space control [4].

Both traditional task space and extended operational space control are applied to minimize the kinetic energy (scalar function) of the mechanism during execution of the figure 8 shaped tracking task. In the case of traditional task space control, the gradient of kinetic energy $\nabla T(\mathbf{y}, \dot{\mathbf{y}})$ is projected onto the null space (local tangent space of the self-motion manifold at an instant), $\mathbf{N}^T(\mathbf{y})$,

$$\mathbf{F}^y = \mathbf{G}'^T(\mathbf{y})[\Lambda(\mathbf{y})\mathbf{F}^\star + \mu(\mathbf{y}, \dot{\mathbf{y}}) + \mathbf{p}(\mathbf{y})] - \mathbf{N}^T(\mathbf{y})\nabla T(\mathbf{y}, \dot{\mathbf{y}}) \quad (38)$$

In the case of extended operational space control, the gradient of kinetic energy is multiplied by $\mathbf{V}^T$ and concatenated into the term

$$\mathbf{F}^\star = \begin{bmatrix} k_{z_1}(z_d - z) + k_{z_2}(\dot{z}_d - \dot{z}) + \ddot{z}_d \\ -\mathbf{V}^T \nabla T(\mathbf{y}, \dot{\mathbf{y}}) \end{bmatrix} \quad (39)$$

When implementing this controller, $\mathbf{y} = \mathbf{y}(z, v)$ is evaluated using Equation (13), which depends explicitly on self-motion coordinates v that represent the redundancy of the robot. This is in contrast with the projection in traditional task space control that fails to fully exploit robot redundancy.

Results for the two cases are shown in Figure 10. Controller gains of $k_{z_1} = 100$ (proportional) and $k_{z_2} = 20$ (derivative) were used. The function T_o represents the baseline kinetic energy under the figure 8 shaped tracking task, with no null/self-motion space control. The function T_{os} represents the case of traditional task space control, with the gradient of the kinetic energy projected onto the null space $\mathbf{N}^T(\mathbf{y})$. The function T_{eos} represents extended operational space control, with the gradient of the kinetic energy projected onto the null space $\mathbf{V}^T$. The kinetic energy level with extended operational space control, T_{eos} , is much lower than in the baseline case. More significantly, T_{eos} is reduced with respect to T_{os} . For an animation of the results, see Supplementary Materials.

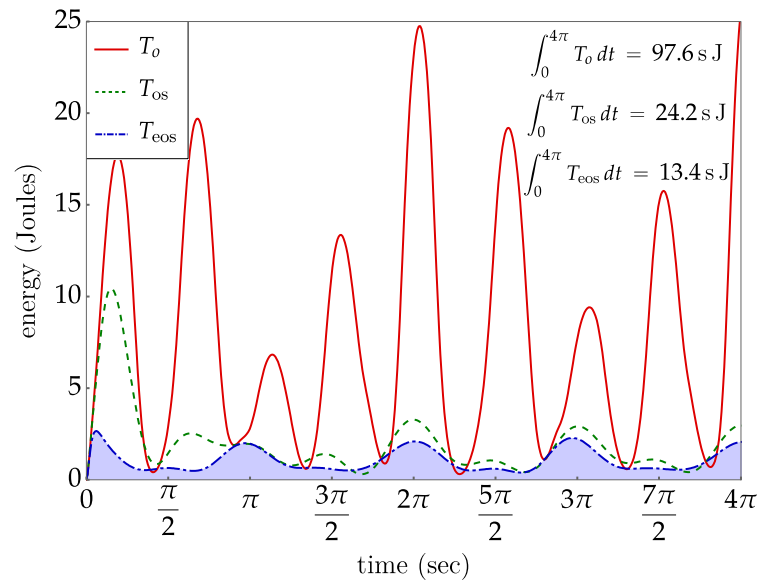


Figure 10. Plots of kinetic energy versus time. T_{os} represents the case of traditional task space control, with the gradient of kinetic energy projected onto the null space $\mathbf{N}^T(\mathbf{y})$. T_{eos} represents the case of extended operational space control, with the gradient of kinetic energy projected onto the null space $\mathbf{V}^T$.

It is important to note that a dynamically consistent generalized inverse of the task Jacobian is defined [13] to minimize the kinetic energy of the solution of the velocity kinematic equation, $\dot{z} = G'(y)\dot{y}$. In fact, the dynamically consistent inverse is a mass-weighted version of the Moore–Penrose inverse that has the property of yielding the least velocity norm solution. The mass-weighted inverse solution of the velocity equation is precisely the least kinetic energy (analogous to least norm). Despite this property of the dynamically consistent inverse, it can be seen in Figure 10 that it does not produce minimal kinetic energy over a time series. In retrospect, it should not be expected to do so, since the kinetic energy property that it satisfies is an instantaneous one that is dependent upon the specific configuration. There is no guarantee that self-motion configurations that the path follows are consistent with low kinetic energy relative to other paths.

5.4. Extended Operational Space Example with Obstacle Avoidance

Obstacle avoidance (particularly using artificial potential fields) is another common application for redundant robots. It is relevant to present an example with this approach in order to characterize the utility of implementing the artificial potential field approach within the extended operational space formalism. An obstacle is introduced into the robot workspace that is to be avoided by the top of the vertical guiderail of the robot, as shown in Figure 11. The task space controller is used to track the figure 8 shaped output. Controller gains of $k_{z_1} = 100$ (proportional) and $k_{z_2} = 20$ (derivative) were used. Rather than tracking a specific trajectory in the self-motion space, a reactive artificial potential field control was applied to avoid the obstacle, with input to the self-motion space. A repulsive artificial potential field was used, of the form

$$U_p = \frac{c}{2} \left(\frac{1}{d(y)} \right)^2 \quad (40)$$

where $c = 1$ is a constant coefficient, and $d(y) = l + y_2(t)$ is the distance function. The constant l is the offset to the top of the vertical guide rail, i.e., the point on the robot for which obstacle avoidance is specified. From left to right in Figure 11, the obstacle descends downward and then levels off. The point of influence for the robot is the top of the vertical guide rail. The field gradient is applied as control input into the self-motion space. Figure 11 (right) depicts the obstacle within the potential field.

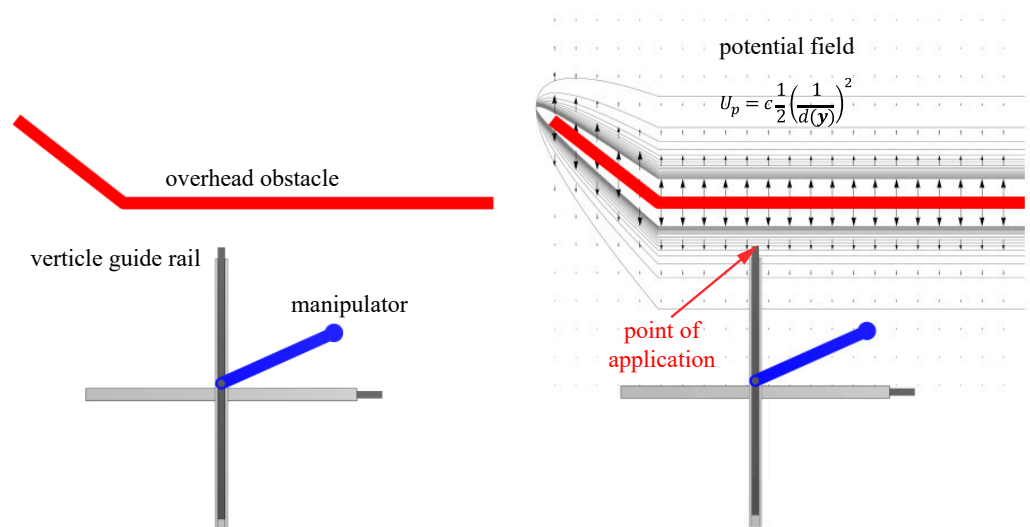


Figure 11. (Left) Overhead obstacle present in robot workspace. (Right) Robot depicted under the influence of a repulsive potential field.

In the input $F^\star$ of the decoupled system, the self-motion space tracking term is replaced by the (negative) gradient of the potential field $-\nabla_y U_p$, pre-multiplied by V^T . This defines

$$F^\star = \begin{bmatrix} k_x(z_d - z) + k_v(\dot{z}_d - \dot{z}) + \ddot{z}_d \\ -V^T \nabla_y U_p \end{bmatrix} \quad (41)$$

that depends on self-motion coordinate v through its influence on the $y(z, v)$ of Equation (13). The controller block diagram for this application is shown in Figure 12.

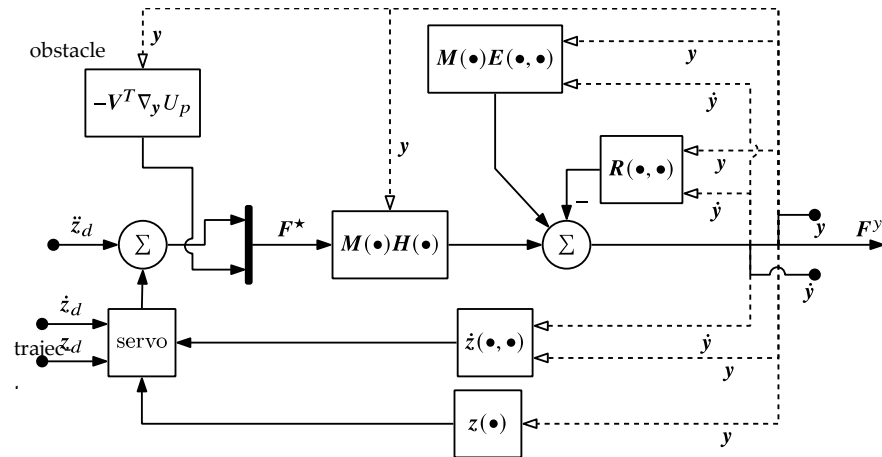


Figure 12. Block diagram of a controller, specifying task space trajectory tracking and artificial potential field obstacle avoidance in the self-motion manifold.

Results of the output trajectory tracking problem with simultaneous obstacle avoidance are shown in Figures 13 and 14. The orange line in Figure 14 is the trace of the upper tip of the vertical guide rail, which is the point of application of the repulsive potential field. The obstacle avoidance control dynamically decouples from the task trajectory control, due to its point of application within the self-motion space. Since the self-motion trajectory generated is not periodic, there is no reason to believe that the input trajectory obtained will be periodic, i.e., the robot will not be cyclic in this control generated motion. Nevertheless, v in the left plot of Figure 13 is nearly constant and z is periodic, so the joint displacements of Figure 13 (right) are nearly cyclic, as Presented in Supplementary Materials.

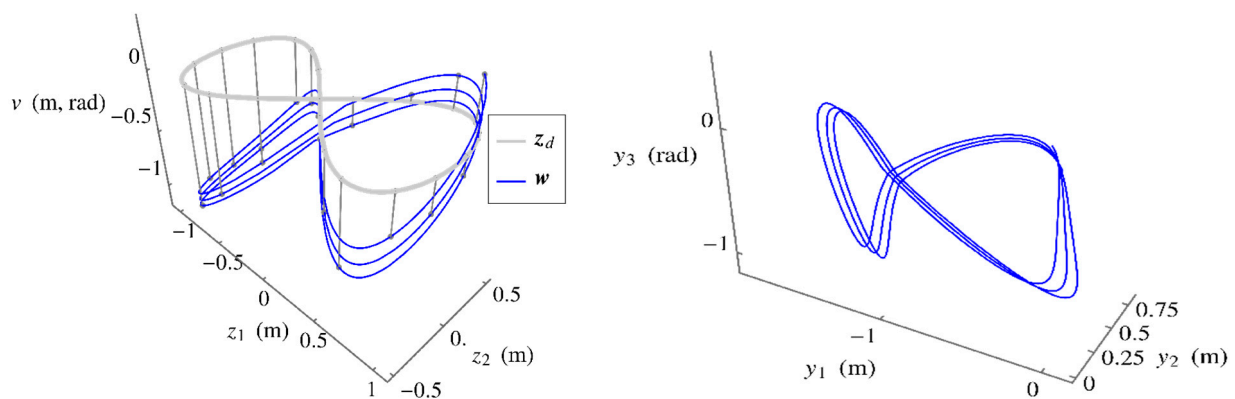


Figure 13. (Left) Plot of the operational trajectory produced by the extended operational space controller compared to the reference trajectory. (Right) Input space trajectory showing slightly acyclic behavior.

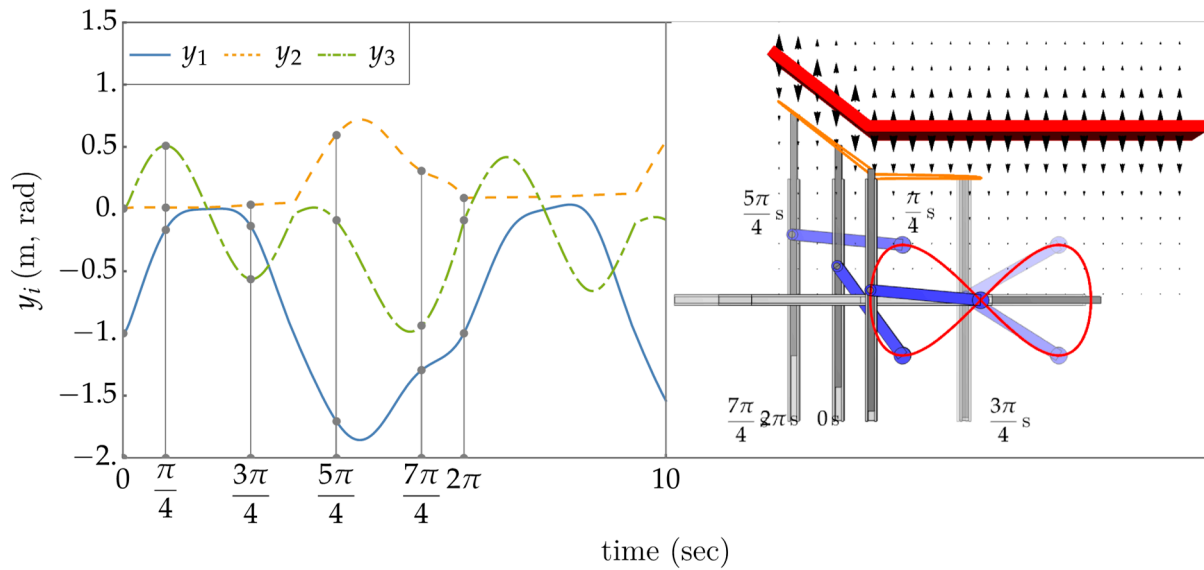


Figure 14. (Left) Plot of the input space coordinates versus t . (Right) Composite frame image (see animation in Supplementary Materials).

6. Eight-Degree-of-Redundancy Test Problem

The planar manipulator of Figure 15 (Left) with eight unit-length members has ten input coordinates $y \in \mathbb{R}^{10}$ and two output coordinates $z \in \mathbb{R}^2$. Its kinematic equation is

$$z = \begin{bmatrix} y_1 + \sum_{i=3}^{10} \cos\left(\sum_{j=3}^i y_j\right) \\ y_2 + \sum_{i=3}^{10} \sin\left(\sum_{j=3}^i y_j\right) \end{bmatrix} = G(y)$$

with Jacobian $G'(y) = \begin{bmatrix} 1 & 0 & col_3 & \cdots & col_{10} \\ 0 & 1 & & & \end{bmatrix}$, where $col_k = \begin{bmatrix} -\sum_{i=3}^{10} \left\{ \sin\left(\sum_{j=3}^i y_j\right) \cdot \sigma_k^i \right\} \\ \sum_{i=3}^{10} \left\{ \cos\left(\sum_{j=3}^i y_j\right) \cdot \sigma_k^i \right\} \end{bmatrix}^T$ is the k th column of the Jacobian, $3 \leq k \leq 10$, with σ_k^i defined as $\sigma_k^i = 0$ if $k > i$ and $\sigma_k^i = 1$ otherwise. In addition to its analytical simplicity, this is an attractive example for analysis, since the Jacobian matrix of $G'(y)$ is of full rank throughout the manipulator configuration space. Therefore, any pathological behavior of the manipulator cannot be attributed to a rank-deficient Jacobian.

To obtain the input space ODE of motion for this manipulator, unit point masses m_i were defined at the end of each link, with the global positions $r^1 = y_1 u_x$, $r^2 = y_1 u_x + y_2 u_y$, and $r^k = \left(y_1 + \sum_{i=3}^k \cos\left(\sum_{j=3}^i y_j\right) \right) u_x + \left(y_2 + \sum_{i=3}^k \sin\left(\sum_{j=3}^i y_j\right) \right) u_y$, $3 \leq k \leq 10$, where $u_x = [1 \ 0]^T$ and $u_y = [0 \ 1]^T$. The ODEs of motion of Equation (24) in input space are given by the matrices $M(y) = \sum_{i=1}^{10} m_i r_y^i r_y^i$, $S(y) = -\left(\sum_{i=1}^{10} m_i r_y^i \frac{d}{dt} (r_y^i) \right) \dot{y}$, and $Q^y(y) = -g \sum_{i=1}^{10} m_i r_y^i [0 \ 0 \ 1]^T$, where r_y^i is the Jacobian of r^i with respect to y , and $g = 9.80665 \text{ m/s}^2$ is the acceleration of gravity, pointing downward in Figure 15. In the following, two operational space controllers are compared to track the following figure 8 shaped task trajectory:

$$z_d(t) = 3 \begin{bmatrix} \sin(t) \\ \sin(t) \cos(t) \end{bmatrix} \quad (42)$$

For both traditional and extended operational space control, initial conditions of the manipulator were $\mathbf{y}(0) = -\frac{\pi}{4} [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$ and $\dot{\mathbf{y}}(0) = 0_{10 \times 1}$. This initial configuration of the manipulator is shown in Figure 15 (right).

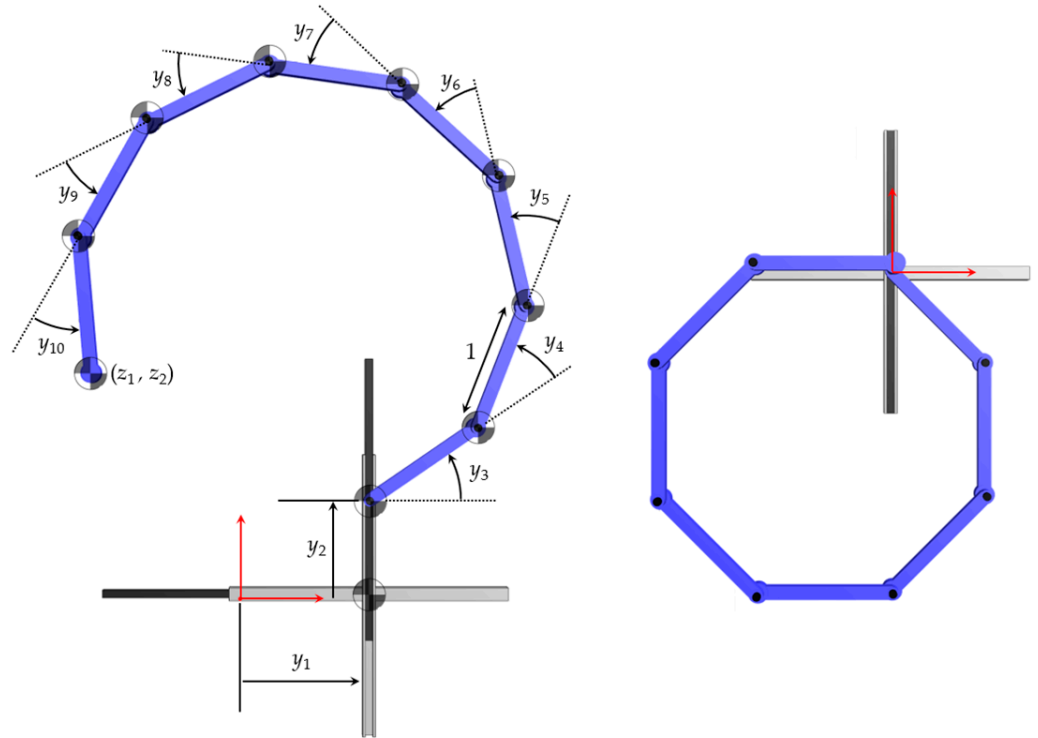


Figure 15. (Left) Manipulator mechanism structural rendering annotated with link axes, centers of mass, and joint displacements. (Right) Initial configuration $\mathbf{y}_0$.

6.1. Traditional Task Space Control

The first test is with the traditional task space controller, causing the end-effector of the manipulator to follow the desired figure 8 shaped trajectory of Equation (42) with no gravitational effects, i.e., neglecting the term $\mathbf{Q}^y(\mathbf{y})$ of Equation (24). This assumes that the manipulator moves in a horizontal plane, perpendicular to the gravity vector. The traditional task space controller projects the dynamics of the manipulator to its task space $\mathbf{z}$ by means of

$$\mathbf{F}^z = \Lambda(\mathbf{y})\ddot{\mathbf{z}} + \mu(\mathbf{y}, \dot{\mathbf{y}}) + \mathbf{p}(\mathbf{y}) \quad (43)$$

where $\Lambda(\mathbf{y}) = [\mathbf{G}'(\mathbf{y})\mathbf{M}^{-1}(\mathbf{y})\mathbf{G}'^T(\mathbf{y})]^{-1}$, $\mathbf{p}(\mathbf{y}) = -\Lambda(\mathbf{y})\mathbf{G}'(\mathbf{y})\mathbf{M}^{-1}(\mathbf{y})\mathbf{Q}^y(\mathbf{y})$, and $\mu(\mathbf{y}, \dot{\mathbf{y}}) = -\Lambda(\mathbf{y})\mathbf{G}'(\mathbf{y})\mathbf{M}^{-1}(\mathbf{y})\mathbf{S}(\mathbf{y}, \dot{\mathbf{y}}) - \Lambda(\mathbf{y})\dot{\mathbf{G}}'(\mathbf{y})\dot{\mathbf{y}}$. A Computed-Torque Control (CTC) law for $\mathbf{F}^z$ [26] that achieves asymptotic tracking of desired trajectories for the output variables can be constructed by substituting for $\ddot{\mathbf{z}}$ in (43) an $\mathbf{F}^\star$ defined as $\mathbf{F}^\star = k_z(\mathbf{z}_d - \mathbf{z}) + k_{\dot{z}}(\dot{\mathbf{z}}_d - \dot{\mathbf{z}}) + \ddot{\mathbf{z}}_d$. This can be transformed to generalized input control forces $\mathbf{F}^y$ through $\mathbf{G}'^T$, $\mathbf{F}^y = \mathbf{G}'^T[\Lambda(\mathbf{y})\mathbf{F}^\star + \mu(\mathbf{y}, \dot{\mathbf{y}}) + \mathbf{p}(\mathbf{y})]$. Gains k_z and $k_{\dot{z}}$ can be selected to achieve the desired tracking with the desired dynamic performance. Values $k_z = 100$ and $k_{\dot{z}} = 20$ were selected to cause the tracking error $\mathbf{z}_d - \mathbf{z}$ to converge to zero asymptotically, with a critically damped behavior and a settling time of ≈ 0.5 s. This meant that 95% of the error would be removed in 0.5 s. The application of the controller given by Equation (43) generated the results shown in Figure 16, in which one can observe that the input time history is not cyclic. Animations in Supplementary Materials emphasize this behavior.

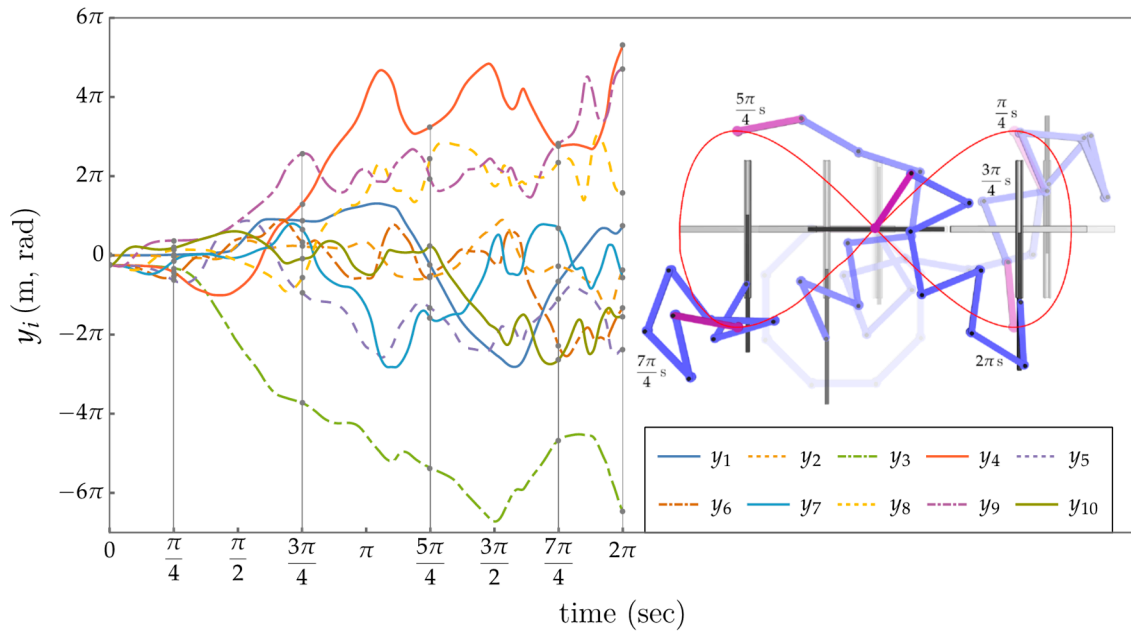


Figure 16. Control results using the traditional operational space controller. Non-cyclic evolution of input coordinates (Left). Some snapshots of the manipulator executing the desired task trajectory (Right).

6.2. Extended Operational Space Control

For the extended operational space controller, the parameterization was initialized at $\bar{y} = y(0)$ with U^1 and V^1 evaluated using Equation (10) and initial $\bar{v} = 0$. With these values and matrices of Section 3.2, the following control law was defined:

$$F^y = MHF^\star + ME - R \quad (44)$$

where

$$F^\star = \begin{bmatrix} k_z(z_d - z) + k_{\dot{z}}(\dot{z}_d - \dot{z}) + \ddot{z}_d \\ k_v(v_d - v) + k_{\dot{v}}(\dot{v}_d - \dot{v}) + \ddot{v}_d \end{bmatrix} \quad (45)$$

and the desired trajectory for v was $v_d(t) = 0$. Gains were chosen as $k_z = 100$, $k_{\dot{z}} = 20$, $k_v = 100$ and $k_{\dot{v}} = 20$.

The application of this controller generated the results shown in Figure 17, where gravity has been suppressed. Comparing this result with Figure 16, it is seen that the extended operational space controller not only achieves a cyclic behavior in the inputs, but also generates a much smoother and lower magnitude input, compared to the wilder variation in inputs in Figure 16. An even greater contrast in behavior of the two control approaches is provided by the kinetic energy plots in Figure 18. Clearly the motion generated by the traditional task space approach is the more extreme. To better appreciate the distinct dynamic nature of the motions, see the attached animations. Despite the large motion experienced by the manipulator in Figure 17, the initial basis (U^1 , V^1) was valid during the entire trajectory, without having to reparametrize due to the rank deficiency of G^U .

Interestingly, setting the second row of $F^\star$ to zero in Equation (45), which corresponds to the CTC controller [26] for v , yields exactly the same results as those in Figure 17. This means exerting no control at all on the dynamics of v , i.e., leaving them unactuated. This works only because the desired evolution for v coincides with its initial value (0), its initial velocity is zero, and there are no external forces like gravity perturbing the dynamics, so v can be left unactuated while controlling only z . If gravity is considered, then it is necessary to keep the second row of $F^\star$ to achieve the desired control.

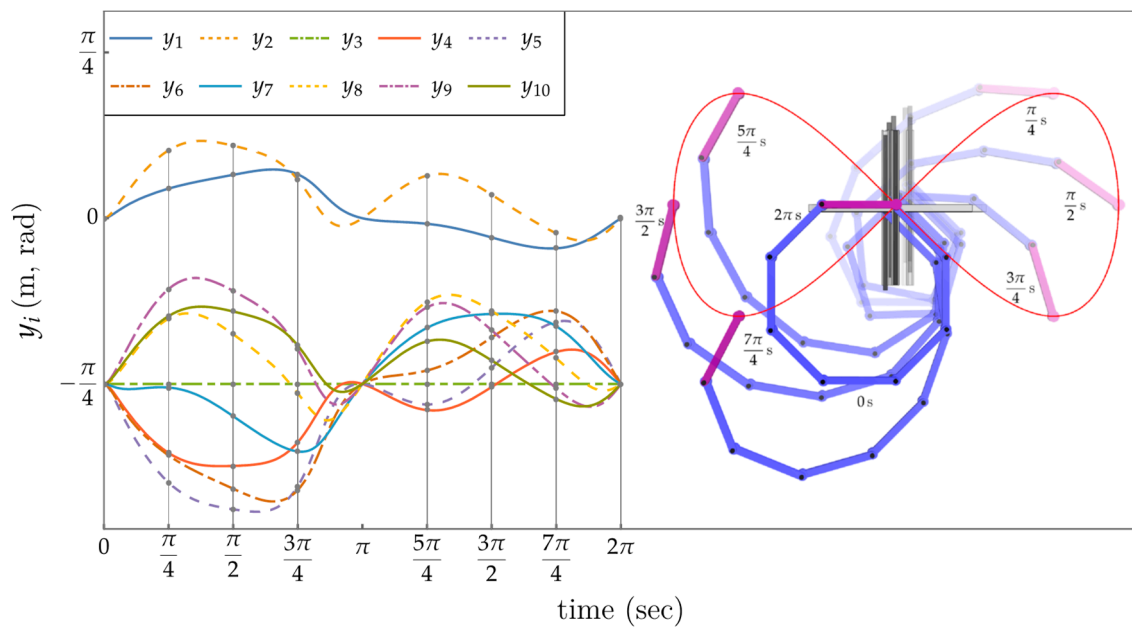


Figure 17. Control results using the extended operational space controller. Cyclic evolution of input coordinates after one cycle of the output variables (**Left**). Some snapshots of the manipulator executing the desired trajectory (**Right**).

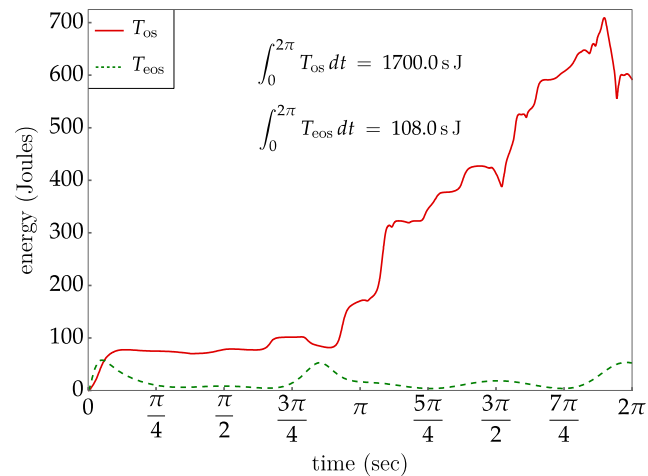


Figure 18. Kinetic energy of traditional task space- and extended operational space-controlled robots vs. time.

7. Conclusions and Recommendations for Future Research

7.1. Conclusions

The extended operational space kinematics, dynamics, and control formulation presented is applicable to a broad spectrum of redundant serial robots. The analytical form of robot inverse kinematics presented depends explicitly on self-motion coordinates that represent robot redundancy. This was shown to correct fundamental errors in generalized inverse velocity kinematics formulations that have been used in the redundant robot literature for over half a century. The extended operational space ODE obtained, in terms of output and self-motion coordinates, quantitatively represent robot redundancy and were shown to be ideally suited for robot control.

Based on the extended operational configuration space ODEs presented, a robot control architecture was defined that exploited the potential offered by redundant serial robots, using self-motion coordinates that characterized robot redundancy. Four one-degree-of-redundancy applications were presented in Section 5, demonstrating the accuracy

and functionality of the control architecture, including output trajectory tracking, while minimizing system kinetic energy and avoiding obstacles. Using traditional task space tracking control as a baseline, comparisons were made between task space and extended operational space control. For the figure 8 shaped tracking task, both the task space controller and the extended operational space controller performed the tracking task without any visible error. Both controllers exploited knowledge of system dynamics in the feed-forward path and the same modest proportional–derivative gains in the feedback path. The combination leads to feedback linearization and a critically damped behavior. Robustness to plant estimates and disturbances is to be assessed in the future.

The self-motion control in examples presented in Section 5 demonstrated classic dynamic decoupling between task and self-motion spaces. This is not surprising, since it was included by design in both the traditional task space formulation and the extended operational space formulation. However, the extended operational space formulation provided a means for exploiting self-motion coordinates that represent robot redundancy, an option that was not available in previous redundant serial robot kinematics formulations. As noted, in the traditional task space approach, there are no explicit coordinates that span the self-motion space, only an instantaneous tangent space. Consequently, task space self-motion control is limited to projecting a vector quantity of interest onto a linear space that is decoupled from the task control. The extended operational space approach was shown to be especially effective in the gradient-based minimization of desired quantities, performing these minimizations better than the task space approach. Tracking of self-motion coordinates was carried out easily and with great fidelity, which could not be achieved with traditional task space control.

A further comparison between traditional and extended operational formulations applied to an eight-degree-of-redundancy robot in Section 6 showed that controllers based on the traditional task space formulation produced much larger variations in internal self-motion than the extended operational space formulation. The new formulation moderated self-motion under control, while preserving the cyclicity of the configuration under specified cyclic task space trajectories. This was illustrated in both animations of the manipulator and in the comparison of mechanism kinetic energy in Figure 18.

7.2. Future Research

Research is needed in defining singularity-free subdomains of the extended operational configuration space. Following the approach presented herein, recent results on redundant robot kinematics and dynamics of non-serial robots [27,28] may be extended to control of non-serial robots.

While an extended operational space adaptation of feedback linearization was successfully employed, future work should consider more sophisticated control approaches. Among these are the use of Model Predictive Control (MPC) [29] to find globally optimal control inputs, as well as approaches such as sliding-mode control to improve robustness to internal modeling errors and external disturbances.

The examples treated in this paper have enabled a comparison between the extended operational space control method presented herein and the traditional task space control method [4]. However, the latter is not the only operational space robot control method that has appeared in the literature. There exist several variations, most derived using the generalized inverse velocity relation of Equation (4), which implies that they inherit the deficiencies discussed in Section 2.2. Some relevant benchmarks with which to compare in the future are the eight controllers evaluated in [5] that can be classified into three families, depending on the type of command that is specified to control the robot: velocity command, acceleration command, and torque command. These controllers incorporate null-space projections at velocity, acceleration, and torque levels, respectively, which exploit redundant degrees of freedom to minimize scalar functions that represent proximity to a preferred configuration, obstacle avoidance, or singularity avoidance. The authors are in the process of comparing the performance of extended operational space controllers with these families

of controllers. It is expected that these comparisons will confirm at least two advantages of the extended operational space formulation.

First, regarding kinetic energy minimization, controllers derived based on the acceleration formulation of [30] incorporate a null-space acceleration that includes the time derivative of the gradient of a cost function. If the cost function is kinetic energy, this implies that the null-space acceleration will depend on joint accelerations that must be measured and fed back to the controller to generate control torques. However, typical feedback control laws in robots only feed states back, i.e., configuration coordinates and their velocities, and not accelerations. Acceleration feedback would produce an algebraic loop, as it would be necessary to use joint accelerations to compute control torques that are applied to the manipulator, in turn generating joint accelerations and leading to a circular relationship. Regarding torque-level controllers derived from [4], the comparison of Figure 10 has shown that the extended operational space controller can achieve a major reduction in kinetic energy.

Second, controllers based on generalized inverse velocity equations lack stability of null-space motion. These controllers typically focus on ensuring the stability of task space coordinates. However, as pointed out in [5], the stability of null-space motion is not easily guaranteed for these controllers, as their closed-loop equations are rather complicated. In fact, it is often necessary to include artificial damping terms in the null-space projection to drive null-space motions to zero and prevent the internal motion of the robot from becoming unstable. This can be observed in the eight-degree-of-redundancy example of Figure 16, where task space control led to motion that was rather chaotic and potentially unstable in the long term. On the contrary, as demonstrated in Equation (37), the extended operational space formulation guarantees that the tracking error of all extended operational coordinates (both task and redundant coordinates corresponding to null-space motion) obey the homogeneous second-order linear differential equation of Equation (37). This implies that the tracking error of redundant coordinates that generate null-space motion, not only task coordinates, can be made to converge exponentially to zero with the desired transient behavior. This can be observed by comparing the smooth and ordered motion of the robots in Figures 9 and 17.

Another family of controllers that will be compared with the extended operational space formulation in the future are those that minimize control torques. An overview of these controllers is given in [31], which highlights the need to include artificial damping to stabilize null-space motion and proposes a new torque-minimizing controller under large external forces applied along the dimensions of the task coordinates. All controllers reviewed in [31] are based on generalized inverse velocity equations, with the exception of [32], which formulates a quadratic program and includes the velocity equation without inverting it. In light of the discussion above regarding smoothness and stability of motions generated by extended operational space controllers, it is expected that they will exhibit superior performance regarding torque minimization.

Supplementary Materials: The following supporting video information can be downloaded at <https://www.mdpi.com/article/10.3390/robotics13120170/s1>, Video S1: task tracking using traditional operational space control, Video S2: task and self-motion tracking using extended operational space control, Video S3: task tracking and kinetic energy minimization using traditional and extended operational space control, Video S4: task tracking and obstacle avoidance using extended operational space control, Video S5: tasktracking of a hyper redundant manipulator using traditional operational space control, Video S6: task tracking of a hyper redundant manipulator using extended operational space control.

Author Contributions: E.J.H. and V.D.S. conceived and designed the study; E.J.H. developed the differential geometry results; V.D.S. and A.P. developed the control results; E.J.H., V.D.S. and A.P. wrote the article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article and Supplementary Materials.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chiaverini, S.; Oriolo, G.; Maciejewski, A.A. Redundant Robots. In *Springer Handbook of Robotics*, 2nd ed.; Siciliano, B., Khatib, O., Eds.; Springer: Berlin, Germany, 2016.
- Whitney, D.E. Resolved Motion Rate Control of Manipulators and Human Prostheses. *IEEE Trans. Man-Mach. Syst.* **1969**, *10*, 47–53. [\[CrossRef\]](#)
- Haug, E.J. Redundant Serial Manipulator Inverse Position Kinematics and Dynamics. *J. Mech. Robot.* **2024**, *16*, 081008. [\[CrossRef\]](#)
- Khatib, O. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE J. Robot. Autom.* **1987**, *RA-3*, 43–53. [\[CrossRef\]](#)
- Nakanishi, J.; Cory, R.; Mistry, M.; Peters, J.; Schaal, S. Operational Space Control: A Theoretical and Empirical Comparison. *Int. J. Robot. Res.* **2008**, *27*, 737–757. [\[CrossRef\]](#)
- Park, K.C.; Chang, P.-H.; Lee, S. Analysis and control of redundant manipulator dynamics based on an extended operational space. *Robotica* **2001**, *19*, 649–662. [\[CrossRef\]](#)
- Pars, L.A. *A Treatise on Analytical Dynamics*; Ox Bow Press: Woodbridge, CT, USA, 1965.
- Haack, W.; Wendland, W. *Lectures on Partial and Pfaffian Differential Equations*; Pergamon Press: Oxford, UK, 1972.
- De Luca, A.; Oriolo, G. Nonholonomic Behavior in Redundant Robots Under Kinematic Control. *IEEE Trans. Robot. Autom.* **1997**, *13*, 776–782. [\[CrossRef\]](#)
- Shamir, T.; Yomdin, Y. Repeatability of Redundant Manipulators: Mathematical Solution of the Problem. *IEEE Trans. Autom. Control* **1988**, *33*, 1004–1009. [\[CrossRef\]](#)
- Bowling, A.; Harmeyer, S. Repeatable Redundant Manipulator Control Using Nullspace Quasivelocities. *J. Dyn. Syst. Meas. Control* **2010**, *132*, 31007. [\[CrossRef\]](#)
- Klein, C.A.; Huang, C.-H. Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators. *IEEE Trans. Syst. Man Cybern.* **1983**, *SMC-13*, 245–250. [\[CrossRef\]](#)
- Khatib, O. Inertial Properties in Robotic Manipulation: An Object-Level Framework. *Int. J. Robot. Res.* **1995**, *13*, 19–36. [\[CrossRef\]](#)
- Deo, A.; Walker, I. Overview of Damped Least-Squares Methods for Inverse Kinematics of Robot Manipulators. *J. Intell. Robot. Syst.* **1995**, *14*, 43–68. [\[CrossRef\]](#)
- Hogan, N. Impedance Control: An Approach to Manipulation: Part I—Theory, Part II—Implementation, Part III—Applications. *J. Dyn. Syst. Meas. Control* **1985**, *107*, 1–24. [\[CrossRef\]](#)
- Hollerbach, J.; Suh, K.C. Redundancy Resolution of Manipulators Through Torque Optimization. *IEEE J. Robot. Autom.* **1987**, *RA-3*, 308–316. [\[CrossRef\]](#)
- Simas, H.; Di Gregorio, R. A Technique Based on Adaptive Extended Jacobians for Improving the Robustness of the Inverse Numerical Kinematics of Redundant Robots. *J. Mech. Robot.* **2010**, *11*, 020913. [\[CrossRef\]](#)
- Dietrich, A.; Ott, C.; Albu-Schäffer, A. An Overview of Null Space Projections for Redundant, Torque-Controlled Robots. *Int. J. Robot. Res.* **2015**, *34*, 1385–1400. [\[CrossRef\]](#)
- Haug, E.J. *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume II: Modern Methods*, 4th ed.; Ed & Carol Haug Charitable Foundation: Silver City, NM, USA, 2024.
- Lee, J.M. *Introduction to Smooth Manifolds*, 2nd ed.; Springer: New York, NY, USA, 2013.
- Robbin, J.W.; Salamon, D.A. *Introduction to Differential Geometry*; Springer: Berlin, Germany, 2022.
- Peidro, A.; Haug, E.J. Obstacle Avoidance in Operational Configuration Space of Kinematically Redundant Serial Manipulators. *Machines* **2024**, *12*, 10. [\[CrossRef\]](#)
- Atkinson, K.E. *An Introduction to Numerical Analysis*, 2nd ed.; Wiley: New York, NY, USA, 1989.
- Corwin, L.J.; Szczerba, R.H. *Multivariable Calculus*; Marcel Dekker: New York, NY, USA, 1982.
- Haug, E.J.; Peidro, A. Redundant Manipulator Kinematics and Dynamics on Differentiable Manifolds. *J. Comput. Nonlinear Dyn.* **2022**, *17*, 111008. [\[CrossRef\]](#)
- Chung, W.K.; Fu, L.C.; Kröger, T. Motion control. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin, Germany, 2016; pp. 174–177.
- Haug, E.J. Redundant Non-Serial Implicit Manipulator Kinematics and Dynamics. *J. Mech. Robot.* **2024**, *16*, 061017. [\[CrossRef\]](#)
- Haug, E.J. Redundant Non-Serial Compound Manipulator Kinematics and Dynamics. *Mech. Mach. Theory* **2024**, *200*, 105717. [\[CrossRef\]](#)
- Nicolis, D.; Allevi, F.; Rocco, P. Operational space model predictive sliding mode control for redundant manipulators. *IEEE Trans. Robot.* **2020**, *36*, 1348–1355. [\[CrossRef\]](#)
- Hsu, P.; Hauser, J.; Sastry, S. Dynamic control of redundant manipulators. *J. Robot. Syst.* **1989**, *6*, 133–148.

31. Woolfrey, J.; Lu, W.; Liu, D. A control method for joint torque minimization of redundant manipulators handling large external forces. *J. Intell. Robot. Syst.* **2019**, *96*, 3–16.
32. Zhang, Y.; Guo, D.; Ma, S. Different-level simultaneous minimization of joint-velocity and joint-torque for redundant robot manipulators. *J. Intell. Robot. Syst.* **2013**, *72*, 301–323.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.